# ALGORITMI E METODI MATEMATICI IN COMPUTER GRAPHICS

**Massimo Picardello** 

Università di Roma "Tor Vergata" Dipartimento di Matematica

La Computer Graphics è la scienza che descrive come una scena tridimensionale modellata viene proiettata prospetticamente su un rettangolo, detto viewport, in un piano di visuale, in maniera da creare in guesto rettangolo un'immagine quanto più possibile fotorealistica. Per questo occorre convenire anzitutto su cosa sia un'immagine. In questo libro, un'immagine è una matrice rettangolare di elementi che corrispondono a celle di colore costante, ossia che contengono ciascuno un colore, o meglio le coordinate che individuano un colore in un appropriato spazio di colore. Questi elementi si chiamano pixel (abbreviazione per picture element). Ciascun pixel, quindi, corrisponde ad una cella rettangolare nel viewport che viene colorata con un colore costante, quello individuato dal pixel: l'immagine nel viewport è una tessellazione di guesti pixel colorati. Evidentemente, ai fini del fotorealismo, occorre chela risoluzione sia elevata, ossia che i pixel siano molti, in modo che le celle rettangolare siano piccole: altrimenti si ottiene una scalettatura visibile. Occorre inoltre che la trasformazione dalla scena tridimensionale all'immagine sul viewport sia operata utilizzando un metodo prospettico realistico, ed una resa accurata degli effetti delle luci sui materiali di cui si compone la scena modellata (rendering).

Il primo volume tratta le questioni preliminari: i modelli con cui si descrivono i colori, la teoria matematica della prospettiva, algoritmi veloci per convertire rette e curve nella scena tridimensionale a segmenti o curve bidimensionali nel viewport (*scan conversion*), ed i fenomeni che intervengono quando la risoluzione non è abbastanza elevata (*aliasing*).

Il secondo volume presenta gli algoritmi classici della Computer Graphics: dapprima si studia la rimozione di aree nascoste (z-buffer, partizione binaria dello spazio ed ordinamenti a lista di priorità, scansioni di righe, suddivisione di area, ray tracing) e poi il rendering dell'illuminazione, della trasparenza e delle ombre (modelli di illuminazione, mappe di tessitura, di rilievo, di ombre, ray tracing ricorsivo, radiosità).

Infine, il terzo volume tratta dei metodi analitici e soprattutto probabilistici dell'Illuminazione globale. Una parte ancora in via di sviluppo verrà presto aggiunta per descrivere metodi adeguati al rendering di caustiche: mappa dei fotoni, algoritmo di Metropolis, mezzi permeabili.

# **VOLUME 1**

PRELIMINARI: MODELLI DI COLORE, ALIASING, SCAN CONVERSION E PROIEZIONI PROSPETTICHE

# PRIMA PARTE TEORIA DEL COLORE

MODELLI MATEMATICI PER IL COLORE

## 1 Distribuzione spettrale della luce e lunghezza d'onda dominante

La luce visibile consiste di onde elettromagnetiche di lunghezza d'onda compresa fra 400nm (violetto) e 700nm (rosso).

Quando percepiamo un colore, la luce che vediamo non è di solito monocromatica, bensì ha una distribuzione di energia al variare della lunghezza d'onda.

Questa distribuzione viene chiamata distribuzione spettrale ed indicata con  $P(\lambda)$ .



Figura 1: Distribuzione spettrale della luce monocromatica

Nella Figura 1,  $P(\lambda)$  è una distribuzione, la delta di Dirac, a  $\lambda = \lambda_0$ : vediamo luce monocromatica a lunghezza d'onda  $\lambda_0$ .



Figura 2: Distribuzione spettrale della luce bianca o grigia

Nella Figura 2,

$$P(\lambda) = \begin{cases} 1 & \text{se } 400 \le \lambda \le 700 \\ 0 & \text{altrimenti} \end{cases}$$

Qui 1=100% sta simbolicamente per la luminosità massima, su un monitor con 255 livelli di grigio avremmo forse scelto 255. Vediamo luce bianca (o grigia, a seconda dell'intensità). La luce bianca (o grigia), cioè senza dominanti cromatiche, ha una distribuzione spettrale equidistribuita fra le varie frequenze. Se sommiamo le distribuzioni in questi due esempi, come nella Figura 3, otteniamo luce colorata (lunghezza d'onda  $\lambda_0$ ) con una componente addizionale di grigio: cioè vediamo il colore corrispondente alla lunghezza d'onda  $\lambda_0$  ma desaturato.

Si dice che  $\lambda_0$  è la lunghezza d'onda dominante.



Figura 3: Distribuzione spettrale di un colore dominante desaturato

## 2 Lo stesso colore può essere ottenuto da distribuzioni spettrali differenti

DEFINIZIONE 1. Due colori la cui somma è il bianco si dicono complementari.

Sia  $P_w(\lambda)$  la distribuzione spettrale della luce bianca che scegliamo come bianco standard. Ad esempio, supponiamo che  $P_w(\lambda)$  sia come alla figura 2 (di solito però la distribuzione spettrale del bianco è piatta, perché dipende anche dalla curva di risposta del nostro sistema visivo: si veda la Figura 11) nel seguito.

Ora sia  $P(\lambda)$  la distribuzione spettrale di un altro colore. Allora il colore complementare a  $P(\lambda)$  ha distribuzione  $Q(\lambda) = P_w(\lambda) - P(\lambda)$ .



Figura 4: Colori complementari

Se consideriamo una distribuzione spettrale che approssima quelle della Figura 3, ma senza componenti discrete monocromatiche, percepiamo lo stesso colore della distribuzione in Figura 3.

Cautela: la lunghezza d'onda dominante potrebbe non coincidere esattamente col punto di massimo di  $P(\lambda)$ . Si veda il grafico in Figura 6.

La lunghezza d'onda dominante è  $\lambda_0$ , i punti di massimo di  $P(\lambda) : \lambda_+, \lambda_-$ . Pertanto, differenti distribuzioni spettrali  $P(\lambda)$ , possono corrispondere alla



Figura 5: Distribuzione spettrale con una componente discreta monocromatica



Figura 6: Distribuzione spettrale con più componenti discrete monocromatiche

percezione dello stesso colore.

Tutti i colori (oppure i loro complementari: si veda la sezione 12 nel seguito) si possono classificare in termini di tre parametri:

- 1. croma (o tinta): la lunghezza d'onda dominante
- 2. luminosità (l'intensità del colore)
- 3. saturazione o purezza (una bassa saturazione significa che diluizione con componenti di grigio è elevata).



Figura 7: Distribuzione spettrale per la stessa tinta a due diversi livelli di saturazione

## 3 Croma, Saturazione e Luminosità: tre parametri per classificare i colori

Il croma è il colore monocromatico (o il suo complementare) alla massima saturazione e luminosità. Può essere messo in corrispondenza con un angolo fra  $0^{\circ}$  e  $360^{\circ}$  (scala del croma). Si ottiene quindi il seguente diagramma croma-saturazione.



Figura 8: Diagramma croma-saturazione

La circonferenza esterna corrisponde alla massima saturazione. Per saturazione uguale a zero si ottiene al centro il grigio (il bianco se siamo alla massima intensità). Facendo variare anche l'intensità si ottiene un cono.



Figura 9: Diagramma croma-saturazione-intensità

#### 4 Percezione del colore: teoria del tristimolo

Una teoria della biofisica della percezione ottica, confermata sperimentalmente, asserisce che la retina dispone di tre tipi di sensori, detti coni, con sensibilità centrata sulle bande di lunghezza d'onda corrispondenti rispettivamente al blu al verde e al rosso. Questa teoria è comoda per la modellazione del colore per emissione (come quello dei fosfori di un monitor) nel quale il colore si parametrizza con le intensità nei tre canali primari rosso, verde e blu (RGB).



Figura 10: Curve di sensibilità ai colori primari RGB

Si noti come i sensori blu assorbano meno luce: siamo meno sensibili alla luce blu, e più in generale alla banda a bassa lunghezza d'onda (alta frequenza) dal violetto al blu.

#### 5 Curva di sensibilità al colore

Dalle curve di sensibilità ai colori RGB ricaviamo la curva di sensibilità globale del nostro sistema visivo al colore, cioè la sua distribuzione di sensibilità relativa al variare della lunghezza d'onda; il massimo della sensibilità relativa si ha in corrispondenza di  $\lambda = 550nm$  (giallo-verde). La curva della sensibilità relativa nella Figura 5.1 è la somma dei tre grafici nella Figura 10.



Figura 11: Curva di sensibilità al colore

## 6 Curve di corrispondenza al colore (color matching curves)

Segue dalla teoria del tristimolo che il sistema visivo percepisce ogni colore come una distribuzione lineare a coefficienti (cioè intensità) positivi delle componenti primarie rosso, verde e blu.

In realtà questa asserzione non è precisa. Per capire perché è necessario determinare sperimentalmente le curve di corrispondenza al colore.

DEFINIZIONE 2. Si chiamano curve di corrispondenza al colore i grafici delle tre funzioni  $\bar{r}$ ,  $\bar{g}$ ,  $\bar{b}$  di  $\lambda$  il cui valore è l'intensità delle tre primarie rossa, verde e blu che corrisponde alla percezione del colore saturo monocromatico di lunghezza d'onda  $\lambda$  (o più in generale il colore saturo di lunghezza d'onda dominante  $\lambda$ ) di luminosità costante massima al variare di  $\lambda$ .

Denotiamo queste curve con  $\bar{r}_{\lambda}$ ,  $\bar{g}_{\lambda}$ ,  $\bar{b}_{\lambda}$ . Esse sono illustrate nella figura 6.1. Questi grafici rappresentano le intensità percentuali di rosso, verde e blu tali che il nostro sistema visivo veda il colore di lunghezza d'onda  $\lambda$  alla massima purezza e ad intensità massima e costante.



Figura 12: Le curve RGB di corrispondenza del colore (color matching curves)

Come si vede, la curva del rosso è al di sotto dell'asse delle ascisse nell'intervallo tra  $\lambda_0 \in \lambda_1$ . Questo significa che i colori in questa banda alla massima saturazione e luminosità non sono sintetizzabili sommando le componenti rosse, verdi e blu; però, se ad uno di questi colori si aggiunge una opportuna quantità di rosso (quella che compensa esattamente il valore negativo) allora il colore risultante è sintetizzabile (è in questo modo che sono stati misurati i valori negativi del grafico). Quindi non tutti i colori visibili sono riproducibili nel modello RGB. Questo fatto non dipende dalla qualità del monitor usato, ma dalla psicofisica del sistema visivo.

## 7 Risoluzione del colore da parte del nostro sistema visivo al variare della lunghezza d'onda

La capacità del sistema visivo di riconoscere colori della stessa luminosità e saturazione che differiscono solo per la tinta (cioè per la lunghezza d'onda) è stata determinata sperimentalmente su campioni di osservatori. Le differenze minime di lunghezza d'onda a cui il sistema visivo riconosce colori diversi cambiano con la lunghezza d'onda. La risoluzione è peggiore in prossimità del violetto e del rosso, come mostrato nella Figura ??.



Figura 13: Risoluzione del sistema visivo umano al variare della lunghezza d'onda

Questo risultato sperimentale è in accordo con l'altro fatto sperimentale visto nell' Esempio 10, cioè che il sistema visivo è meno sensibile alla componente blu che alle altre. Al diminuire della saturazione e della luminosità i colori si appiattiscono verso un colore costante (bianco, grigio o nero) e la risoluzione del sistema visivo, ovviamente, decresce.

## 8 Le curve di corrispondenza CIE (modello CIE-XYZ, detto anche LAB)

Un modello di colore nel quale si può ottenere ogni colore visibile sommando tre primarie è stato proposto nel 1931 dalla Commission Internazionale de l'Eclairage (CIE). Questo modello, che usa tre curve di corrispondenza che indichiamo con  $\bar{x}, \bar{y}, \bar{z}$  risolvere il problema dei valori negativi della Figura 12. Il problema è dovuto alla incapacità del nostro sistema visivo a sintetizzare i colori alla massima luminosità e saturazione in certe bande di frequenza partendo da componenti R, G, B a causa della insufficiente curva di risposta dei suoi coni. Per evitare questo problema, la primaria CIE che misura la luminosità viene scelta come la sensibilità del sistema visivo alla luminosità del colore. Quindi la curva  $\bar{y}$  è la stessa della Figura 11. Le altre curve sono definite in maniera appropriata (come differenze di opportuni canali di colori complementari, rispettivamente verde-magenta e blu-giallo, o, più precisamente delle corrispondenti color-matching curves del modello RGB o CMY). Si ottengono le curve seguenti:



Figura 14: Le primarie CIE-XYZ

Si noti che  $\bar{x}$  si annulla alla lunghezza d'onda per cui le color-matching curves blu e verde coincidono, cioè quella dove si annulla la curva  $\bar{z}$ . Per tale lunghezza d'onda, pari a circa 500 nm, la sensibilità del sistema visivo alla luminosità del colore è tutta basata sul canale relativo a  $\bar{x}$  (cioè verdemagenta). Però, anche se  $\bar{x}$  si annulla per  $\lambda = 500nm$ , le tre color-matching curves non diventano mai negative. Con queste primarie è quindi è possibile ricostruire qualsiasi colore. La componente nel canale di rappresenta la luminosità del colore ricostruito. Indichiamo con  $\underline{X}, \underline{Y}, \underline{Z}$  i tre canali del modello CIE. Le curve di corrispondenza dell' Esempio 14 ci dicono in che percentuali  $\bar{x}_{\lambda}, \bar{y}_{\lambda}, \bar{z}_{\lambda}$  si devono mescolare le intensità di questi canali per ricostruire colori che appaiano di luminosità costante.

## 9 Ricostruzione del colore nel modello CIE-XYZ

Consideriamo un colore di distribuzione spettrale  $P(\lambda)$ . Quali sono le sue coordinate X, Y, Z nel modello CIE? Dobbiamo considerare i prodotti scalari con le tre funzioni di ricostruzione  $\bar{x}, \bar{y}, \bar{z}$  opportunamente normalizzati:

$$X = c \int_{-\infty}^{+\infty} P(\lambda)\bar{x}(\lambda) \, d\lambda$$
$$Y = c \int_{-\infty}^{+\infty} P(\lambda)\bar{y}(\lambda) \, d\lambda$$
$$Z = c \int_{-\infty}^{+\infty} P(\lambda)\bar{z}(\lambda) \, d\lambda$$

La costante di normalizzazione è scelta in modo da avere il valore convenzionale 100 al punto di bianco di una immagine stampata, cioè

$$c = \frac{1}{\int_{-\infty}^{+\infty} P_{bianco}(\lambda)\bar{y}(\lambda) \, d\lambda}$$

dove  $P_{bianco}(\lambda)$  è la distribuzione spettrale del tipo di luce bianca che scegliamo come luce standard.

Il prodotto scalare al denominatore è la luminosità complessiva percepita dal sistema visivo quando esso guarda questa luce bianca (massima intensità d grigio).

Per oggetti che emettono luce, come i monitor, non ha senso scegliere la costante di normalizzazione basandosi sulla luce bianca standard. Infatti la distribuzione spettrale del bianco su un monitor è regolabile, e dipende dal singolo monitor. Per convenzione in tal caso si sceglie di fissare c = 600 lumieres/watt.

#### 10 Coordinate CIE

Le coordinate X, Y, Z del colore **C** nel modello CIE sono i pesi con cui si ricostruisce (o meglio, si sintetizza) il colore **C** come combinazione lineare delle tre funzioni i cui grafici sono le curve di corrispondenza al colore dei tre canali X, Y, Z:

$$\mathbf{C} = X\underline{X} + Y\underline{Y} + Z\underline{Z}$$

#### 11 Coordinate di cromaticità

Queste coordinate si definiscono a partire dalle coordinate CIE X, Y, Z come i valori percentuali rispetto a X + Y + Z (per chiarezza, diamo un nome alla quantità X + Y + Z: chiamiamola *contenuto cromatico*). In altre parole, poniamo:

$$x = \frac{X}{X + Y + Z} \tag{1}$$

$$y = \frac{Y}{X + Y + Z} \tag{2}$$

$$z = \frac{Z}{X + Y + Z} \tag{3}$$

Perciò x + y + z = 1 = 100% Nell'ottante  $X + Y + Z \ge 0$  del modello di colore *CIE* le coordinate x, y, z corrispondono alla proiezione radiale sul piano X + Y + Z = 1.

Queste sono le coordinate dei colori di massima luminosità.

Poiché il sistema visivo vede solo i colori tra  $\lambda = 400nm$ , e  $\lambda = 700nm$ , e con una curva di risposta che si avvicina a zero a questi estremi, non tutto il triangolo sotteso dai vertici (1,0,0), (0,1,0), (0,0,1) corrisponde ai colori visibili di massima luminosità.

La Figura 12 rappresenta lo spazio dei colori visibili:



Figura 15: Spazio dei colori visibili



Figura 16: Diagramma di cromaticità

Si tratta quindi di un cono contenuto nell'ottante positivo, con vertice nell'origine. La base del cono, cioè la faccia piatta, quella che non tocca l'origine, si chiama il *diagramma di cromaticità*.

E chiaro che per determinare un punto in questa faccia primaria bastano due parametri.

Decidiamo di scegliere come parametri  $x \in y$ .

Infatti, poiché , x+y+z = 1 si ha z = 1-x-y; pertanto possiamo ricostruire ogni colore  $\mathbf{C} = X\underline{X} + Y\underline{Y} + Z\underline{Z}$ , con questi due parametri.

In effetti, essi corrispondono alla versione di  $\mathbf{C}$  di massimo contenuto cromatico X+Y+Z=1. Per ricostruire  $\mathbf{C}$  si specifica, ad esempio, anche la sua luminosità, cioè la coordinata Y. Così abbiamo costruito una corrispondenza

$$\mathbf{C} \to (X, Y, Z) \to (x, y, Y) ; \tag{4}$$

la regola di corrispondenza si ricava dall'equazione x + y + z = 1 = 100%. Nell'ottante  $X + Y + Z \ge 0$  del modello di colore CIE le coordinate x, y, z corrispondono alla proiezione radiale sul piano X + Y + Z = 1 da cui si ottiene

$$\frac{x}{y} = \frac{X}{Y}$$
$$\frac{z}{y} = \frac{1 - x - y}{y} = \frac{Z}{Y}$$

Perciò

$$X = \frac{x}{y}Y$$
$$Y = y$$
$$Z = \frac{1 - x - x}{y}Y$$

#### 12 Diagramma di cromaticità

La parte curva del bordo del diagramma di cromaticità nella Figura corrisponde a tinte sature di massimo contenuto cromatico. Le loro distribuzioni spettrali sono percepite dal sistema visivo come colori monocromatici, che variano dal violetto al rosso.

All'interno del diagramma ci sono varianti meno sature degli stessi colori.

Un punto centrale corrisponde al bianco (saturazione zero, luminosità massima). Questo punto è scelto convenzionalmente come il colore corrispondente ad una fissata distribuzione spettrale vicina a quella della luce solare, ad esempio a mezzogiorno (5600°K). Esso ha coordinate vicine a  $x = \frac{1}{3}, y = \frac{1}{3}, z = \frac{1}{3}$ . Indichiamo questo punto con w. Si possono scegliere punti di bianco differenti, ad esempio in corrispondenza a temperature di corpo nero diverse. Una temperatura di corpo nero più elevata si avvicina all'azzurro, una più bassa si avvicina al rosso. Spesso i monitor vengono calibrati perché il. loro punto di bianco corrisponda alla temperatura di 9300°K: nella Figura 17 riportiamo la traiettoria nel diagramma di cromaticità dei punti di bianco al variare della loro temperatura, ed indichiamo il corrispondente punto del diagramma di cromaticità con D93. Analogamente indichiamo il bianco arancio) con D40.

Il segmento da w ad un punto del bordo corrispondente al colore di lunghezza d'onda  $\lambda_0$ , percorre tutta la scala della saturazione di quel colore, dal bianco al colore puro (completamente saturo).

Il colore **C** di luminosità massima e di distribuzione spettrale  $P(\lambda)$ , corrisponde al punto interno al diagramma di cromaticità ottenuto dalla combinazione convessa (cioè l'integrale normalizzato) con peso  $P(\lambda)$  dei punti del



Figura 17: Diagramma di cromaticità e lunghezza d'onda dominante

bordo del diagramma a lunghezza d'onda  $\lambda$  (quindi quelli corrispondenti alla parte curva del bordo).

A questo punto le coordinate  $x \in y$  di **C** si calcolano come nella sezione precedente.

Questo identifica C (o meglio la sua proiezione a massimo contenuto cromatico) come un punto del diagramma di cromaticità.

Ad esempio nella Figura 18 percorrendo la corda da w a  $\mathbf{C}$  si tocca il bordo in un punto che corrisponde alla lunghezza d'onda  $\lambda_0$ . Questa lunghezza d'onda è la lunghezza d'onda dominante di  $\mathbf{C}$ .

Questo però è vero se la corda da w a **C** interseca il bordo nella parte curva. Se lo interseca nella parte piatta si ottengono i colori complementari,



Figura 18: Diagramma di cromaticità e lunghezza d'onda dominante

che ora definiamo.

## 13 Visualizzazione geometrica dei colori complementari nel diagramma CIE

Rivediamo la definizione di colore complementare alla luce della parametrizzazione con le coordinate  $x, y \in z$ .

Un colore di massimo contenuto cromatico corrisponde ad un punto sul piano  $X + Y + Z = 1, X, Y, Z \ge 0.$ 

Se sommiamo due colori  $\mathbf{C}_1 = (x_1, y_1, z_1)$  e  $\mathbf{C}_2 = (x_2, y_2, z_2)$  otteniamo il colore corrispondente alle coordinate  $\mathbf{C}_1 + \mathbf{C}_2$ .

Se i contenuti cromatici  $X_1 + Y_1 + Z_1$  di  $\mathbf{C}_1$  e  $X_2 + Y_2 + Z_2$  di  $\mathbf{C}_2$  sono bassi, si ottiene un colore di basso contenuto cromatico, ed il colore corrispondente di massimo conntenuto cromatico è  $\frac{1}{L}(X_1 + X_2, Y_1 + Y_2, Z_1 + Z_2,$ dove  $L = ||(X_1 + X_2, Y_1 + Y_2, Z_1 + Z_2)||.$ 

In tal modo, quando succede che la somma dei due colori è il colore bianco? Per linearità, sul piano di massimo contenuto cromatico X + Y + Z = 1, la somma di  $C_1$  e  $C_2$  è esattamente il punto di mezzo della corda che li congiunge nel diagramma di cromaticità.

Abbiamo quindi la seguente visualizzazione geometrica:

Il complementare di  $\mathbf{C}_1$  è il punto opposto a  $\mathbf{C}_1$  rispetto al punto di bianco w (alla stessa distanza da w ma dall'altro lato); w è il punto di mezzo della corda che congiunge  $\mathbf{C}_1$  e  $\mathbf{C}_2$ .

Nel cono sotteso dal tratto piatto del bordo del diagramma di cromaticità e con centro in w si trovano i colori complementari dei colori dello spettro visibile (colori monocromatici). Si tratta della banda dal porpora al magenta.



Figura 19: Diagramma di cromaticità e colori complementari



Figura 20: La banda dal porpora al magenta nel diagramma di cromaticità

#### 14 Diagramma di cromaticità a luminosità

Il diagramma di cromaticità riguarda solo la sezione del cono di colori visibili che corrisponde al piano di massimo contenuto cromatico X + Y + Z = 1. Se fossimo interessati a un contenuto cromatico inferiore dovremmo scegliere piani di equazione  $X + Y + Z = \ell \text{ con } 0 \le \ell \le 1$ .

I colori puri (quelli con valori massimi di  $X \in Z$ ) sono percepiti in maniera differente a seconda della luminosità Y. Ad esempio, la sovrapposizione di rosso e giallo, in dosi opportune, ad alta luminosità è un colore simile a quello della pelle di un individuo di razza bianca, mentre a bassa luminosità è il marrone.

#### 15 Gamma dei colori riproducubili

Quali sono i colori riproducibili da un monitor?

Il monitor ha tre tipi di fosfori di emissione con distribuzione spettrale centrata rispettivamente sulle bande del rosso, del verde e del blu. Queste distribuzioni corrispondono a colori, quindi a coordinate di cromaticità x e y.

Ecco le coordinate corrispondenti ad un monitor tipico, che adotta lo spazio di colore sRGB (per ciascun monitor esse sono fornite dal costruttore, ma non si discostano mai molto da queste, a meno che il monitor non adotti un altro spazio di colore, magari con gamut più ampio, come Adobe RGB (98) - nella successiva Figura 21 confronteremo questi due gamuts).

	fosfori rossi	fosfori verdi	forfori blu
Х	0.64	0.30	0.15
у	0.33	0.60	0.06

Di nuovo per linearità, la gamma di colori di massima grandezza cromatica riprodotta dal monitor (cioè nel piano X + Y + Z = 1) è il triangolo nel diagramma di cromaticità con vertice in questi tre punti.

*Nota:* poiché la gamma riprodotta dal monitor corrisponde ad un triangolo mentre i bordi del diagramma di cromaticità sono curvi, nessun monitor può visualizzare l'intera gamma di colori visibili.

Lo stesso accade per le stampanti, la cui gamma riproducibile è il triangolo generato dai tre colori alle distribuzioni spettrali dei tre inchiostri ciano, magenta e giallo usati dalla stampante.

Se si usano più inchiostri questo triangolo diventa un poligono, ed approssima meglio l'intero diagramma di cromaticità, ma non può esaurirlo. Per le stampanti, i tre colori corrispondenti ai tre inchiostri sono percepiti in maniera diversa a seconda del tipo di luce bianca usata per generare le immagini e a seconda del tipo di carta per stamparle, quindi la gamma dipende da questi fattori.

#### 16 Il modello LUV

A causa della differente risoluzione del colore da parte del sistema visivo alle diverse lunghezze d'onda, la percezione delle variazioni del colore in seguito ad uno spostamento dei parametri  $x \in y$  del modello CIE a partire dal colore  $\mathbf{C}_1$  per arrivare al colore  $\mathbf{C}_2$  non dipende solo dalla lunghezza dello spostamento, cioè dalla distanza dei punti  $\mathbf{C}_1 \in \mathbf{C}_2$ , ma anche dalla posizione di  $\mathbf{C}_1$ e  $\mathbf{C}_2$ .

Per lo stesso motivo, se i parametri  $x \in y$  variano in linea retta con velocità costante da  $(x_0, y_0)$  a  $(x_1, y_1)$ , il sistema visivo può percepire una variazione



Figura 21: La gamma dei colori riproducibili

di colore non uniforme.

Un modello di colore in cui la variazione sia percettivamente uniforme è stato sviluppato dalla Commissione Internazionale de L'Eclairage nel 1976.

Esso utilizza una variante non lineare, detta LUV, del modello XYZ. In altre parole, le coordinate LUV dipendono tramite espressioni non lineari (che includono potenze) da quelle XYZ. Qui ci limitiamo a presentare queste espressioni senza giustificazione.

e nuove coordinate sono chiamate  $\ell,\, u,\, v$ e sono definite rispetto al punto di bianco.

Scriviamo  $x_w, y_w, z_w$  le coordinate CIE-XYZ del punto di bianco, e consideriamo due nuove quantità:

$$u' = \frac{X}{4X + 15Y + 3Z}$$
$$v' = \frac{Y}{X + 15Y + 3Z}$$

Ecco le espressioni corrispondenti per il punto di bianco:

$$u'_{w} = \frac{4X_{w}}{X_{w} + 15Y_{w} + 3Z_{w}}$$
$$v'_{w} = \frac{4Y_{w}}{X_{w} + 15Y_{w} + 3Z_{w}}$$

Allora  $\ell^*$ ,  $u^*$ ,  $v^*$  sono date da:

$$\ell^* = 116 \left(\frac{Y}{Y_w}\right)^{\frac{1}{3}} - 16$$
$$u^* = 13\ell^*(u' - u'_w)$$
$$v^* = 13\ell^*(v' - v'_w)$$

## 17 Uniformizzazione delle coordinate di colore di due monitor

Come visto nella Figura 21, la gamma di colore di un monitor dipende dalle coordinate di cromaticità dei suoi fosfori, che non sono le stesse per tutti i monitor. Questo significa che un'immagine che ai vari pixel ha determinati valori di luminosità dei tre canali RGB appare leggermente diversa su diversi monitor, a meno che non si applichi una trasformazione di coordinate per correggere le discrepanze. Questa trasformazione di coordinate nel software corrente di trattamento delle immagini (ad esempio nelle ultime versioni di Adobe Photoshop) viene effettuata nella fase di lettura dell'immagine; i dati necessari (ad esempio le coordinate di cromaticità dei fosfori) vengono inseriti una volta per tutte quando si sceglie il *profilo del monitor* in Photoshop.

La trasformazione di coordinate è basata sulla identificazione della gamma riproducibile dal monitor dentro lo spazio di colore CIE. Quindi dobbiamo scrivere la matrice J di trasformazione dalle coordinate R, G, B alle coordinate X, Y, Z:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = J \begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$
(5)

Osserviamo che i coefficienti  $X_r, Y_r, Z_r$  sono le coordinate CIE del rosso più intenso riproducibile dal monitor. Infatti questo rosso di massima luminosità corrisponde alle coordinate (R, G, B) = (1, 0, 0) e quindi alle coordinate X, Y, Z seguenti:

$$\begin{pmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} X_r \\ Y_r \\ Z_r \end{pmatrix}$$
(6)

Analogamente per gli altri coefficienti di J.

Una volta determinata J (lo faremo in seguito) per trasformare le coordinate R, G, B del primo monitor al secondo monitor basta usare le matrici rispettive  $J_1$  e  $J_2$  di conversione alle coordinate CIE e formare la matrice di uniformizzazione  $J_2^{-1}J_1$ .

Naturalmente, ogni colore  $\mathbf{C}_1$  che è nella gamma del primo monitor, ma non in quella del secondo monitor, viene così trasformato in coordinate  $\mathbf{C}_2 = J_2^{-1} J_1 \mathbf{C}_1$  che sono al di fuori del cubo unitario del modello RGB, e quindi non riproducibili dal secondo monitor.

In tal caso approssimiamo con il punto ad esso più vicino del cubo unitario.

## 18 La trasformazione di coordinate per un monitor

Calcoliamo ora la matrice J di trasformazione da R, G, B al modello CIE per un dato monitor.

Come visto nella (5) basta determinare le coordinate CIE dell'emissione dei suoi fosfori.

Per questo scriviamo  $E_r$  per il contenuto cromatico del rosso  $E_r = X_r + Y_r + Z_r$ , ed usiamo le relazioni (1), (2), (3) per passare alle coordinate di cromaticità:

$$x_r = \frac{X_r}{E_r} \tag{7}$$

$$y_r = \frac{Y_r}{E_r} \tag{8}$$

$$z_r = \frac{Z_r}{E_r} = \frac{1 - x_r - y_r}{E_r} \,. \tag{9}$$

Da qui si ha

$$X_r = x_r E_r$$
$$Y_r = y_r E_r$$
$$Z_r = z_r E_r = (1 - x_r - y_r) E_r.$$

Si definiscono analogamente  $E_g$  ed  $E_b$ . Perciò la matrice J nella (5) diventa

$$J = \begin{pmatrix} x_r E_r & x_g E_g & x_b E_b \\ y_r E_r & y_g E_g & y_b E_b \\ (1 - x_r - y_r) E_r & (1 - x_g - y_g) E_g & (1 - x_b - y_b) E_b \end{pmatrix}$$
(10)

Le coordinate di cromaticità dei fosfori sono di solito specificate dal costruttore del monitor; altrimenti, esse si possono misurare con un colorimetro. Per trovare J basta quindi determinare  $E_r$ ,  $E_g$ ,  $E_b$ .

Per far questo si può procedere in uno dei due modi seguenti:

1. Si misura con un fotometro di alta qualità la luminosità massima  $Y_r$ ,  $Y_g$ ,  $Y_b$  dell'emissione dei fosfori (cioè dei colori rosso, verde e blu più intensi emessi dal monitor; la qualità dello strumento deve essere sufficientemente alta da garantire buona precisione anche per il blu, colore per il quale la sensibilità è minore). Così dalle (7), (8), (9) si ricava

$$E_r = \frac{Y_r}{y_r}$$
$$E_g = \frac{Y_g}{y_g}$$
$$E_b = \frac{Y_b}{y_b}$$

Abbiamo già osservato che le cromaticità  $y_r$ ,  $y_g$ ,  $y_b$  dei fosfori sono fornite dal costruttore del monitor o misurabili: comunque sia possiamo assumerle note, e quindi dalle equazioni precedenti ricaviamo  $E_r$ ,  $E_g$ ,  $E_b$ .

2. Si misurano le coordinate  $X_w$ ,  $Y_w$ ,  $Z_w$  del punto di bianco (cioè della luce bianca più intensa) del monitor. Esse corrispondono alle coordinate (R, G, B) = (1, 1, 1). Perciò da (5) e da (10) si ottiene:

$$\begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix} = \begin{pmatrix} x_r & x_g & x_b \\ y_r & y_g & y_b \\ 1 - x_r - y_r & 1 - x_g - y_g & 1 - x_b - y_b \end{pmatrix} \begin{pmatrix} E_r \\ E_g \\ E_b \end{pmatrix}$$
(11)

In questo sistema le incognite sono  $E_r$ ,  $E_g$ ,  $E_b$  mentre  $X_w$ ,  $Y_w$ ,  $Z_w$  sono costanti note. Risolvendo il sistema si trovano  $E_r$ ,  $E_g$ ,  $E_b$ .

Il secondo metodo è più preciso, perché misurare le coordinate  $X_w$ ,  $Y_w$ ,  $Z_w$  del punto di bianco è più preciso che misurare le coordinate di cromaticità dei colori puri rosso, verde e blu (come già osservato, ci vuole un fotometro di alta qualità per misurare con precisione le coordinate del blu, dove la sensibilità è minore).

# **SECONDA PARTE**

# ALIASING

## CAMPIONAMENTO DI SEGNALI, ANALISI SPETTRALE, RICOSTRUZIONE ED ALIASING

#### CAMPIONAMENTO

La teoria del campionamento descrive le relazioni tra un segnale continuo e i suoi campioni, cioè i suoi valori ad intervalli fissi della variabile. Consideriamo segnali di una sola variabile, a valori reali; se il segnale è un'immagine, questo vuol dire che ne considereremo una riga alla volta. Chiamiamo variabile spaziale la variabile del segnale: nel caso si tratti di una riga di pixel di un'immagine, la variabile spaziale descrive la posizione del centro di ciascun pixel nella riga (ed il valore del segnale consiste nell'intensità di illuminazione del pixel); se invece si tratta di un segnale acustico, la variabile spaziale è in realtà il tempo (ed il valore del segnale è l'intensità sonora al variare del tempo). Finora abbiamo considerato segnali nel dominio dello spazio; la loro trasformata di Fourier ci permette di rappresentarli nel dominio della frequenza: in tal caso il segnale viene scomposto come somma od integrale di onde sinusoidali.



Fig.1. Segnali periodici e loro approssimanti di Fourier

I segnali periodici, come quelli mostrati in Fig.1, possono essere rappresentati come la somma di funzioni sinusoidali sfasate le cui frequenze sono multipli interi (armoniche) della frequenza fondamentale del segnale (che è il reciproco del

(a)

periodo). Nel caso di segnali non periodici, la trasformata di Fourier si applica a segnali definiti su tutta la retta reale, quindi di durata, o lunghezza, infinita. In realtà i segnali che considereremo hanno lunghezza o durata finita: ad esempio, un'immagine consiste di righe di un numero finito di pixel. Ma possiamo ugualmente applicare la trasformata di Fourier prolungando il segnale al di fuori del suo intervallo di definizione, ponendolo uguale a zero al di fuori. Allora la trasformata di Fourier ci dà l'ampiezza di ciascuna onda sinusoidale che contribuisce alla scomposizione del segnale. Talvolta la trasformata di Fourier di un segnale si chiama il suo *spettro di Fourier*, o spettro di frequenze. Nel caso di un segnale non periodico, lo spettro di frequenze non consiste di multipli interi di qualche frequenza fondamentale: può contenere qualsiasi frequenza. La ricostruzione del segnale originale a partire dal suo spettro di Fourier è quindi ottenuta non dalla somma di funzioni sinusoidali, bensì da un integrale, che si chiama trasformata inversa di Fourier.

Alternativamente, un segnale di lunghezza finita L può essere prolungato a tutta la retta per periodicità. Supponiamo ad esempio che il segnale abbia supporto in un intervallo di lunghezza L, e si annulli ai bordi dell'intervallo. Dopo la periodicizzazione, il segnale originale diventa un periodo del segnale prolungato. In tal caso il suo spettro di Fourier è discreto: le frequenze che vi compaiono sono i multipli di 1/L. Però questo spettro discreto consiste nella restrizione a questa successione discreta di frequenze della trasformata di Fourier del segnale originale (inteso come prolungato a zero a tutta la retta reale). Quindi questo punto di vista discreto alternativo è una interpolazione di quello continuo: i valori dello spettro di solito ha un salto), e, se si prende il periodo dove il segnale periodicizzato di solito ha un salto), e, se si prende il periodo L via via più grande, questi valori approssimano sempre meglio quelli dello spettro continuo. *Per maggiori dettagli si veda il corso di Analisi Armonica, capitoli 5 e 11, ed i cenni seguenti.* 

Qui ci limitiamo a rammentare alcune definizioni e le loro interrelazioni, cominciando con la definizione di trasformata di Fourier di un segnale *f*, che indichiamo con *F*:

$$F(u) = \int_{-\infty}^{\infty} f(x) (\cos(2\pi u x) - i \sin(2\pi u x)) dx$$

ovvero, in termini di esponenziali complessi,

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i u x} dx$$

dove *i* è l'unità immaginaria.

Anche se stiamo limitando l'attenzione a segnali a valori reali, per ogni frequenza u il valore F(u) è un numero complesso, che possiamo scrivere come R(u)+ i l(u)

dove R(u) e I(u) sono rispettivamente la sua parte reale e la sua parte immaginaria. L'ampiezza di F(u) è data da

$$\left|F(u)\right| = \sqrt{R^2(u) + I^2(u)}$$

e lo sfasamento (anche conosciuto come angolo di fase) è dato da

$$\phi(u) = \operatorname{arctg}\left(\left|\frac{I(u)}{R(u)}\right|\right)$$

Rammentiamo anche la definizione di trasformata inversa di Fourier, che ci permette di trasformare una funzione integrabile F(u) dal dominio della frequenza al dominio dello spazio, nel modo seguente:

$$f(x) = \int_{-\infty}^{\infty} F(u) (\cos(2\pi u x) + i \sin(2\pi u x)) dx$$

ovvero

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{2\pi i u x} dx$$

Spesso, quando si disegna il grafico di una trasformata di Fourier, si disegna solo il grafico della sua ampiezza rispetto alla frequenza, ignorando l'angolo di fase (perché altrimenti il grafico dovrebbe essere disegnato come una curva in tre dimensioni, una per la variabile reale spaziale e due per rappresentare i valori complessi della trasformata). La Fig.2 mostra diversi segnali in entrambi i domini.



Fig.2. Esempi di segnali e loro trasformate di Fourier

In questi grafici, come già osservato prima, i numeri sull'asse delle ascisse rappresentano la posizione dei centri dei pixel; nel dominio della frequenza invece i numeri sull'asse delle ascisse misurano i cicli per pixel (o meglio, i cicli per un intervallo spaziale pari alla distanza fra i centri di due pixel consecutivi). La componente a frequenza u=0 rappresenta la componente dello spettro a frequenza continua (cioè l'integrale del segnale originale, a meno di normalizzazione).

Poiché i pixel sono discreti, è naturale considerare la variante alternativa "discreta" dello spettro di Fourier dei segnali (quella esposta più sopra quando si è parlato di periodicizzazione). Esiste una versione discreta della definizione di trasformata di Fourier che è adatta per questo scopo, la trasformata discreta di Fourier, che introduciamo alla pagina seguente (per maggiori dettagli si veda il <u>corso di Analisi Armonica</u>, capitolo 11).

La trasformata discreta di Fourier è definita da

$$F(n) = \sum_{k=0}^{N-1} f(k) \left( \cos\left(\frac{2\pi nk}{N}\right) - i \sin\left(\frac{2\pi nk}{N}\right) \right) = \sum_{k=0}^{N} \exp\left(-\frac{2\pi i nk}{N}\right), \text{ dove}$$
$$0 \le n \le N-1$$

e la trasformata discreta di Fourier inversa è

$$f(k) = \sum_{k=0}^{N-1} F(n) \left( \cos\left(\frac{2\pi nk}{N}\right) + i \sin\left(\frac{2\pi nk}{N}\right) \right) = \sum_{k=0}^{N} \exp\left(\frac{2\pi i nk}{N}\right), \text{ dove } 0 \le k \le N-1$$

In queste formule, il segnale viene considerato come un segnale periodico nel periodo [0, N], campionato a passo 1: a meno di cambiamento di scala, lo si può quindi immaginare come un segnale periodico di periodo 1 campionato a passi di 1/N, o più in generale di periodo L campionato a passi di L/N. E' facile mostrare che, se scegliamo un passo di campionamento sufficientemente piccolo, cioè N sufficientemente grande, si ottiene una buona approssimazione del comportamento della trasformata continua di Fourier per la maggior parte dei segnali.

Con la trasformata discreta di Fourier si ottiene sempre uno spettro finito. Equivalentemente, grazie alla periodicizzazione del segnale *f*, si possono definire la trasformata discreta di Fourier e la sua inversa tramite sommatorie da -N/2 a N/2 invece che da 0 a N. Questo accorgimento ci permette di evidenziare con facilità il fatto seguente: se il segnale (che, rammentiamo, si suppone a valori reali) è pari, allora la parte immaginaria della sua trasformata discreta di Fourier è ovunque nulla, perché il contributo di ogni termine sinusoidale per ogni valore della variabile x è cancellato dal suo contributo uguale e di segno opposto per il valore opposto della variabile. Pertanto tutti i termini in seno scompaiono, e rimangono solo quelli in coseno: così la trasformata risulta essere reale e pari. In questi casi, quando disegnamo il grafico della trasformata, non abbiamo bisogno di limitarci a disegnare del suo modulo |F(u)|. Quindi potremo avere grafici che stanno al di sotto dell'asse delle ascisse, in corrispondenza a valori negativi di F(u).

Supponiamo di considerare segnali il cui spettro di Fourier sia limitato in banda, cioè nullo al di fuori di un intervallo di frequenze. Poiché consideriamo segnali reali, la trasformata di Fourier ha parte reale pari e parte immaginaria dispari, come segue immediatamente dalle formule viste sopra (per maggiori dettagli si veda il corso di Analisi Armonica, capitolo 5, sezione 2). Supponiamo allora che il

segnale abbia spettro di Fourier nullo al di fuori di un intervallo di *frequenze* [- $\omega_c$ ,  $\omega_c$ ].

Scegliamo la frequenza  $\omega_c$  come la più piccola possibile con questa proprietà .La parità ci consente di limitare l'attenzione alle frequenze positive, cioè all'intervallo [0,  $\omega_c$ ], cosa che faremo sistematicamente in tutti i grafici. Diremo allora che  $\omega_c$  è la frequenza di taglio (o di Nyquist) del segnale, o alternativamente che esso ha banda limitata da  $\omega_c$ .

Consideriamo un ciclo di un segnale la cui componente a frequenza più alta è alla frequenza  $2\pi\omega_c$ . Questa componente è una funzione sinusoidale con  $\omega_c$  punti di massimo e  $\omega_c$  punti di minimo in ciascun periodo, come mostrato in Fig.3.



Fig.3. Una funzione sinusoidale può avere campioni nulli alla frequenza di Nyquist, a seconda della sua fase

Quindi è intuitivo che siano necessari almeno  $2\omega_c$  campioni per ottenere una buona approssimazione della forma della componente della frequenza più alta del segnale. Si noti però che  $2\omega_c$  campioni bastano a dare una idea precisa della forma di questa componente solo se i campioni sono presi esattamente ai punti

delle ascisse corrispondenti ai punti di massimo e di minimo (Fig.3(a)). Se essi sono presi altrove allora l'ampiezza non viene rappresentata correttamente (Fig.3(b)) e può addirittura essere identicamente zero se i campioni sono presi nei punti intermedi fra i massimi ed i minimi (Fig.3(c)). Se campioniamo sotto la frequenza di Nyquist, allora i valori campionati che otteniamo per la componente di frequenza massima sono gli stessi che avremmo ottenuto campionando il segnale ad una frequenza più bassa, come mostrato in Fig.4. (*Per maggiori dettagli si veda il corso di Analisi Armonica, capitolo 10, sezione 5.*)



Fig.4. Una funzione sinusoidale campionata ad una frequenza inferiore a quella di Nyquist ha gli stessi campioni di una sinusoide a frequenza inferiore (*aliasing*)

Questo esempio mostra che alte frequenze nel segnale originale viste ad un passo di campionamento insufficiente si presentano come basse frequenze nel segnale ricostruito: tale fenomeno è conosciuto come aliasing.

Se un segnale, ad esempio un'immagine, viene campionato ad una frequenza insufficiente, allora, dopo che si ricostruisce il segnale a partire da tali campioni, componenti di alta frequenza del segnale originale appaiono alla stregua di componenti di bassa frequenza nel segnale ricostruito a partire da questi campioni inadeguati. Questo è il tipico esempio di aliasing.

Un esempio di aliasing per un'immagine è mostrato in Fig.5.



Fig.5. Aliasing nel campionamento di bande di grigio di ampiezza via via minore

La Fig.5(a) è l'immagine originale accompagnata dal grafico della sua luminosità lungo una retta orizzontale: la luminosità ha salti di ampiezza costante ma di lunghezza via via piu piccola: in altre parole, la frequenza spaziale aumenta da sinistra a destra. La Fig.5(b) è ottenuta scegliendo un pixel ogni otto e ripetendo otto volte il valore di intensità su quel pixel: questo corrisponde ad un campionamento a frequenza otto volte minore. Gli effetti di aliasing sono evidenti: le bande non diventano via via più strette con regolarità, il loro spessore fluttua.

La forma di un segnale è determinata dal suo spettro di frequenza. Quanto più l'andamento del segnale presenta punti angolosi ed oscillazioni brusche, tanto più elevate sono le componenti ad alta frequenza; i segnali con discontinuità hanno uno spettro in frequenza a supporto infinito. La Fig.6 rappresenta il campionamento di una scena consistente di triangoli, con punti di campionamento su un reticolo.



Fig.6. Intere figure possono essere omesse in un campionamento puntuale a passo inadeguato

La scena consiste di triangoli scuri su sfondo bianco. Al bordo di questi triangoli l'intensità luminosa ha un salto, quindi lo spettro di frequenza non è a banda limitata, e la nostra presentazione euristica del teorema del campionamento
rivela che nessun passo di campionamento è sufficientemente piccolo da permettere una ricostruzione esatta di questa immagine. Ciò che si ottiene nella ricostruzione è una scalettatura a passo uguale a quello del campionamento, perché un punto di campionamento o è dentro o e fuori di un triangolo, ma non può rappresentare un valore intermedio e graduale di appartenenza al triangolo.

La scalettatura è una delle due manifestazioni tipiche di fenomeni analoghi all'aliasing nella grafica al computer. L'altra manifestazione tipica si ha nel caso di tessiture ed oggetti con una naturale periodicità quando sono visti in prospettiva: ad esempio, un pavimento con piastrelle regolari. Quando le piastrelle sono proiettate sul piano prospettico, diventano via via più piccole all'aumentare della loro distanza dall'osservatore. Un campionamento a passo regolare quindi salta alternativamente intere righe o bande di piastrelle, esattamente come avviene in Fig.5(b). La conseguenza è una deformazione periodica e molto evidente del pattern della piastrellatura o tessitura, con effetti qualitativi più pesanti che non siano quelli della scalettatura.

# FILTRAGGIO

Sulla scia delle tecniche legate al campionamento, ora mostriamo come creare un nuovo segnale filtrato rimuovendo le alte frequenze che contengono il rumore, cioè disturbo. Tronchiamo lo spettro di frequenza del segnale al di sopra di una data frequenza di taglio (chiameremo questa operazione prefiltraggio). Il nuovo segnale, essendo a banda limitata, è ricostruibile senza perdite a partire da un numero finito di campioni, per il teorema del campionamento (per la cui dimostrazione si veda il corso di Analisi Armonica, capitolo 7, sezione 4, e capitolo 10, sezione 1), in base al quale la frequenza di campionamento necessaria per ricostruire il segnale senza perdite deve essere almeno il doppio della frequenza di taglio. **Cautela:** ciò che viene ricostruito senza perdite non è il segnale originale, ma quello prefiltrato a banda di frequenza limitata.

Perciò, quanto più bassa è la frequenza di taglio  $\omega_c$ , tanto minore è la frequenza di campionamento necessaria per ricostruire esattamente il segnale filtrato e quindi il numero di campioni utilizzati, ma meno il segnale assomiglia all'originale (è più addolcito, con picchi e spigoli più smussati).

Tagliare la banda spettrale del segnale significa applicare un filtro passa basso (a frequenza  $\omega_c$ ). Le alte frequenze vengono annullate dal filtro, le basse frequenze restano. Il filtro passa basso causa sfocatura (*blurring*) nel dominio dello spazio, poiché elimina i dettagli fini, che sono immagazzinati nelle componenti di alta frequenza (Fig.7).



Fig.7. Immagine filtrata con un filtro passa-basso nel dominio della frequenza: il contrasto diminuisce

Ogni filtro, per definizione, agisce sulla trasformata di Fourier del segnale come la moltiplicazione puntuale per una funzione della variabile frequenza, che chiameremo la funzione filtro. Nel dominio dello spazio, la moltiplicazione per la funzione filtro diventa la convoluzione per la sua antitrasformata di Fourier. *Per maggiori dettagli si veda il corso di Analisi Armonica, capitolo 5, sezione 3.* 

Consideriamo allora il caso del filtro passa basso a frequenza di taglio  $\omega_c$ . Nel dominio della frequenza, la sua funzione filtro vale zero al di fuori dell'intervallo [- $\omega_c$ ,  $\omega_c$ ], e vale 1 all'interno di questo intervallo, perché all'interno il filtro non deve cambiare niente, e quindi deve moltiplicare per 1. Si dice che il suo grafico è un'onda quadra. Nel dominio dello spazio questo filtro passa basso agisce come l'operatore di convoluzione con un peso oscillante di tipo *sinc*, e più precisamente come *sinc*( $2\omega_c x$ ), dove *sinc*(x) = *sin*( $\pi x$ )/ $\pi x$ .

La funzione *sinc* ha supporto illimitato, ma noi abbiamo bisogno di limitare l'attenzione ad un numero finito di campioni, affinché il procedimento numerico abbia termine (ovviamente il tempo di elaborazione cresce col numero di campioni utilizzati). Per questo dobbiamo troncare il filtro *sinc*. Purtroppo, il troncamento di una funzione genera discontinuità a salto, che, in virtù del

fenomeno di Gibbs (<u>corso di Analisi Armonica</u>, capitolo 3, sezione 18), l'approssimazione di Fourier trasforma in fluttuazioni oscillatorie spurie in prossimità del salto, cioè della frequenza di taglio (ringing). Per evitare questi effetti si modifica il tipo di troncamento del filtro, in modo che la funzione filtro non abbia salti. Ad esempio si può far uso di una funzione filtro col grafico triangolare anziché rettangolare (una tale funzione è la convoluzione di un'onda quadra con sé stessa, quindi la sua antitrasformata di Fourier é sinc<sup>2</sup>). Oppure si può utilizzare una funzione filtro Gaussiana (<u>corso di Analisi Armonica</u>, capitolo 5, sezione 4), che non è a supporto compatto, ma decresce così rapidamente che agli effetti numerici si può considerare tale. Questi tipi di filtraggio portano ad approssimazioni numeriche accettabili sia nell'approssimazione del segnale, sia nel tempo di calcolo.

## **RICOSTRUZIONE ED ALIASING**

Supponiamo che il segnale *f* abbia spettro in frequenza (cioè trasformata di Fourier) a banda limitata (cioè che lo spettro sia nullo sopra una frequenza che indicheremo con  $\omega_c$ ), e campioniamo il segnale *f* a frequenza  $\omega_s = 2 \omega_c$ . Per comodità, anche se non è necessario, supponiamo che il segnale *f* sia a valori reali (se cosi' non è possiamo applicare quanto segue separatamente alla parte reale ed a quella immaginaria di *f*). Allora la trasformata di Fourier di *f* (che è a valori complessi) ha parte reale pari, parte immaginaria dispari, ed entrambe sono nulle al di fuori dell'intervallo [- $\omega_c$ ,  $\omega_c$ ], che ha lunghezza 2  $\omega_c$ =  $\omega_s$ . Chiamiamo  $\phi$  il segnale campionato. Dalla teoria del campionamento segue che lo spettro in frequenza di  $\phi$  si ottiene periodicizzando quello di *f*, cioè replicandolo a passo  $\omega_s$ .

Sia x un valore fissato della variabile spaziale (in altre parole, x rappresenta un pixel se il segnale è l'intensità di colore in una riga di pixel di un'immagine fotografica, oppure un tempo se il segnale è un segnale musicale). Denotiamo con  $\delta_x$  la distribuzione impulsiva (*delta di Dirac*) centrata al punto x. Rammentiamo che se si moltiplica la distribuzione  $\delta_x$  per la funzione f, il risultato è il multiplo di  $\delta_x$  dato dalla moltiplicazione per il numero f(x): cioè  $f \delta_x = f(x) \delta_x$ (per maggiori dettagli si veda il corso di Analisi Armonica, capitolo 8). Da questo fatto segue immediatamente che campionare un segnale equivale a moltiplicarlo nel dominio dello spazio con una distribuzione treno di impulsi (comb): la distribuzione treno di impulsi è la serie delle delta di Dirac centrate ai punti del campionamento. In altre parole, questa distribuzione differisce dalla funzione zero solo ai punti del campionamento, e la si può approssimare con una serie di funzioni impulsive -in inglese spike- (cioé di funzioni non negative, di integrale 1 e col grafico "alto e stretto"). I termini della serie approssimante sono traslati equispaziati (di passo dato dal passo di campionamento 1/(2mc) di una funzione spike  $h^{\tau}$ , maggiore o uguale a zero, infinitamente derivabile, di integrale 1 e tale che, per ogni intervallo arbitrariamente piccolo centrato ad uno dei punti del campionamento e per ogni  $\varepsilon$  positivo arbitrario, esiste un intervallo di ampiezza  $\tau$ tale che l'integrale della corrispondente funzione spike  $h^{\tau}$  vale su quell'intervallo più di  $1 - \varepsilon$  purché  $\tau$  sia sufficientemente piccolo. La serie approssima il treno di impulsi al tendere di  $\tau$  a 0.

Rammentiamo qui alcuni risultati dimostrati nel <u>corso di Analisi Armonica</u>. La trasformata discreta di Fourier di una distribuzione treno di impulsi con i "denti"

(cioé le delta di Dirac che la compongono) spaziati di passo  $\tau$  è un altro treno di impulsi con i denti spaziati di 1/ $\tau$  (Fig. 8(a)).



Fig.8. (a) Il treno di impulsi e la sua trasformata di Fourier; (b) trasformata di Fourier di un segnale campionato.

Questo segue da due risultati preliminari: il fatto che il treno di impulsi di passo 1 coincide con la sua trasformata di Fourier nel senso delle distribuzioni (formula di somma di Poisson, corso di Analisi Armonica, capitolo 7, sezione 3 e capitolo 9, sezione 2), e la regola di dilatazione per la trasformata di Fourier (corso di Analisi Armonica, capitolo 5, sezione 2). Poiché la moltiplicazione nel dominio del tempo corrisponde a una convoluzione nel dominio della freguenza, la trasformata di Fourier del segnale campionato è la convoluzione della trasformata di Fourier della distribuzione treno di impulsi con il segnale originale. Il risultato è la replica iterata dello spettro di Fourier originale, cioè una successione di copie replicate e traslate dello spettro originale. Infatti, come visto sopra, ciascun campione corrisponde a moltiplicare il segnale originario per una delta di Dirac centrata al'istante di campionamento: quindi la successione dei campioni corrisponde alla moltiplicazione del segnale originale per una serie di delta di Dirac equispaziate (un treno di impulsi nel dominio dello spazio). Ma abbiamo detto che la trasformata di Fourier di questo treno di impulsi spaziali è un altro treno di impulsi nel dominio della frequenza, via via più radi quanto più fitti sono quelli originali nel dominio dello spazio, quindi una serie di traslati di delta di Dirac. La moltiplicazione col treno di impulsi nel dominio dello spazio corrisponde alla convoluzione con questo nuovo treno di impulsi nel dominio della frequenza, cioè con questa successione di delta equispaziate. Ma grazie ad un altro risultato del *corso di Analisi Armonica, capitolo 8, sezione 13*, la convoluzione con una delta di Dirac centrata ad una frequenza, diciamo,  $\omega_0$  è una traslazione di passo  $\omega_0$ . Pertanto la trasformata di Fourier della successione dei campioni di passo  $\tau$  è una somma di traslati di passo 1/  $\tau$  della trasformata di Fourier del segnale originale. Se questa trasformata aveva supporto da -  $\omega_c$  a  $\omega_c$  ed il passo di campionamento è minore o uguale di  $1/\omega_s = 1/(2 \omega_c)$ , allora queste copie replicate sono separate fra loro di almeno  $2 \omega_c$ , cioè la lunghezza del loro supporto, e quindi non si sovrappongono. In tal caso la serie che ci dà la trasformata di Fourier della successione dei campioni non è altro che una successione di repliche a supporto disgiunto dello spettro originario. Questo fatto è illustrato nella Figura 8(b).

Allora come si fa a ricostruire il segnale originale a partire dai suoi campioni? Abbiamo visto che il risultato del campionamento di un segnale (Fig.9(a)) ad una frequenza di campionamento finita è un segnale con uno spettro di frequenze a supporto illimitato (Fig.9(b)). Se analizziamo nel dominio della frequenza questo segnale campionato, vi troviamo una successione di copie replicate e traslate dello spettro originale. La moltiplicazione in questo dominio della trasformata di Fourier del segnale per una funzione tipo onda quadra (cioè un filtro passa basso, che lascia invariato lo spettro di Fourier nell'intervallo da -  $\omega_c$  a  $\omega_c$  e lo tronca a 0 al di fuori) elimina queste copie replicate dello spettro (Fig.9(c)), lasciando solo una singola copia dello spettro originale (Fig.9(d)), dalla quale, grazie alla trasformata inversa di Fourier, si ricostruisce senza perdite il segnale originale. In altre parole, se il segnale ha ampiezza di banda limitata, lo possiamo ricostruire esattamente a partire dai suoi campioni moltiplicando la trasformata di Fourier della successione dei campioni con un opportuno filtro passa basso nel dominio della frequenza, o equivalentemente, convolvendo i campioni nel dominio dello spazio con la trasformata di Fourier di questo filtro, la quale è sinc(Ax) con  $A = 2 \omega_c = \omega_s$  (Teorema del campionamento: si veda il corso di Analisi Armonica, capitolo 7, sezione 4 e capitolo 10, sezione 1)



Fig.9. Campionamento adeguato: nessuna sovrapposizione nel dominio della frequenza e ricostruzione senza perdite

In effetti, il ricorrere al dominio della frequenza è necessario solo per dimostrare la validità del procedimento: il calcolo numerico avviene sempre soltanto nel dominio dello spazio, e si svolge convolvendo con i valori della funzione *sinc* (o altri filtri simili), perché l'effettuare i calcolo esplicito delle trasformate di Fourier sarebbe di gran lunga troppo oneroso dal punto di vista numerico (ogni singolo valore richiederebbe il calcolo numerico di un integrale sull'intera retta reale). E' comunque istruttivo esaminare alcuni segnali filtrati con vari filtri ed i loro spettri di Fourier, cosa che facciamo ora.

Premettiamo che, nel disegnare la trasformata di Fourier nei disegni delle Fig.9 – 11, abbiamo fatto una semplificazione inaccurata: i filtri nel dominio della

frequenza non sono stati disegnati all'altezza giusta. La loro altezza dovrebbe essere 1 in Fig.9 e 2 in Figg.10 - 11 per restituire la copia originale dello spettro con la grandezza che aveva all'inizio. Questo segue dal fattore di normalizzazione della proprietà di dilatazione della trasformata di Fourier (si veda il <u>corso di Analisi Armonica</u>, capitoli 5 e 8).

Il filtro passa basso (cioè con grafico rettangolare) che si utilizza in questo procedimento di filtraggio consente una ricostruzione esatta del segnale originale perché nel suo supporto esso lascia invariati i valori dello spettro originale. Se si usano filtri di tipo diverso, il segnale ricostruito non è esattamente uguale all'originale. A titolo di esempio, consideriamo il caso di un filtro il cui grafico è di forma triangolare (cioè dato dalla convoluzione di un filtro passa basso per sé stesso). Le Figg. 9(e) e 9(f) mostrano il risultato della ricostruzione dei campioni con un filtro triangolare. E' immediato verificare che la convoluzione con questo filtro è una interpolazione lineare dei campioni.

Se la frequenza di campionamento è troppo bassa, si ha un fenomeno chiamato *aliasing*: le copie replicate degli spettri in frequenza si sovrappongono, come in Fig.10.



Fig.10. Campionamento inadeguato: ricostruzione con aliasing

In questo caso, il filtro passa basso del processo di ricostruzione non può rimuovere quelle parti iniziale e finale degli spettri replicati adiacenti a quello originale le quali si sovrappongono allo spettro del segnale originale. Quindi componenti ad alta frequenza degli spettri replicati sono mescolate con le componenti a bassa frequenza del segnale originale, e sono trattati come basse frequenze (cioè non eliminate dal filtro) durante il processo di ricostruzione. Pertanto un passo di campionamento inadeguato causa l'aliasing: alcune componenti spettrali a frequenze che sono alte (cioè da eliminare) prima della ricostruzione si mascherano come frequenze basse dopo. Ci sono due vie per risolvere questo problema. Possiamo scegliere di campionare una frequenza sufficientemente alta, un approccio possibile solo se il segnale ha una ampiezza di banda finita; oppure possiamo filtrare il segnale prima del campionamento,

rimuovendo tutte le componenti sopra  $\omega_c$  come mostrato in Fig.11. Osserviamo che quest'ultimo approccio ha una fase iniziale di prefiltraggio: infatti, esso consiste nel filtrare prima lo spettro di Fourier del segnale originale con un opportuno filtro passa basso, per trasformare tale segnale in un nuovo segnale a banda spettrale limitata (il quale appare quindi addolcito rispetto all'originale), e dopo il filtraggio campionare il nuovo segnale addolcito alla frequenza di campionamento appropriata (quella data dal doppio della frequenza di taglio del filtro). Questi campioni rappresentano la versione digitalizzata del segnale: da essi si può ricostruire il segnale, ma la ricostruzione non ci dà il segnale originale, bensì quello addolcito. Osserviamo ancora che il prefiltraggio nel dominio della frequenza equivale ad una convoluzione con una *sinc* nel dominio dello spazio.



Fig.11. Prefiltraggio passa-basso per rendere adeguato il campionamento

Mettiamo però in guardia il lettore sul fatto che nell'hardware video la ricostruzione è fatta convolvendo il segnale con una funzione diversa dalla *sinc* 

(e quindi non esatta). Infatti i campioni nel frame buffer vengono trasformati in un segnale video continuo, mediante un processo di sample and hold (campionamento con mantenimento): se il segnale che stiamo ricostruendo è una riga di un'immagine, il valore di ogni campione viene mantenuto per un intervallo spaziale pari ad un pixel. Questo processo corrisponde a convolvere la successione dei campioni con un filtro passa-banda largo un pixel, come mostrato in Fig. 12, e fa pensare ai pixel come se fossero tessere quadrate che coprono il display. Il segnale risultante presenta salti nel passare da un pixel al successivo: questi salti corrispondono a componenti ad alta frequenza che i campioni non possedevano, un effetto noto come rastering. Benché l'hardware video dovrebbe campionare e mantenere l'intensità di ogni pixel, i circuiti che generano il voltaggio che pilota il tubo catodico di solito non hanno una risposta abbastanza veloce da mostrare discontinuità nell'intensità tra pixel adiacenti. Anche la distribuzione gaussiana dell'intensità di luce di un fosforo illuminato dal raggio catodico smussa le discontinuità del segnale a campioni mantenuti e quindi aiuta a ridurre questo problema. Quindi il segnale campionato è ricostruito dall'equivalente di una convoluzione con un filtro passa banda, seguito da una convoluzione con una Gaussiana. Il rastering è particolarmente evidente nello zoom, che aumenta l'area allocata a un singolo pixel, e quindi colora quest'area ormai grande di una tinta piatta costante che al bordo del pixel scatta di colpo ad un valore diverso. Il rastering è anche più evidente in stampa, nella registrazione digitale di film, e nei display LCD, che sono tutte tecnologie in cui la transizione di luminosità fra un pixel ed il successivo è molto netta.



Fig.12. Rastering: sample and hold e smussamento gaussiano dei pixel

Abbiamo già visto, nel teorema del campionamento, che un segnale con spettro di Fourier nell'intervallo di frequenze da  $-\omega_c$  a  $\omega_c$  deve essere campionato a una frequenza maggiore o uguale di  $2\omega_c$ , cioè ad un passo minore o uguale di  $1/(2\omega_c)$ , per poter essere ricostruito senza perdite. Ma questo è il caso ottimale, quello che permette la frequenza di campionamento più bassa possibile. Il valore della più bassa frequenza di campionamento accettabile dipende dal filtro che si vuole usare per la ricostruzione, ed  $1/(2\omega_c)$  è ciò che si ottiene con la moltiplicazione nel dominio della frequenza col filtro passa banda da  $-\omega_c$  a  $\omega_c$ , cioè con la convoluzione nel dominio dello spazio con una *sinc* non troncata. Se il filtro per ricostruire i campioni è diverso, come avviene sempre quando ricostruiamo un'immagine (perché, quanto meno, si tronca la *sinc* per ridurre i calcoli), allora il segnale ricostruito non è identico all'originale, ed inoltre, per ottenere un risultato comunque accettabile, la frequenza di campionamento deve essere più alta.

Consideriamo, per esempio, una frequenza di campionamento di poco maggiore di  $2\omega_c$  sul segnale  $cos(2\pi \omega_c x)$ . I campioni risultanti si dispongono lungo due archi di un'onda sinusoidale a bassa frequenza, e quindi sembra che il segnale campionato sia un'onda sinusoidale a bassa frequenza, come mostrato in Fig.13. Cioè, con questo passo di campionamento, il segnale originale viene modulato in ampiezza a bassa frequenza. La modulazione a bassa frequenza rimane dopo la ricostruzione (e quindi si ottiene un segnale ricostruito mal modulato) se il segnale è ricostruito con un filtro passa banda largo un pixel (perché in tal caso la ricostruzione si fa tramite il mantenimento dei campioni, che sono modulati a bassa frequenza, per un intervallo lungo un pixel).



Fig.12. Modulazione a bassa frequenza che resta dopo la ricostruzione tramite un filtro passa banda largo un pixel

Invece, se la convoluzione è calcolata con una *sinc* non troncata, il segnale originale viene ricostruito senza errori, per il teorema del campionamento (si veda il calcolo di questo esempio nel <u>corso di Analisi Armonica</u>, capitolo 10, *sezione 5*). Poiché è inevitabile usare filtri troncati (e quindi non ideali) prima e dopo il campionamento, questo significa che si devono utilizzare frequenze di campionamento più elevate per ridurre le perdite.

## RICOSTRUZIONE NUMERICA DEL SEGNALE TRAMITE FILTRAGGIO DEI DATI CAMPIONATI

Il modo grafico in cui abbiamo presentato il teorema del campionamento (per questa dimostrazione si veda il corso di Analisi Armonica, capitolo 10, sezione 1) presuppone che i dati campionati (a frequenza sufficientemente elevata rispetto alla frequenza di Nyquist  $\omega_c$ ) vengano trasformati secondo Fourier, ottenendo in tal modo una successione di repliche dello spettro di Fourier originale. Questa successione viene poi filtrata con un filtro passa basso a freguenza di taglio  $\omega_{c_1}$ ed infine si calcola la trasformata inversa di Fourier. Ma nell'implementazione numerica non c'è bisogno di passare al dominio delle frequenze. Il calcolo si può svolgere direttamente nel dominio dello spazio, dove la moltiplicazione spettrale per il filtro passa basso diventa la convoluzione con una funzione sinc di passo appropriato (il passo è tale che gli zeri della sinc coincidano con i punti di campionamento, eccetto che all'origine; per la rilevanza di questo fatto, si veda l'enunciato del teorema del Campionamento dato nel Corollario 7.4.5 nelle note del corso di Analisi Armonica, capitolo 7, sezione 4, e le successive Note 7.4.6 e 7.4.7. Grazie a questa convoluzione si ottiene la seguente formula di ricostruzione del segnale f ad ogni valore x della variabile spaziale:

$$f(x) = \sum_{k=-\infty}^{\infty} f\left(\frac{k}{2\omega_c}\right) \operatorname{sinc}(2\omega_c x - k)$$

A questo punto è facile capire come si inizia il processo numerico della ricostruzione del segnale. Si inizia con un procedimento di interpolazione a *metà strada* fra due campioni consecutivi. Supponiamo di conoscere i valori campionati (esatti a meno di arrotondamento numerico) ai punti { $k/2\omega_c$  }. Nella prima fase, ricostruiamo i valori esatti del segnale ai punti di mezzo rispetto a punti di campionamento, cioé i punti { $x_k = (k+1/2)/2\omega_c$  }, per tutti gli m interi. Per ottenere questi valori basta svolgere la somma della formula (1) precedente, che richiede l'impiego dei fattori  $sinc(2\omega_c x - k)$ . Dalla scelta di x segue che questi fattori sono precisamente i numeri sinc(k + 1/2), al variare di k sugli interi. Naturalmente, la somma è una serie, e quindi avremmo bisogno di una successione infinita di tali fattori, ma in realtà ci possiamo arrestare quando arriviamo ad una precisione sufficiente. Se ad esempio desideriamo ricostruire il segnale con una precisione numerica di con la terza cifra decimale esatta, potremo ragionevolmente scartare i valori della *sinc* inferiori a  $10^{-4}$ . Infatti, i numeri inferiori a  $0.5x10^{-3}$  sono precisamente quelli il cui arrotondamento ha le

prime tre cifre decimali nulle; consideriamo fattori 5 volte minori per sicurezza, in quanto stiamo trascurando non un unico addendo, ma un'intera coda della serie in (1). Se i valori campionati sono tutti dello stesso segno, allora la serie è a segni alterni, perché i valori di sinc(k + 1/2) sono a segni alterni (il grafico della *sinc* si annulla solo sugli interi non nulli, ed ha valori alternativamente positivi e negativi negli intervalli [k, k+1]). In tal caso, il teorema di Leibnitz per le serie a segni alterni (*si veda il corso di Analisi Armonica, capitolo 1, sezione 1 2*) ci assicura che l'errore commesso nell'arrestare la somma è minore (in modulo) del primo addendo trascurato, e quindi non avremmo bisogno di un ordine di grandezza di vantaggio per tenerci sul sicuro. Purtroppo, però, i valori campionati potrebbero avere segni qualsiasi. Naturalmente, tutte queste stime devono essere riscalate se consideriamo segnali con valori elevati.

Ricordiamo che sinc(x) <  $1/\pi x$ . Quindi, nel nostro caso servono solo valori di x = k + 1/2 tali che  $\pi x < 10^4$ , cioé circa 3500 valori: si tratta di un calcolo assai veloce per un elaboratore elettronico, purché siano stati precedentemente tabulati (una volta per tutte) i valori della sinc a questi punti intermedi. Se invece ci basta la prima cifra decimale, allora sono necessari solo i primi 35 valori intermedi della sinc, che sono riportati per riferimento nella Tabella 10.1 del corso di Analisi Armonica, capitolo 10, sezione 3. Poiché la funzione sinc è pari, è sufficiente riportare in questa tabella i valori della funzione sinc solo per x positivo. Per comprendere come ottimizzare le fasi ulteriori della ricostruzione del segnale (cioé a punti diversi da quelli a meta' strada fra due punti di campionamento consecutivi), dobbiamo ricordare che i valori ricostruiti grazie al procedimento di interpolazione appena descritto sono i valori esatti del segnale, a meno di arrotondamento. Perciò, aggiungendo guesti valori ai campioni di partenza, otteniamo precisamente i valori del segnale originale campionati a frequenza doppia. Allora, se ripetiamo con questa successione raddoppiata di valori il procedimento di interpolazione appena descritto per i valori intermedi, ora otteniamo i valori esatti ai multipli di 1/4. Naturalmente, dobbiamo ora utilizzare una funzione sinc di frequenza raddoppiata (ricordiamo che la sinc deve annullarsi ai punti di campionamento, tranne che in zero). Ma raddoppiare la frequenza significa comprimere il grafico della sinc lungo l'asse delle ascisse: non ne cambia i valori, sposta solamente i punti dove sono raggiunti (dimezzandone la grandezza). Perciò, in questa nuova fase di interpolazione che ha l'obiettivo di ricostruire il segnale ai multipli di 1/4, si usa la stessa tabella di valori della sinc impiegata nella fase precedente. E così via: iterando il procedimento di interpolazione ricostruiamo esattamente il segnale a ciascun valore razionale di x, tranne che per il comporsi degli errori di arrotondamento. Tutte le fasi richiedono sempre la stessa tabella di fattori numerici interpolanti, e quindi il metodo si presta particolarmente bene ad essere implementato in hardware con un unico chip.

### ESERCIZIO SULL'ALIASING

**Esercizio.** Un display ha 1024 pixel in orizzontale, numerati da x=0 a x=1023. Gli inviamo il segnale  $f(x) = [1 + \sin(\pi \sin(\pi x^2/(50))/2)]$  (qui le parentesi quadre rappresentano la parte intera). Pertanto f vale 0 oppure 1 (anche il valore 2 potrebbe essere raggiunto occasionalmente su questa griglia). Quando il segnale f(x) vale 0 il pixel x è buio, quando vale 1 (o anche 2) è illuminato. Per valori di x piccoli il display mostra una alternanza di barre bianche e nere a larghezza variabile in maniera monotona. Ma esiste un indice critico x sopra il quale le barre smettono di comportarsi in questa maniera monotona e sembrano distribuirsi con larghezze casuali. Approssimativamente per quale x questo succede? Perché?

#### Svolgimento.

La funzione sin( $\pi x^2/50$ ) oscilla fra -1 e 1, e raggiunge tali valori nei punti x tali che  $\pi x^2/50 = 3\pi/2 + 2k\pi$  e  $\pi x^2/50 = \pi/2 + 2k\pi$  rispettivamente (qui k è un intero positivo, perché le ascisse variano da 0 a 1024). Ossia x= (75 + 100 k)<sup>1/2</sup> oppure x= (25 + 100 k)<sup>1/2</sup>, rispettivamente. Si osservi che la distanza fra due tali punti consecutivi è

$$\sqrt{75 + 100k} - \sqrt{25 + 100k} = \sqrt{100k} \left( \sqrt{1 + \frac{3}{4k}} - \sqrt{1 + \frac{1}{4k}} \right).$$

In base allo sviluppo di Taylor,

$$(1+y)^{\frac{1}{2}} = 1 + y/2 + O(y^2)$$
 se  $|y| < 1$ .

Quindi

$$\sqrt{1 + \frac{3}{4k}} - \sqrt{1 + \frac{1}{4k}} = \frac{1}{4k} + O\left(\frac{1}{k^2}\right).$$

Ne segue che la distanza fra un minimo ed il successivo massimo è dell'ordine di

$$\frac{\sqrt{100k}}{4k} = \frac{2.5}{\sqrt{k}} \,.$$

Ora, la funzione  $\pi \sin(\pi x^2/(50m))/2$ ) raggiunge, in questi stessi punti e solo in essi, il proprio valore minimo  $-\pi/2$  ed il massimo  $\pi/2$ . La funzione f in questi punti, e solo in essi, raggiunge il proprio minimo -1 ed il proprio massimo 1. Quindi fra questi punti si attua il passaggio da intensità minima (nero) ad intensità massima (bianco) del segnale. Perché la progressione dei campioni neri e bianchi rispetti l'alternanza delle bande nere e bianche del segnale bisogna e basta che in ogni banda cada almeno un punto di campionamento, e quindi che la distanza fra due campioni (il passo di campionamento, pari a 1, si mantenga inferiore alla lunghezza del segmento in cui il segnale passa dall'intensità minima a quella massima, pari a circa  $2.5/\sqrt{k}$ . Quindi la condizione è  $2.5/\sqrt{k} > 1$ , ossia k <  $2.5^2 \sim 6.25$ . Pertanto per k fra 6 e 7 si comincia a perdere la regolarità della progressione delle bande. Il valore di x a cui questo k corrisponde è fra x=  $(25 + 100 \text{ k})^{\frac{1}{2}} = 725^{\frac{1}{2}} \sim 27 \text{ e } x = (75 + 100 \text{ k})^{\frac{1}{2}} = 775^{\frac{1}{2}} \sim 28$ , quindi fra il ventisettesimo ed il ventottesimo pixel.

Invece che usare l'approssimazione di Taylor, avremmo potuto usare l'identità  $(\sqrt{a} - \sqrt{b})(\sqrt{a} + \sqrt{b}) = a - b$ , da cui segue

$$\sqrt{75 + 100k} - \sqrt{25 + 100k} = \frac{75 - 25}{\sqrt{75 + 100k} + \sqrt{25 + 100k}}$$

e quindi

$$\frac{50}{2\sqrt{75+100k}} \le \sqrt{75+100k} - \sqrt{25+100k} \le \frac{50}{\sqrt{75+100k}}$$

Lasciamo al lettore lo sviluppo del calcolo in questa approssimazione.

# **TERZA PARTE**

# PRIMITIVE DI UNA LIBRERIA GRAFICA

# TRACCIAMENTO DI PRIMITIVE GRAFICHE (SCAN CONVERSION)

# ALGORITMI DI SCANSIONE E TRACCIAMENTO (SCAN CONVERSION) PER RETTE

#### Introduzione

Un algoritmo che disegna una retta su un monitor calcola le coordinate dei pixel che sono il più vicino possibile ad una retta idealmente collocata su una griglia 2D. Vogliamo tracciare la retta evidenziando pixel (cioe' punti nodali della griglia) in maniera che siano quanto piu' possibile allineati e vicini alla retta.

In realta' pero', a causa del passo della griglia, la retta tracciata non consistera' di pixel perfettamente allineati, e quindi avra' uno spessore non nullo. Come si devono scegliere i pixel approssimanti se supponiamo vogliamo che lo spesore non sia piu' grande di uno (cioe' del passo della griglia)?

Per le linee con pendenza 45°, ossia con |m|=1, dovrebbe essere illuminato esattamente un pixel in ogni colonna; mentre per quelle linee che si inclinano al di fuori di questa scala dovrebbe essere illuminato esattamente un pixel in ogni riga. Tutte le linee dovrebbero essere disegnate con luminosità costante, indipendentemente dalla lunghezza e dall'orientamento e il più rapidamente possibile. Ci dovrebbero essere anche disposizioni per disegnare linee più spesse di un pixel, centrate su una linea ideale, che sono influenzate dagli attributi di line-style e pen-style (stile della linea e del tratto), che creano altri effetti di cui si ha bisogno per illustrazioni di qualità migliore. Ad esempio, l'illuminazione dei pixel nel punto finale di una linea molto spessa, deve essere a discrezione del disegnatore che potrà scegliere se gli angoli di tale linea dovranno essere: leggermente smussati, del tutto arrotondati o squadrati.



Ci piacerebbe essere in grado di minimizzare le sporgenze dovute alla discreta approssimazione alla linea ideale utilizzando le tecniche di antialiasing, ovvero a come disegnare le entità grafiche minimizzando gli effetti "scaletta" dovuti all'insuffciente campionamento spaziale.

Per adesso consideriamo solo "ottimale" una linea (di spessore 1) che ha esattamente un pixel illuminato in ogni colonna (o fila se parliamo di linee con pendenza |m| < 1). In seguito verranno presi in considerazione i tratti primitivi e gli stili.

Per visualizzare la geometria, il sistema grafico SRGP (Simple Raster Graphics Package) rappresenta un pixel come un punto circolare centrato ai vertici dei quadrati che compongono la griglia. Questa rappresentazione è l'approssimazione conveniente per tutti i sistemi grafici.



In alcuni sistemi grafici i punti adiacenti coincidono, in altri, potrebbe esserci spazio tra pixel verticali adiacenti; nella maggior parte dei sistemi, i pixel si addensano di piu' nella direzione verticale che in quella orizzontale.

Un'altra variazione nella rappresentazione del sistema di coordinate si presenta in sistemi operativi quali ad esempio il Macintosh, i quali vedono i pixel centrati all'interno del box rettangolare adiacente alle linee della griglia (invece che sulle linee della griglia stessa). Noi rappresenteremo i pixel come cerchi disgiunti centrati su una griglia uniforme, benché ci saranno delle piccole variazioni quando si parlerà dell'antialiasing. L'illuminazione dei pixel che compongono una linea, sulla base della linea ideale, è mostrata nella figura seguente (l'immagine è notevolmente ingrandita), dove i punti neri rappresentano i pixel attraversati dalla linea, che verranno illuminati, mentre i punti bianchi rappresentano i pixel vuoti.



Su uno schermo vero, il diametro del pixel approssimativamente circolare è più grande dello spazio che intercorre tra un pixel e l'altro perciò la nostra rappresentazione simbolica esagera sulla distanza tra i pixel.

Poiché le primitive SRGP sono definite in una griglia completa, i punti finali di una linea hanno coordinate intere. Infatti se immaginiamo di tagliare una linea con un poligono (ad esempio un rettangolo), l'intersezione del margine della linea con il rettangolo di taglio, potrebbe avere un punto finale con coordinate non intere. Presupponiamo che la nostra linea abbia un'inclinazione |m|=1; (le linee con altre inclinazioni possono essere trattate con cambiamenti appropriati che verranno spiegati in seguito). Inoltre, le linee più comuni, ossia quelle orizzontali, verticali e quelle che hanno un'inclinazione di 45° (quindi con |m|=±1) potranno essere facilmente trattate perché passano solo attraverso il centro dei pixel.

#### L'algoritmo incrementale di Bressenham

La strategia più semplice per la scansione della linea è quella di calcolare la pendenza m come  $\Delta y / \Delta x$ , incrementare la x di 1 a partire dal punto che si trova più a sinistra, calcolare y<sub>i</sub> = mx<sub>i</sub>+ B per ogni x<sub>i</sub>, e accendere il pixel alla coordinata (x<sub>i</sub>,Round(y<sub>i</sub>)), dove Round (y<sub>i</sub>) = Floor (0,5+ y<sub>i</sub>).

(Ricordiamo che  $\Delta x = x_1 - x_0 e \Delta y = y_1 - y_0$ )

Questo calcolo seleziona il pixel più vicino alla linea ideale, ossia il pixel la cui distanza dalla linea reale è la più piccola.

Questo algoritmo è il più semplice, ma è anche un po' brutale e totalmente inefficiente perché ogni iterazione richiede il calcolo di una somma, di una moltiplicazione e il calcolo del parametro Floor (ossia l'arrotondamento per ogni punto scritto). Un miglioramento dell'algoritmo possiamo ottenerlo rendendo l'algoritmo incrementale, utilizzando la relazione:

 $y_{i+1} = mx_{i+1} + B = m(x_i + \Delta x) + B = y_i + (m \Delta x)$ 

e, se  $\Delta x = 1$  allora

$$y_{i+1} = y_i + m$$

I valori di x e y sono definiti in base ai valori precedenti (si veda la figura qui sotto). Questo è ciò che noi definiamo un algoritmo incrementale: ad ogni passo facciamo calcoli incrementali basati sul passo precedente.



In questo modo per arrivare al risultato si elimina il calcolo della moltiplicazione, anche se è possibile che sorgano problemi dovuti ad errori per l'arrotondamento di m (m è infatti rappresentata con precisione finita e questo fa sì che l'errore si accumuli durante il calcolo); naturalmente i problemi sussisteranno solamente per linee lunghe.

Questo algoritmo viene utilizzato normalmente dai generatori di linee microprogrammati, in una versione leggermente modificata, che esegue i calcoli in aritmetica intera frazionaria anzichè in virgola mobile. *Line* attua la tecnica illustrata nello pseudocodice che segue.

```
void Line (
                                                            /* Assumes -1 \le m \le 1, xo \le x1 */
                                                           /* Left endpoint */
       int x0, int y0,
       int x1, int y1,
                                                           /* Right endpoint */
       int value)
                                                            /* Value to place in line's pixels */
ł
                                                            /* x runs from x0 to x1 in unit increments. */
       int x;
       double dy = y1 - y0;
       double dx = x1 - x0;
       double m = dy / dx;
       double y = y0;
       for (x = x0; x \le x1; x + +)
               WritePixel (x, Round (y), value);
                                                           /* Set pixel to value */
                                                           /* Step y by slope m */
               v + = m:
       }
       /* Line */
```

## L'algoritmo del punto di mezzo

Un notevole incremento delle prestazioni rispetto all'algoritmo precedente si può ottenere con un algoritmo che prevede solamente l'utilizzo della matematica intera; l'algoritmo fu sviluppato da Bresenham in *[J.E. Bresenham, <u>Algorithm for Computer Control of a Digital Plotter</u>, <i>IBM Systems Journal 4 (1965), 25-30]*, e prevede l'utilizzo delle sole operazioni di somma e confronto. In particolare, esso evita la funzione di arrotondamento e calcola (x<sub>i+1</sub>, y<sub>i+1</sub>) a partire dai parametri (x<sub>i</sub>, y<sub>i</sub>) in maniera incrementale.

La versione di questo algoritmo per i punti non fissi può essere applicata alle linee le cui coordinate del punto finale hanno valori reali scelti arbitrariamente. Inoltre, la tecnica incrementale di Bresenham può essere anche applicata al calcolo intero dei cerchi, anche se non è facilmente applicabile alle coniche. Usiamo un metodo leggermente diverso, la tecnica del punto di mezzo (*midpoint technique*) pubblicata per la prima volta in [*M.L.V. Pitteway*, *Algorithm for Drawing Ellipses on Hyperbolae with a Digital Plotter, Computer Journ. 10* (1967), 282-289]. e adattata da Van Aken [*J.R. Van Aken, An Efficient Ellipse-Drawing Algorithm, Computer Graphics and Appl. 4* (1984), 24-35] e da altriricercatori. Per le linee e i cerchi interi, la formula del punto di mezzo, come mostrato in [*J.R. Van Aken, M. Novak, Curve-Drawing Algorithms for Raster Displays, ACM T rans. On Graphics 4* (1985), 147-169], si riduce alla formula di Bresenham poichè genera gli stessi pixel. Bresenham dimostrò che il suo algoritmo per le linee ed i cerchi discretizzati fornisce l'approssimazione migliore ai cerchi ed alle linee reali, minimizzando l'errore (ovvero la distanza) dalla primitiva reale [*J.E. Bresenham, A Linear Algorithm for Incremental Digital Display of Circular Arcs, Comm. ACM 20 (1977), 100-106*].

Supponiamo che la pendenza della linea sia compresa tra 0 ed 1 (0=m=1). Le altre pendenze possono essere calcolate con riflessioni rispetto agli assi principali. Chiamiamo il punto finale più basso a sinistra ( $x_0$ ,  $y_0$ ) ed il punto finale più in alto di destra ( $x_1$ , $y_1$ ).



Consideriamo la linea della figura qui sopra, dove il pixel selezionato precedentemente appare come un cerchio nero, e i due pixel dai quali scegliere per il prossimo passo sono mostrati come cerchi vuoti. Presupponiamo di avere selezionato il pixel P di coordinate ( $x_p$ ,  $y_p$ ) e di dover scegliere quale accendere tra il pixel con incremento a destra (detto east pixel "E") o il pixel con incremento in alto a destra (detto north-east pixel "NE"). Chiamiamo Q il punto di intersezione della linea "scansionata" con la griglia nel punto  $x_{p+1}$ .

Nella formulazione di Bresenham viene calcolata la differenza tra le distanze verticali da E ed NE a Q; in base al risultato verrà acceso il pixel superiore o quello inferiore al punto di mezzo. Nel caso in cui la linea ideale passa precisamente per il punto di mezzo, l'algoritmo prevede comunque l'accensione di un solo pixel (sempre e comunque quello superiore "NE"). L'errore (ovvero la distanza verticale tra il pixel acceso e la linea ideale) è sempre minore o uguale a  $\frac{1}{2}$ .

Tutto quello di cui abbiamo bisogno ora, è un modo di calcolare da quale parte della linea si trova il punto di mezzo. Rappresentiamo la linea con una funzione implicita di coefficienti a, b e c: F(x, y) = ax + by + c = 0.

Se  $\Delta y = y_1 - y_0$ , e  $\Delta x = x_1 - x_0$ , la formula per ricavare la pendenza può essere scritta come:  $y = \Delta y / \Delta x * x + B;$ 

quindi:

$$F(x,y) = \Delta y * x - \Delta x * y + B * \Delta x = 0$$

Dove  $a = \Delta y$ ,  $b = \Delta - x e c = B * dx$  nella forma implicita.

Può essere facilmente verificato che  $F(x_a; y_a)$  ha la proprietà di valere 0 se il punto  $(x_a; y_a)$  appartiene alla linea, di essere negativa se  $(x_a; y_a)$  è sopra la linea e di essere positiva se è sotto.

Per applicare il criterio del punto di mezzo abbiamo solamente bisogno di calcolare d =  $F(M)=F(x_p+1, y_p+1/2)$  dove con *d* indichiamo la nostra decisione variabile; se d > 0 scelgo NE, se d < 0 scelgo E e se d = 0 posso scegliere il punto che voglio, ad esempio E. Scegliendo E, nel passo successivo M sarà incrementata di un passo nella direzione x. Quindi,

dove però:

$$d_{old} = a(x_p + 1) + a(y_p + \frac{1}{2})$$

quindi  $d_{new} = d_{old} + a = d_{old} + per$  ottenere la differenza incrementale, scriviamo:  $d_{new} = d_{old} + \Delta_E$ . Se invece la nostra scelta è caduta sul punto NE, M è incrementata sia nella direzione x che in quella y. Perciò,

$$d_{new} = F(x_p + 2, y_p + 3/2)$$
  
= a (x<sub>p</sub> + 2) + b(y<sub>p</sub> + 3/2) + c

Sottraendo d<sub>old</sub> da d<sub>new</sub> per ottenere la differenza incrementale scriviamo: d<sub>new</sub> = d<sub>old</sub>+ a + b. L'incremento da aggiungere a d dopo NE è detto  $\Delta_{NE}$ ;  $\Delta_{NE}$  = a + b = d<sub>y</sub> - d<sub>x</sub>.

Riassumiamo la tecnica incrementale del punto di mezzo. Ad ogni passo, l'algoritmo sceglie tra due pixel basati sulla prova della decisione variabile calcolata al passo precedente; poi, aggiorna la decisione variabile aggiungendo  $\Delta E$  o  $\Delta NE$  a seconda della scelta del pixel. Poiché il primo pixel è semplicemente il primo punto finale di coordinate (x0, y0) possiamo direttamente calcolare il valore iniziale di d scegliendo tra E ed NE. Il primo punto di mezzo si troverà sulle coordinate (x0 + 1, y0 +  $\frac{1}{2}$ ), e

$$F(x_0 + 1, y_0 + \frac{1}{2}) = a (x_0 + 1) + b(y_0 + \frac{1}{2}) + c$$
  
=  $ax_0 + by_0 + c + a + \frac{1}{2}b$   
=  $F(x_0, y_0) + a + \frac{1}{2}b$ 

Ma (x<sub>0</sub>, y<sub>0</sub>) è un punto della linea e F(x<sub>0</sub>, y<sub>0</sub>) = 0; perciò d<sub>start</sub>= a +  $\frac{1}{2}$ b = dy -  $\frac{1}{2}$ dx. Usando d<sub>start</sub> scegliamo automaticamente il secondo pixel e continuiamo con questo metodo. Per eliminare la frazione contenuta in d<sub>start</sub>, ridefiniamo la nostra F originale moltiplicandola per 2; F(x,y) = 2 (ax + by + c). Così moltiplichiamo le costanti e la decisione variabile per 2. Ma non influiamo sul valore della decisione variabile, che è la cosa principale per il calcolo del punto di mezzo. Come possiamo vedere nell'algoritmo del punto di mezzo rappresentato nello pseudocodice listato qui sotto, il circuito interno è abbastanza semplice. Il primo enunciato del circuito, il test su d, determina la scelta del pixel, ma in realtà incrementiamo x e y dopo aver aggiornato la decisione variabile (per compatibilità con gli algoritmi del cerchio e dell'ellisse). Questa versione dell'algoritmo funziona solo per quelle linee la cui pendenza è compresa tra 0 e 1. In *[R.F Sproull, Using Program Transformations to Derive Line-Drawing Algorithms, ACM TOG 1 (1982), 259-273]*, la formulazione di quest'algoritmo di Bresenhamviene elegantemente modificata tramite una successione di trasformazioni a partire dal "rozzo" algoritmo iniziale.

```
void MidPointLine (int x0, int y0, int x1, int y1, int value)
{
       int dx = x1 - x0;
       int dy = y1 - y0;
       int d = 2 * dy - dx;
                                                 /* Initial value of d */
                                                 /* Incremental used for move to E */
       int incrE = 2^* dy;
       int incrNE = 2^* (dy - dx);
                                                 /* Incremental used for move to NE */
       int x = x0;
       int y = y0;
       WritePixel (x, y, value);
                                                 /* The start pixel */
       while (x \le x1) {
               if (d \le 0) {
                                                 /* Choose E */
               d + = incrE:
               x++;
                                                 /* Choose NE */
       } else {
               d + = incrNE;
               x++;
               y++;
                                                 /* The selected pixel closest to the line */
       WritePixel (x, y, value);
       /* while */
   }
       /* MidpointLine */
```

Per una linea che va dal punto (5, 8) al punto (9, 11), i valori successivi di d sono 2, 0, 6 e risultano rispettivamente nella selezione di NE, E, NE e poi ancora NE come mostrato nella figura sotto.



La linea appare seghettata in maniera abnorme perché la scala è stata notevolmente ingrandita così come lo spazio che intercorre tra un pixel e l'altro, espediente usato al fine di rendere più chiara la geometria dell'algoritmo. Per la stessa ragione, i disegni nella sezione seguente saranno diversi da come appaiono realmente sullo schermo.

## Problemi:

#### Ordine del punto di mezzo.

Tra le complicazione da considerare c'è quella in cui dobbiamo assicurarci che la linea che viene disegnata dal punto  $P_0$  al punto  $P_1$  sia la stessa della linea che va dal punto  $P_1$  al punto  $P_0$ , così che l'apparenza della linea sia indipendente dall'ordine in cui vengono specificati i punti finali. Per il *midpoint* l'unica situazione in cui la direzione della linea è importante si verifica quando la linea passa esattamente nel punto intermedio fra i due punti candidati, per cui la variabile di decisione vale zero. In questo caso infatti per la linea che si muove da sinistra destra si è deciso di accendere il pixel E; è necessario scegliere SW (sud ovest) se si va nella direzione opposta, anche se, per simmetria, verrebbe istintivo scegliere W (ovest). Aggiustamenti simili sono necessari per inclinazioni diverse.

Esiste poi una situazione alternativa, ossia quella di ordinare i punti terminali prima di effettuare la conversione, che è però inefficiente se pensiamo a linee non piene (ad esempio tratteggiate). Infatti per una linea tratteggiata con un pattern 11110000 (ossia con 4 punti accesi e 4 spenti) che viene convertita, l'ordine dei punti terminali è fondamentale.

#### Partenza dai bordi di un rettangolo di taglio.

Una ulteriore complicazione deriva dalla necessità di troncare le linee (clipping), in modo che siano contenute nel rettangolo di presentazione (viewport).

La parte superiore della prossima figura mostra una linea che è stata tagliata al margine sinistro,  $x = x_{min}$ , del rettangolo sezionato. Il punto di intersezione della linea con il margine ha una coordinata intera x ed una coordinata y solitamente non reale. Il pixel al margine sinistro,  $(x_{min}, Round(mx_{min} + B))$ , è lo steso pixel che si disegnerebbe con questo valore x per la linea non tagliata. (Naturalmente per Round $(mx_{min} + B)$  intendiamo l'intero che più si avvicina a...) Dato questo pixel di valore iniziale, in seguito dobbiamo iniziare la decisione variabile al punto di mezzo tra la posizione E e quella NE nella prossima colonna superiore.

Dando questo punto come iniziale per l'algoritmo di conversione è necessario inizializzare la variabile di decisione  $d_{start}$  per il punto M ad x =  $x_{min}$  + 1 usando l'equazione della linea NON troncata. Se invece prima si tronca ad intero e poi si converte la linea come se fosse fra i nuovi punti troncati il risultato è una linea con inclinazione diversa.

La situazione è più complicata se la linea viene troncata su di un bordo orizzontale anzichè verticale, come mostrato nella parte inferiore della prossima figura. Per la retta raffigurata vi sono più punti ad  $y = y_{min}$  che corrispondono all'estremo inferiore della linea. Naturalmente noi vorremmo che fossero disegnati tutti, ma calcolando semplicemente l'intersezione analitica della linea con la linea di scansione  $y = y_{min}$  e poi spostando il valore x del punto d'intersezione, si ha il pixel A, non il punto più a sinistra nella serie di pixels mostrati, ovvero il pixel B. Dalla figura, appare chiaro che il pixel più a sinistra della serie di pixel disegnati (B) è quello che si trova proprio sopra e a destra del posto della griglia in cui la linea incrocia sul punto di mezzo  $y = y_{min} - \frac{1}{2}$ .

Se si calcola l'intersezione con la retta  $y = y_{min}$  e si arrotonda la x risultante si inizia a disegnare dal punto A, anzichè da B. La soluzione corretta è quella di calcolare l'intersezione della retta con y = y<sub>min</sub>.



#### Variare l'intensità della linea a seconda della pendenza.

Consideriamo le due linee scandite nella seguente figura: come possiamo subito notare, le due linee hanno lo stesso numero di punti, ma la linea con pendenza m=1 (45°) è più lunga di un fattore radice di 2 rispetto a quella orizzontale. Se non verranno presi degli accorgimenti la linea inclinata sarò disegnata con un tratto leggermente più sottile rispetto all'altra; per alcuni sistemi è possibile compensare la variazione di intensità scrivendo le linee inclinate con un valore maggiore rispetto a quelle orizzontali o verticali (questa è però solamente una sottigliezza che viene ignorata dalla maggior parte dei sistemi grafici).

Il problema viene tuttavia risolto grazie all'antialiasing, dove le linee vengono trattate come rettangoli sottili, in cui l'intensità appropriata viene calcolata per i vari pixel.



Considerare la linea come un rettangolo è anche un modo per creare linee spesse. Nella successiva *sezione sul tratteggio* mostreremo come modificare l'algoritmo di scansione di base per lavorare con primitive spesse e con primitive la cui apparenza è influenzata dallo stile del tratto e da quello della linea.

## ALGORITMI DI SCAN CONVERSION PER CIRCONFERENZE

Molti sistemi grafici non dispongono di una primitiva per archi di circonferenza. Per generare questi archi si devono sviluppare opportuni algoritmi che possono essere resi piu' efficienti se si tiene conto della simmetria ad ottanti dell'ellisse, sia per il clipping sia per la scansione. L'equazione di un cerchio centrato all'origine è  $x^2 + y^2 = R^2$ . I cerchi non centrati all'origine possono essere traslati all'origine, scanditi e poi riportati indietro: basta aggiungere ai pixels l'offset appropriato.

Ci sono vari modi facili ma inefficienti per eseguire la scansione di un cerchio. Per risolvere y nell'equazione implicita del cerchio, poniamo y = f(x) come:

$$Y = \pm \sqrt{R^2 - x^2}.$$

Per disegnare un quarto di cerchio (gli altri quarti sono disegnati per simmetria), possiamo partire da x=0 ed incrementare successivamente in passi unitari di Dx=1 fino a Dx=R, risolvendo ogni volta per +y. Questo approccio funziona, ma è inefficiente a causa dei calcoli in virgola mobile e delle radici, inoltre con questa strategia il cerchio disegnato avrà grandi vuoti per i valori di x vicini a R, perché la pendenza del cerchio lì diventa infinita (vedere figura in basso).



Un metodo simile ed altrettanto inefficiente, il quale comunque evita i larghispazi vuoti, è quello di tracciare (R cos $\theta$ , R sin $\theta$ ) per  $\theta$  che va da 0° a 90°.

Simmetria ad ottanti (*Eight-Way Symmetry*)

Possiamo migliorare il processo per disegnare di cui si è parlato nella sezione precedente traendo grandi vantaggi dalla simmetria in un cerchio. Per prima cosa consideriamo un cerchio centrato all'origine. Se il punto di coordinate (x, y) è sul cerchio, allora possiamo calcolare facilmente altri sette punti sul cerchio (usando la simmetria per gli ottanti, calcolando quindi tutte le combinazioni ±x e ±y), come mostrato nella figura 11. Un altro vantaggio di questa procedura è data dal fatto che abbiamo bisogno di calcolare solo 45° della circonferenza totale per determinare il tutto. Per un cerchio centrato all'origine, gli otto punti simmetrici possono essere esposti con la procedura del CirclePoints (la procedura è facilmente applicabile anche in caso di cerchi centrati arbitrariamente con coordinate intere):

void CirclePoints (int x, int y, int value)
{

```
WritePixel (x, y, value);
WritePixel (y, x, value);
WritePixel (y, -x, value);
WritePixel (x, -y, value);
WritePixel (-x, -y, value);
WritePixel (-y, -x, value);
WritePixel (-y, x, value);
WritePixel (-x, y, value);
/* CirclePoints */
```

}

Diventa necessario modificare la procedura del circle-points quando x = y per evitare che ciascuno dei punti venga scritto due volte.



Algoritmo del punto di mezzo per un cerchio (*Midpoint Circle Algorithm*) Un generatore incrementale di punti più efficiente dei metodi che abbiamo appena discusso è sviluppato in [*J.E. Bresenham*, <u>A Linear Algorithm for</u> <u>Incremental Digital Display of Circular Arcs</u>, Comm. ACM 20 (1977), 100-106]. Concepito per essere utilizzato con i plotters, l'algoritmo genera tutti i punti su un cerchio centrato all'origine incrementando completamente i punti intorno al cerchio.

Deriviamo un algoritmo simile utilizzando ancora il criterio del punto di mezzo, il quale, nel caso di raggi e punti centrali interi, genera lo stesso ottimale set di pixel. Consideriamo un cerchio di 45°, il secondo ottante (spicchio) da x = 0 a x =  $y = R /\sqrt{2}$  e usiamo la procedura seguente per disegnare i punti sull'intero cerchio. Come per l'algoritmo del punto di mezzo della linea la strategia è quella di accendere il pixel che tra due è più vicino al cerchio calcolando la funzione al punto di mezzo. Nel secondo ottavo, se il pixel P di coordinate (x<sub>p</sub>, y<sub>p</sub>) è stato precedentemente scelto come quello più vicino al cerchio la sceltra del prossimo pixel è tra il pixel E ed SE (Fig. 14).



Poniamo  $F(x, y) = x^2 + y^2 - R^2$ ; la funzione è 0 sul cerchio, negativa all'interno e positiva all'esterno. Si può dimostrare che se il punto di mezzo tra il pixel E ed il pixel SE si trova fuori dal cerchio, allora il pixel SE è più vicino al cerchio. D'altra parte, se il punto di mezzo è all'interno del cerchio allora il pixel E sarà quello più vicino al cerchio.

Come per le linee scegliamo sulla base della decisione variabile d, che è il valore della funzione al punto di mezzo,

$$d_{old} = F (x_p+1, y_p - \frac{1}{2}) = (x_p + 1)^2 + (y_p - \frac{1}{2})^2 - R^2$$

se d<sub>old</sub> < 0 si sceglie E, e il prossimo punto di mezzo sarà incrementato di uno verso l'alto (in x). Quindi,

$$d_{new} = F (x_p + 2, y_p - \frac{1}{2}) = (x_p + 2)^2 + (y_p - \frac{1}{2})^2 - R^2$$

e  $d_{new} = d_{old} + (2x_p+3)$ ; perciò l'incremento

 $\Delta_{\mathsf{E}}=2\mathsf{x}_{\mathsf{p}}+3.$ 

Se  $d_{old} \ge 0$ , si sceglie SE e il prossimo punto di mezzo avrà un incremento verso l'alto di 1 (in x) e verso il basso di 1 (in y). Allora:

$$d_{new} = F (x_p+2, y_p - 3/2) = (x_p + 2)^2 + (x_p - 3/2)^2 - R^2$$

se d<sub>new</sub> = d<sub>old</sub> +  $(2x_p - 2y_p + 5)$ , l'incremento  $\Delta_{SE} = 2x_p - 2y_p + 5$ .

Ricordiamo che, nel caso delle linee  $\Delta_E \Delta_{NE}$  erano costanti; invece nel caso quadratico  $\Delta_E$  e  $\Delta_{SE}$  variano ad ogni passo e sono funzioni dei valori particolari di  $x_p$  e  $y_p$  al pixel scelto nella precedente iterazione. Poiché queste funzioni sono espresse in termini di ( $x_p$ ,  $y_p$ ), chiamiamo P punto di valutazione. Le funzioni e  $\Delta$  possono essere calcolate direttamente in ogni passo collegando ai valori di x e y il pixel scelto nella precedente iterazione.

Questo calcolo diretto non richiede molto tempo poiché le funzioni sono solamente funzioni lineari.

In breve, facciamo gli stessi due passi ad ogni iterazione dell'algoritmo come facevamo per la linea:

(1)scegliamo il pixel basato sul segno della variabile d calcolato durante la precedente iterazione;

(2) aggiorniamo la decisione variabile d con il  $\Delta$  che corrisponde alla scelta del pixel.

La sola differenza rispetto all'algoritmo lineare è che aggiornando d, calcoliamo la funzione lineare del punto di valutazione.

Tutto ciò che ci resta da fare ora è calcolare la condizione iniziale. Limitando l'algoritmo ai raggi interi nel secondo ottavo, sappiamo che il pixel iniziale si trova sul cerchio di coordinate (0R). Quindi il prossimo punto di mezzo si trova al punto di coordinate (1, R -  $\frac{1}{2}$ ), e F (1, R -  $\frac{1}{2}$ )= 1 + (R<sup>2</sup> –R +  $\frac{1}{4}$ ) - R<sup>2</sup> = 5/4 - R. Adesso possiamo scrivere l'algoritmo (il listato è nella prossima figura). Possiamo notare come questo algoritmo sia strutturalmente simile a quello per la retta.

Il problema con questa versione è che siamo abituati ad utilizzare un'aritmetica reale a causa "dell'inizializzazione frazionale" di d. Benché la procedura possa essere facilmente modificata per lavorare con quei cerchi che non sono localizzati su centri interi (o che non hanno raggi interi), vorremmo una versione dell'algoritmo intera e più efficiente. Perciò usiamo un semplice programma di trasformazione per eliminare le frazioni.

Per prima cosa definiamo una nuova decisione variabile h, dove h = d -  $\frac{1}{4}$  e sostituiamo con (h +  $\frac{1}{4}$ ) la d nel codice. Ora l'inizio è h = 1 – R e il confronto d < 0 diventa h < -  $\frac{1}{4}$ .

```
void MidpointCircle (int radius, int value)
                /* Assumes center of circle is at origin */
  int x = 0;
  int y = radius;
  double d = 5.0 / 4.0 - radius,
  CirclePoints (x, y, value);
while (y > x) {
  if (d < 0)
                                 /* Select E */
          d + = 2.0 * x + 3.0;
                                  /* Select SE */
  else {
          d + = 2.0 * (x - y) + 5.0;
          y--;
  }
  x++;
  CirclePoints (x, y, value);
             /* while */
  }
             /* MidpointCircle */
```

Comunque poichè h comincia con un valore intero ed è incrementato da valori interi ( $\Delta_E e \Delta_{SE}$ ), possiamo cambiare il confronto e farlo diventare semplicemente h < 0. Ora abbiamo un algoritmo intero in termini di h. Per coerenza con l'algoritmo lineare sostituiremo la d con la h completamente. L'algoritmo finale intero è mostrato in figura 16.
```
void MidpointCircle (int radius, int value)
               /* Assumes center of circle is at origin. Integer arithmetic only */
{
  int x = 0;
  int y = radius;
  int d = 1 - radius;
  CirclePoints (x, y, value);
while (y > x) {
  if (d < 0)
                                /* Select E */
          d + = 2 * x + 3;
                                /* Select SE */
  else {
          d + = 2 * (x - y) + 5;
          y--;
  }
  x++;
  CirclePoints (x, y, value);
            /* while */
  }
}
            /* MidpointCircle */
```

La seguente figura mostra il secondo ottavo di un cerchio di 17 raggi generato dall'algoritmo, e il primo ottavo generato per simmetria (se si confontano i risultati con la figura all'inizio di questa sezione si vedono i miglioramenti ottenuti).



# Differenze del secondo ordine (Second-order differences)

L'algoritmo visto si può migliorare utilizzando nuovamente la tecnica di calcolo incrementale già applicata una volta. Le funzioni  $\Delta$  sono infatti espressioni lineari, e vengono calcolate direttamente. Ogni funzione polinomiale può tuttavia essere calcolata in modo incrementale, come si è fatto per le variabili di decisione della retta e della circonferenza, mediante la seguente strategia: valutare la funzione direttamente in due punti adiacenti, calcolare la di erenza (che per polinomiali è sempre polinomiale di grado inferiore) ed applicare la differenza ad ogni iterazione. In questo caso una tecnica per velocizzare ulteriormente il calcolo della circonferenza è quindi quella di valutare le di erenze parziali del primo e del secondo ordine, cioè di valutare anche le funzioni  $\Delta E e \Delta SE$  in modo incrementale.

Se nella iterazione corrente si è scelto E, il punto di valutazione si sposta da P =  $(x_p, y_p)$  a P<sub>E</sub> =  $(x_p + 1, y_p)$ . Come abbiamo visto la differenza del primo ordine calcolata in P vale  $\Delta E_{old} = 2xp + 3$ . Quindi  $\Delta E_{new}$  calcolata in P<sub>E</sub> diventa  $\Delta E_{new} = 2(x_p + 1) + 3$ , e la differenza del secondo ordine  $\Delta E_{new} - \Delta E_{old} = 2$ .

Se invece si è scelto SE nell'iterazione corrente il punto di valutazione si sposta da P = (x<sub>p</sub>, y<sub>p</sub>) a P<sub>SE</sub> = (x<sub>p</sub> +1; y<sub>p</sub> -1). Quindi  $\Delta_{Enew}$  in PSE vale  $\Delta_{Enew}$  = 2(xp +1)+3 e la differenza del secondo ordine è  $\Delta_{Enew}$  -  $\Delta_{Eold}$ .

Il codice completo (nel linguaggio di programmazione C) è illustrato in figura 18.

```
void MidpointCircle (int radius, int value)
                 /* This procedure uses second-order partial differences to */
                 /* compute increments in the decision variable. */
                 /* Assumes center of circle is at origin. */
{
  int x = 0;
  int y = radius;
  int d = 1 - radius;
  int deltaE = 3;
  int deltaSE = -2 * radius +5;
  CirclePoints (x, y, value);
while (y > x) {
                                  /* Select E */
  if(d < 0)
             {
          d + = deltaE;
          deltaE + = 2;
          deltaSE + = 2;
        } else {
                                /* Select SE */
          d + = deltaSE;
          deltaE + = 2;
          deltaSE + = 4;
         y--;
        }
        x++;
  CirclePoints (x, y, value);
  }
           /* while */
           /* MidpointCircle */
}
```

# ALGORITMI DI SCAN CONVERSION PER ELLISSI

Consideriamo un'ellisse centrata nell'origine (0, 0). Essa è determinata dall'equazione

$$F(x, y) = b^2 x^2 + a^2 y^2 - a^2 b^2 = 0$$

dove 2a è la lunghezza del diametro lungo l'asse x, e 2b è la lunghezza del diametro lungo l'asse y. La tecnica del punto di mezzo discussa per le linee e i cerchi può anche essere applicata in generale alle coniche. In questa sezione consideriamo l'ellisse centrata nell'origine e con i due assi paralleli agli assi coordinati. Inoltre, per semplificare l'algoritmo, disegniamo solo l'arco dell'ellisse che si trova nel primo quadrante, poiché gli altri tre quadranti possono essere disegnati per simmetria. Notiamo inoltre le ellissi standard centrate nei punti di coordinate intere possono essere disegnati tramite una semplice transazione. L'algoritmo qui presentato si basa su quello di Da Silva [D. Da Silva, Raster] Algorithms for 2D Primitives, M.Sc. Thesis, Computer Science Dept., Brown University, Providence, RI, USA, 1989], che combina la tecnica usata in [M.L.V. Pitteway, Algorithm for Drawing Ellipses on Hyperbolae with a Digital Plotter, Computer Journ. 10 (1967), 282-289], [J.R. Van Aken, An Efficient Ellipse-Drawing Algorithm, Computer Graphics and Appl. 4 (1984), 24-35] e [M.R. Kappel, An Ellipse-Drawing Algorithm for Raster Displays, in Fundamental Algorithms for Computer Graphics (R. Earnshaw, ed.), NATO ASI Seminars, Springer-Verlag, Berlino, 1985, 257-280], con l'uso delle differenze parziali (ossia della discretizzazione delle derivate parziali).

Per prima cosa dividiamo il quadrante in due regioni; il limite tra le due regioni è quello in cui la curva ha una pendenza di –1 (si veda la figura a fine sezione). Determinare questo punto è più complesso di quanto fosse per i cerchi. Il vettore perpendicolare alla tangente della curva nel punto P, detto *gradiente*, è definito come:

grad 
$$F(x, y) = \partial F / \partial x \mathbf{i} + \partial F / \partial y \mathbf{j} = 2b^2 x \mathbf{i} + 2a^2 y \mathbf{j}$$

Il limite tra le due regioni è il punto in cui la pendenza della curva è –1, e questo punto corrisponde ad un vettore gradiente con componenti entrambe 1, ossia un gradiente multiplo di **i** + **j**. La componente **j** del gradiente è più estesa della componente **i** nella regione 1, e viceversa nella regione 2. Quindi, se al prossimo punto di mezzo abbiamo  $a^2(y_p - \frac{1}{2}) = b^2(x_p+1)$ , allora ci spostiamo dalla regione 1 alla regione 2. Così come con l'algoritmo del punto di mezzo, valutiamo la funzione nel punto di mezzo tra due pixels ed usiamo il segno per determinare se il punto di mezzo si trova dentro o fuori l'ellisse, e quindi, quale pixel è il più vicino all'ellisse. Inoltre, nella regione 1, se il pixel corrente è localizzato al punto di coordinate (x<sub>p</sub>, y<sub>p</sub>), allora la decisione variabile per la regione 1,  $d_1$ , è F(x, y) calcolata al punto di coordinate ( $x_p + 1$ ,  $y_p - \frac{1}{2}$ ), il punto di mezzo tra E e SE.



Ora ripetiamo il processo che abbiamo usato nella sezione precedente per derivare i due  $\Delta_s$  per il cerchio. Per spostarci verso E, il punto di mezzo successivo si incrementa di 1 in x. Quindi

$$\begin{split} D_{old} &= F(x_p + 1, \, y_p - \frac{1}{2}) = b^2 \left(x_p + 1\right)^2 + a^2 \left(, \, y_p - \frac{1}{2}\right)^2 - a^2 b^2, \\ D_{new} &= F(x_p + 2, \, y_p - \frac{1}{2}) = b^2 \left(x_p + 2\right)^2 + a^2 \left(, \, y_p - \frac{1}{2}\right)^2 - a^2 b^2 \end{split}$$

Poichè d <sub>new</sub>= d<sub>old</sub> +b<sup>2</sup> (2x<sub>p</sub> + 3), l'incremento è  $\Delta_E = b^2 (2x_p + 3)$ . Per un passo verso SE, il prossimo punto di mezzo si incrementa di uno in alto in x e di uno in basso in y. In tal caso,

$$D_{new} = F(x_p + 2, y_p - 3/2) = b^2 (x_p + 2)^2 + a^2 (y_p - 3/2)^2 - a^2 b^2$$

Poichè  $d_{new} = d_{old} + b^2 (2x_p + 3) + a^2(-2y_p + 2)$ , l'incremento  $\Delta_{SE} = b^2 (2x_p + 3) + a^2(-2y_p + 2)$ .

Nella regione 2, se il pixel corrente è quello di coordinate  $(x_p, y_p)$ , la decisione variabile  $d_2$  è  $F(x_p + \frac{1}{2}, y_p - 1)$ , il punto di mezzo tra S e SE. I calcoli simili a questi che sono stati espletati nella regione 1 dovrebbero essere ripetuti anche nella regione 2.

Dobbiamo anche determinare la condizione iniziale. Assumendo valori interi di a e b, l'ellisse ha inizio nel punto di coordinate (0, b), e il primo punto di mezzo da calcolare si trova nel punto  $(1, b- \frac{1}{2})$ . Allora:

$$F(1, b- \frac{1}{2}) = b^{2} + a^{2}(b- \frac{1}{2})^{2} - a^{2}b^{2} = b^{2} + a^{2}(-b + \frac{1}{4}).$$

Per ogni iterazione nella regione 1, non dobbiamo solamente testare la decisione variabile  $d_1$  e aggiornare la funzione  $\Delta$ , ma dobbiamo anche vedere se esiste la necessità di cambiare regione calcolando il gradiente nel punto di mezzo tra E e SE. Quando il punto di mezzo attraversa la regione 2, i pixels da determinare sono quelli compresi tra SE e S e non più quelli tra S e SE. Se l'ultimo pixel scelto nella regione 1 è localizzato nel punto di coordinate ( $x_p$ ,  $y_p$ ), allora la

decisione variabile  $d_2$  comincia nel punto ( $x_p + \frac{1}{2}, y_p - 1$ ). Smettiamo di disegnare i pixels nella regione 2 quando il valore di y vale 0.

Come con l'algoritmo del cerchio, possiamo anche determinare le funzioni  $\Delta$  direttamente ad ogni iterazione del circuito o calcolarle per differenza. Da Silva mostra che il calcolo parziale di secondo ordine usato per  $\Delta_{S}$  può essere usato anche per il gradiente [D. Da Silva, Raster Algorithms for 2D Primitives, M.Sc. Thesis, Computer Science Dept., Brown University, Providence, RI, USA, 1989]. Inoltre questo lavoro tratta i casi di ellissi ruotate e le condizioni complicate dei limiti nei casi di ellissi molto sottili. Lo pseudocodice usa il calcolo più semplice e diretto invece della formula delle differenze di secondo ordine nonostante questa sia più efficace. Nel caso di valori interi di a e b possiamo eliminare la frazione con una trasformazione di programma e usare solo l'aritmetica intera.



# PRIMITIVE 2D DI SPESSORE MAGGIORE DI UN PIXEL

# Introduzione

Il modo per disegnare primitive di spessore maggiore di uno è concettualmente semplice, basta sostituire la scrittura di un punto convertito con la scrittura di una figura di dimensione e forma specificata (penna). Una linea di spessore unitario può quindi essere pensata come disegnata usando una penna con dimensione pari ad un singolo punto. Questa semplice descrizione lascia tuttavia una serie di questioni. La prima riguarda la forma della penna, tipicamente circolare o (per semplicità) rettangolare, con l'ulteriore problema, per quest'ultima, di decidere se ruotarla secondo la tangente della primitiva che si disegna o tenerla di orientamento fisso.

Altre questioni riguardano la forma degli estremi della linea, l'unione fra più primitive e l'interazione dello spessore con altri attributi, come lo stile di scrittura.

# Replica di punti per riga o colonna

Il modo più semplice per realizzare primitive di spessore arbitrario è quello di scrivere più punti, al posto dell'unico calcolato, sulla stessa colonna (o riga, a seconda della pendenza della linea). Questo modo funziona ragionevolmente bene per linee, aggiungendo i punti sulle colonne per inclinazione entro i 45° gradi, e sulle righe altrimenti. Il risultato è buono, tuttavia le linee terminano sempre con un segmentino orizzontale o verticale: questo può essere spiacevole per linee di un certo spessore, come mostrato nella figura sotto



Un secondo problema di questo algoritmo è che produce dei buchi ben visibili all'unione di linee con inclinazione diversa, se esse sono disegnate con replica in direzioni perpendicolari (ossia se la prima viene disegnata con replica sulle righe e l'altra con replica sulle colonne). Inoltre lo spessore delle linee, inteso come la distanza fra il centro della linea ed il bordo, varia con l'inclinazione, essendo minore per linee inclinate. Infine non si riescono a trattare le primitive con spessore pari, dato che il punto generato è inevitabilmente a metà dello spessore. In conclusione questa tecnica rappresenta una approssimazione veramente rozza di entità spesse, accettabile solo per spessori limitati e per lavori poco professionali.

# Penna mobile

Un semplice modo di applicare una penna di area maggiore di uno è quello di utilizzare una forma quadrata o rettangolare con centro sul punto correntemente convertito e che rimanga sempre parallela agli assi: il risultato che ci si può aspettare è mostrato in Fig. 2b. Da notare come la linea sia simile a quella ottenuta per replica di punti tranne che per gli estremi, ora prolungati di mezzo quadrato.



Come per il caso precedente lo spessore delle linee varia con l'inclinazione, ma in modo opposto: le linee orizzontali o verticali sembrano le più sottili. Si può eliminare questo eccetto ruotando il rettangolo in modo da seguire l'inclinazione delle primitive, ma allora è meglio utilizzare una penna circolare, come in 3b (Fig. 2.17), che occore anche il vantaggio di avere i punti terminali arrotondati.



Per quanto riguarda la realizzazione di questo tipo di primitive spesse, è opportuno notare come sia inefficiente scrivere tutta l'area sottesa dalla penna ad ogni punto convertito, in quanto la maggioranza dei punti vengono scritti più volte. La soluzione corretta è quella di calcolare ogni volta la parte non ancora scritta in precedenza, ed aggiornare solo quella.

# Riempimento fra contorni

Il terzo metodo per visualizzare una primitiva spessa è quello di generare il suo contorno esterno a distanza t/2 da entrambi i lati della primitiva ideale a spessore nullo, e poi riempirla con il colore voluto.

Per le linee questo significa disegnare un rettangolo di spessore *t* e di lunghezza pari a quella della linea originale, e poi utilizzare una delle tecniche note di riempimento di poligoni. Naturalmente è necessario convertire i vertici del rettangolo in modo che abbiano coordinate intere, e questo può peggiorare il risultato finale.

Per le circonferenze si disegnano i due cerchi limite esterno ed interno, con i raggi che differiscono per *t*, e si riempe lo spazio fra di essi.

Per le ellissi la situazione non è così semplice, nel senso che la curva generata tenendosi a distanza fissa da una ellisse non è una ellisse, ma è descritta da equazioni di ottavo grado! Naturalmente queste ultime sono pesanti da convertire e quindi, come al solito, si approssima. Il modo comunemente scelto è quello di costruire i limiti dell'area da riempire con due ellissi i cui assi differiscano di t e di ignorare la lieve differenza rispetto al calcolo corretto. Lo stesso problema può verificarsi per altri tipi di primitive a volte disponibili nelle librerie grafiche.

# Approssimazione mediante sequenza di linee spesse

Dato che è possibile approssimare qualsiasi primitiva con una sequenza di segmenti connessi, ossia con una spezzata che unisce punti appartenenti alla primitiva, un modo per disegnare entità più spesse di uno, è quello di convertirle in spezzate (purchè si sappia disegnare queste ultime con lo spessore voluto e si sia risolto il problema dell'unione morbida fra segmenti di inclinazione diversa). Naturalmente dobbiamo essere in grado di calcolare l'approssimazione, che in genere produce segmenti con estremi non interi, in modo efficiente e abbastanza sofisticato da fornire segmenti corti dove la curvatura è alta e lunghi dove è bassa.

# **QUARTA PARTE**

# GEOMETRIA ANALITICA DEL PIANO E DELLO SPAZIO, TRASFORMAZIONI AFFINI E TRASFORMAZIONI PROSPETTICHE

Questa Parte è tratta dal libro "Appunti di Algebra Lineare" di L. Zsido e M. Picardello

#### CAPITOLO 1

# Vettori e geometria euclidea ed analitica nel piano

Rinviamo il lettore ad un libro di testo di Algebra Lineare e Geometria per riferimenti e maggiori dettagli su questo capitolo ed i successivi.

#### 1.1. L'equazione della retta nel piano

Ci sono due modi di scrivere l'equazione di una retta:

• Equazione parametrica: la retta che passa per il punto  $\mathbf{p}$  ed ha la direzione del vettore  $\mathbf{v}$  ha equazione

$$\mathbf{r}(t) = \mathbf{p} + t\mathbf{v}$$

 $\operatorname{con} t \in \mathbb{R}.$ 

• Equazione cartesiana: consideriamo due punti distinti  $\mathbf{p} = (p_1, p_2)$  e  $\mathbf{q} = (q_1, q_2)$  non verticalmente allineati, cioè tali che  $p_1 \neq q_1$ . La retta che passa per questi due punti ha pendenza  $m = \frac{q_2 - p_2}{q_1 - p_1}$ , e quindi i suoi punti  $(x_1, x_2)$ , per proporzionalità, soddisfano l'equazione  $x_2 - p_2 = m(x_1 - p_1)$ , cioè

$$x_2 = p_2 - m(x_1 - p_1). (1.1.1)$$

Se i due punti sono allineati verticalmente, allora si scambia il ruolo di ascisse ed ordinate e si ottiene l'equazione

$$x_1 = p_1 - m(x_2 - p_2), \qquad (1.1.2)$$

dove ora il denominatore è necessariamente non nullo.

Si noti che l'equazione cartesiana è del tipo

$$ax_1 + bx_2 = d \tag{1.1.3}$$

dove  $a, b \in d$  sono costanti reali. Questa formulazione ha il vantaggio che non contiene denominatori, e quindi non la si deve cambiare se la retta è verticale.

NOTA 1.1.1. (Equazione della retta in forma normale.) Si osservi che, se poniamo  $\mathbf{n} = (a, b) \in \mathbf{x} = (x_1, x_2)$ , l'equazione (1.1.3) si scrive  $\mathbf{n} \cdot \mathbf{x} = d$ . Se d = 0, allora la retta è quella perpendicolare a  $\mathbf{n}$  che passa per l'origine (perché se d = 0 l'origine soddisfa l'equazione). Se invece  $d \neq$  0, allora la retta è costituita dai punti la cui proiezione sulla direzione di **n** vale  $d\frac{\mathbf{n}}{\|\mathbf{n}\|}$  (esercizio), cioè è la parallela a quella precedente a distanza  $\frac{|d|}{\|\mathbf{n}\|}$  in direzione **n** (quindi a distanza  $\frac{|d|}{\|\mathbf{n}\|}$  dall'origine).

In particolare, se si normalizza il vettore  $\mathbf{n}$ , il coefficiente al lato di destra dell'equazione della retta rappresenta la distanza dall'origine; l'equazione così normalizzata si chiama equazione normale:

$$\frac{\mathbf{n}}{\|\mathbf{n}\|} \cdot \mathbf{x} = \frac{d}{\|\mathbf{n}\|}$$

In forma parametrica, per equazione normale si intende quella in cui il vettore direzionale è normalizzato ed il punto di partenze (per t = 0) è un vettore perpendicolare alla retta:

$$\mathbf{r}(t) = \mathbf{p} - P_{\mathbf{v}}(\mathbf{p}) + t \frac{\mathbf{v}}{\|\mathbf{v}\|} = \mathbf{p} - \frac{\mathbf{p} \cdot \mathbf{v}}{\|\mathbf{v}\|^2} \mathbf{v} + t \frac{\mathbf{v}}{\|\mathbf{v}\|}$$

dove  $t \in \mathbb{R}$  e  $P_{\mathbf{v}}(\mathbf{p}) = \frac{\mathbf{p} \cdot \mathbf{v}}{\|\mathbf{v}\|^2} \mathbf{v}$  è la proiezione del vettore  $\mathbf{p}$  lungo  $\mathbf{v}$ . La distanza della retta dall'origine è la proiezione di un suo punto qualsiasi, ad esempio  $\mathbf{p}$ , perpendicolarmente a  $\mathbf{v}$ , e quindi è

$$\|\mathbf{p} - rac{\mathbf{p} \cdot \mathbf{v}}{\|\mathbf{v}\|^2} \mathbf{v}\|$$
 .

ESEMPIO 1.1.2. (Proiezione ortogonale di un punto su una retta.) Calcoliamo la proiezione ortogonale **p** del punto **x** sulla retta **R** di equazione ax + by = d. Sappiamo dalla Nota 1.1.1 che il versore  $\mathbf{n} = \frac{1}{\sqrt{a^2+b^2}}(a, b)$  è il versore normale a **R**. Possiamo allora trovare **p** in due modi equivalenti.

Il primo consiste nel considerare la retta uscente da  $\mathbf{x}$  in direzione  $\mathbf{n}$ , cioè di equazione parametrica  $\mathbf{r}(t) = \mathbf{p} + t\mathbf{n}$ , e trovare il punto di intersezione con  $\mathbf{R}$ : questo significa imporre che il punto  $\mathbf{r}(t)$  soddisfi l'equazione ax + by = d, trovare il valore  $t_0$  di t (positivo o negativo a seconda del verso del versore normale) per cui questo succede e ricavare  $\mathbf{p} = \mathbf{r}(t_0)$ .

Il secondo modo consiste nello scomporre **x** rispetto ad una base ortonormale in cui uno dei vettori sia parallelo a **R**. Per semplicità rinormalizziamo i coefficienti dell'equazione della retta ax + by = din maniera da scriverla come ax + by = d', dove ora  $a^2 + b^2 = 1$  e  $d' = d/\sqrt{a^2 + b^2}$ . Trattiamo prima il caso in cui **R** passi per l'origine, cioè sia d = d' = 0. Allora il versore normale a **R** è  $\mathbf{n}_1 = (a, b)$ . Un versore perpendicolare a  $\mathbf{n}_1$ , e quindi giacente lungo **R**, è  $\mathbf{n}_2 = (-b, a)$ . Da (??) abbiamo la decomposizione ortogonale  $\mathbf{x} = \mathbf{x} \cdot \mathbf{n}_1 \mathbf{n}_1 + \mathbf{x} \cdot \mathbf{n}_2 \mathbf{n}_2$ . Allora  $\mathbf{p} := \mathbf{x} - \mathbf{x} \cdot \mathbf{n}_1 \mathbf{n}_1$  è la proiezione ortogonale cercata: infatti **p** giace in  $\mathbf{R}$  ed è tale che  $\mathbf{x} - \mathbf{p}$  è proporzionale a  $\mathbf{n}_1$  e quindi perpendicolare a  $\mathbf{R}$ .

Nel caso generale in cui  $d' \neq 0$  il procedimento precedente ci dà la proiezione **p**' di **x** non su **R** ma sulla sua parallela **R**' che passa per l'origine. Ora la proiezione **p** su **R** si ottiene spostandosi perpendicolarmente a **R**' di una distanza |d'|:

$$\mathbf{p} = \mathbf{p}' + d'\mathbf{n}_1$$

(nel secondo membro abbiamo preso la somma invece della differenza perché, se ad esempio a, b > 0 allora la retta  $\mathbf{R}$  è a destra di  $\mathbf{R}'$  se d' > 0 (cioè d > 0) e a sinistra altrimenti, quindi per spostarsi da  $\mathbf{R}$ a  $\mathbf{R}'$  bisogna muoversi nella direzione di  $\mathbf{n}_1$  se d' > 0 e al contrario se d' < 0.

In particolare, se una retta ha equazione  $\mathbf{n} \cdot \mathbf{x} = d$  (e quindi equazione normale  $\mathbf{n} \cdot \mathbf{x} / ||\mathbf{n}|| = d / ||\mathbf{n}||$ ), la distanza di un punto  $\mathbf{p}$  dalla retta è

$$\frac{\mathbf{n} \cdot \mathbf{p} - d}{\|\mathbf{n}\|} \,. \tag{1.1.4}$$

Infatti, per quanto osservato sopra, questa è la differenza fra le distanze dall'origine della retta data e della sua parallela che passa per  $\mathbf{p}$ .

#### ESEMPIO 1.1.3. (Base e altezza di un triangolo.)

Dati i tre vertici  $\mathbf{x}_i$ , con i = 1, 2, 3, consideriamo la base del triangolo data dal segmento che unisce  $\mathbf{x}_1$  a  $\mathbf{x}_2$ . Per tracciare l'altezza dal vertice  $\mathbf{x}_3$  basta applicare il procedimento del precedente Esempio 1.1.2. La base giace sulla retta di equazione parametrica

$$\mathbf{r}(t) = \mathbf{x}_1 + t(\mathbf{x}_2 - \mathbf{x}_1)$$

con  $t \in \mathbb{R}$ , e (se abbiamo scelto  $\mathbf{x}_1 = (x', y')$  e  $\mathbf{x}_2 = (x'', y'')$  non allineati verticalmente, cioè  $x' \neq x''$ ) l'equazione cartesiana (1.1.1)

$$y - y' = m(x - x')$$

con coefficiente angolare

$$m = \frac{y'' - y'}{x'' - x'}$$

(se i due punti sono allineati verticalmente si deve scambiare il ruolo delle coordinate  $x \in y$  in queste uguaglianza, come facemmo in (1.1.2)).

In ciascuna di queste formulazioni applichiamo il corrispondente metodo dell'Esempio precedente per trovare il piede dell'altezza.

ESEMPIO 1.1.4. *(Equazioni delle bisettrici di due rette.)* Grazie a (1.1.4), date due rette  $\mathbf{n}_1 \cdot \mathbf{x} = d_1$  e  $\mathbf{n}_2 \cdot \mathbf{x} = d_2$ , le loro due bisettrici hanno equazione

$$\frac{\mathbf{n}_1 \cdot \mathbf{x} - d_1}{\|\mathbf{n}_1\|} = \left| \frac{\mathbf{n}_2 \cdot \mathbf{x} - d_2}{\|\mathbf{n}_2\|} \right|$$
(1.1.5)

e cioè

е

84

$$\frac{\mathbf{n}_1 \cdot \mathbf{x} - d_1}{\|\mathbf{n}_1\|} = \frac{\mathbf{n}_2 \cdot \mathbf{x} - d_2}{\|\mathbf{n}_2\|}$$
$$\frac{\mathbf{n}_1 \cdot \mathbf{x} - d_1}{\|\mathbf{n}_1\|} = -\frac{\mathbf{n}_2 \cdot \mathbf{x} - d_2}{\|\mathbf{n}_2\|}.$$

ESERCIZIO 1.1.5. Scrivere le equazioni delle bisettrici della seguente coppia di rette:

$$2x + 3y = 1$$
  

$$3x + 2y = 1.$$

#### 1.2. Cerchi nel piano

Il cerchio di centro c e raggio r > 0 è l'insieme dei punti x a distanza da c uguale a r, quindi la sua equazione è

$$\|\mathbf{x} - \mathbf{c}\| = r$$

cioè

$$(x_1 - c_1)^2 + (x_2 - c_2)^2 = r^2.$$
 (1.2.1)

L'equazione è quadratica, non lineare, e quindi esula dal dominio dell'algebra lineare, ma certamente non da quello della geometria: pertanto la studiamo brevemente qui.

Sviluppando i quadrati otteniamo:

$$x^{2} + y^{2} + 2c_{1}x + 2c_{2}y + c_{1}^{2} + c_{2}^{2} - r^{2} = 0.$$

Viceversa, consideriamo più generale l'equazione quadratica

$$x^2 + y^2 + a_1 x + a_2 y + b = 0.$$

Allora, ponendo

$$c_i = \frac{a_i}{2}$$

$$r^2 = c_1^2 + c_2^2 - b,$$
(1.2.2)

questa equazione quadratica si riconduce a (1.2.1) se  $b \leq c_1^2 + c_2^2$  (nel caso si abbia l'uguaglianza il cerchio consiste di un solo punto), ed invece non ha soluzioni reali se  $b > c_1^2 + c_2^2$ .

L'equazione parametrica del cerchio è

$$\begin{aligned} x &= c_1 + r \cos t \\ y &= c_1 + r \sin t \end{aligned} \tag{1.2.3}$$

dove  $0 \leq t < 2\pi$ .

#### 1.3. Esercizi di geometria analitica nel piano

In vari punti di questa Sezione useremo il seguente risultato, troppo banale per essere elencato come esercizio:

NOTA 1.3.1. Dati due vettori  $\mathbf{a} \in \mathbf{b}$ , il *punto medio* del segmento da essi sotteso è la loro media aritmetica  $\mathbf{c} = \mathbf{a} + \mathbf{b}$ . Analogamente, per ripartire il suddetto segmento in N parti, si considerano i punti intermedi  $\mathbf{c}_1, \ldots, \mathbf{c}_{N-1}$  definiti da  $\mathbf{c}_i = \mathbf{a} + \frac{i}{N}(\mathbf{b} - \mathbf{a} \text{ (per } i = 0 \text{ si ritrova } \mathbf{a} \text{ e per } i = N \text{ si ritrova } \mathbf{b}$ , per i valori intermedi di i si ottengono i vettori  $\mathbf{c}_i$ ).

#### 1.3.1. Rette.

ESERCIZIO 1.3.2. (Asse di un segmento, ovvero retta equidistante da due punti dati.) Trovare l'equazione della retta equidistante dai punti (1,2) e (2,4).

SUGGERIMENTO. La retta è quella che passa per il punto medio del segmento sotteso da (1, 2) e (2, 4) e con vettore direzionale perpendicolare a questo segmento.

ESERCIZIO 1.3.3. Trovare  $a \in c$  tali che le equazioni lineari

$$ax + y + 2 = 0$$
  
$$4x + 2y + c = 0$$

abbiano le stesse soluzioni. Qual è il significato geometrico della soluzione?

SVOLGIMENTO. Basta prendere a = 2 e c = 4 e le due equazioni diventano multiple l'una dell'altra, quindi coincidenti. Le equazioni rappresentano rette nel piano: con la scelta a = 2 e c = 4 le due rette coincidono. ESERCIZIO 1.3.4. Trovare  $a \in c$  tali che il sistema lineare

$$ax + y + 2 = 0$$
  
$$4x + 2y + c = 0$$

sia impossibile, e spiegare il significato geometrico della soluzione.

SVOLGIMENTO. Se si prende a = 2 e  $c \neq 4$  e le parti lineari delle due equazioni diventano multiple l'una dell'altra, ma i coefficienti costanti non rispettano la stessa proporzione, quindi il sistema è impossibile. Geometricamente, le due equazioni rappresentano due rette parallele ma non coincidenti, che quindi non si intersecano: le soluzioni del sistema sono proprio le intersezioni delle due rette, quindi in questo caso non esistono.

ESERCIZIO 1.3.5. Sia **r** la retta di equazione 3x + y = 1. Trovare:

- (i) l'equazione della retta simmetrica di  $\mathbf{r}$  rispetto all'asse delle x;
- (*ii*) l'equazione della retta simmetrica di  $\mathbf{r}$  rispetto all'asse delle y;
- (iii) l'equazione della retta simmetrica di  $\mathbf{r}$  rispetto all'origine.

SVOLGIMENTO.

(i) L'equazione della retta simmetrica rispetto all'asse x deve ottenersi cambiando x in -x nell'equazione di  $\mathbf{r}$ , quindi è -3x + y = 1.

Un modo più diretto (ma più lungo) di procedere è è il seguente. Per prima cosa si scrive **r** in forma parametrica. Un suo vettore normale è (3, 1), quindi un suo vettore direzionale è (1, -3), e **r** passa per (0, 1), quindi l'equazione è  $\mathbf{r}(t) = (0, 1) + t(1, -3)$ . A questo punto per ottenere l'equazione parametrica della retta riflessa basta cambiare il segno delle componenti x dei vettori. Dall'equazione parametrica si ricava facilmente quella cartesiana.

Un terzo modo è di determinare due punti per cui passa  $\mathbf{r}$ , ad esempio le intersezioni con gli assi (0, 1) e  $(\frac{1}{3}, 0)$ , ribaltare questi rispetto all'asse x in (0, -1) e  $(\frac{1}{3}, 0)$  e trovare l'equazione della retta per questi nuovi due punti.

- (*ii*) Ragionando come prima si trova l'equazione 3x y = 1.
- (*iii*) -3x y = -1, cioè 3x + y = -1, Si noti che questa è proprio la retta parallela alla precedente che passa alla stessa distanza dall'origine ma dall'altro lato.

ESERCIZIO 1.3.6. Trovare due rette perpendicolari che passano per il punto (2,3) e determinano sull'asse x un segmento

- (i) di lunghezza 4;
- (ii) di lunghezza minima.

Trovare la distanza minima di un punto  $(x_0, y_0)$  dall'asse x per la quale esistono due rette ortogonali che passano per  $(x_0, y_0)$  e staccano sull'asse x un segmento di lunghezza 4.

SVOLGIMENTO. Le rette che passano per (2,3) hanno equazione ax + by = 2a + 3b, con a, b fissati. Per ciascuna di tali rette, la famiglia di rette ad essa ortogonali si ottiene a partire da un vettore ortogonale al suo vettore normale (a, b): le equazioni di queste rette ortogonali sono bx - ay = d, con  $d \in \mathbb{R}$ . Fra queste, quella che passa per (2,3) è bx - ay = 2b - 3a. Le intersezioni delle due rette ortogonali con l'asse x si ottengono ponendo y = 0 nelle loro equazioni, e sono  $x_1 = \frac{2a+3b}{a}$  e  $x_2 = \frac{2b-3a}{b}$  (se a oppure b sono zero una delle due rette è parallela all'asse x e quindi non c'è intersezione: possiamo quindi assumere che entrambi i denominatori siano non nulli). Pertanto la lunghezza del segmento tagliato dalle due rette sull'asse x è

$$|x_2 - x_1| = \left|\frac{2b - 3a}{b} - \frac{2a + 3b}{a}\right| = 3\frac{a^2 + b^2}{|ab|}.$$
 (1.3.1)

Fissato un valore di a, questo segmento ha lunghezza 4 per il valore di b che risolve l'equazione  $3(a^2 + b^2) - 4ab = 0$ , la quale ha due soluzioni reali perché il discriminante  $16a^2 - 12a^2$  è positivo: questo risolve la parte (i) dell'esercizio. Per la parte (ii), osserviamo che la lunghezza minima del segmento si ricava prendendo il minimo rispetto ad a e b del lato di destra di (1.3.1). Ora,  $a^2 + b^2 = (a - b)^2 + 2ab$ , e quindi il lato di destra di (1.3.1) si riscrive come  $2 + \frac{(a-b)^2}{|ab|}$ : evidentemente questa espressione ha minimo per b = a.

Infine, le due rette ortogonali che passano per (x, y) si scrivono come prima:  $ax + by = ax_0 + by_0$  e  $bx - ay = bx_0 - ay_0$ . La lunghezza del segmento staccato sull'asse x da queste due rette si ricava come in (1.3.1), e vale

$$\left|\frac{bx_0 - ay_0}{b} - \frac{ax_0 + by_0}{a}\right| = |y_0|\frac{a^2 + b^2}{|ab|}$$

ed il minimo di queste distanze si trova imponendo che l'equazione quadratica  $|y_0|(a^2+b^2) = 4|ab|$ , per un dato a, abbia una sola soluzione in b. Questo equivale ad imporre che il discriminante sia nullo. Possiamo assumere  $y_0 > 0$  (ci sono due soluzioni simmetriche, una con  $y_0 > 0$  e l'altra data da  $-y_0$ . lasciamo i calcoli al lettore.

#### 1.3.2. Cerchi.

ESERCIZIO 1.3.7. Trovare l'equazione del cerchio che passa per i punti  $(1, 2), (0, 0) \in (3, 3).$ 

SUGGERIMENTO. Si considerano due delle rette equidistanti da coppie di questi punti, come nel precedente Esercizio 1.3.2. Intersecando queste due rette troviamo il centro del cerchio; allora il raggio è uguale alla distanza fra questo centro ed uno qualsiasi dei vertici.

Presentiamo un metodo alternativo geometricamente meno elegante, ma che talvolta richiede meno passaggi. Individuiamo il centro (x, y)della sfera imponendo che sia equidistante dai tre punti assegnati. nel caso presente, questo significa imporre le condizioni

$$(x-1)^{2} + (y-2)^{2} = x^{2} + y^{2} = (x-3)^{2} + (y-3)^{2}$$

dalle quali i termini quadratici si cancellano dopo che si sviluppano i quadrati, e quello che resta è il sistema lineare

$$-2x - 4y + 2 = 0 = -6x - 6y + 18$$

cioè

$$\begin{array}{rcl} x+2y &=& 1\\ x+y &=& 3 \,. \end{array}$$

Il centro del cerchio è la soluzione di questo sistema lineare, data da x = 5, y = -2. Geometricamente questa soluzione rappresenta il punto di intersezione delle due rette x + 2y = 1 e x + y = 3. È facile verificare che la prima di queste due rette è proprio quella equidistante dai punti (1, 2) e (0, 0), e la seconda è equidistante da (0, 0) e (1, 2). Perciò questo secondo metodo in effetti equivale a quello precedente.

ESERCIZIO 1.3.8. Trovare le tangenti al cerchio  $x^2+y^2-2x+2y-1=0$  che si incontrano nel punto (4,4).

SUGGERIMENTO. Scrivere l'equazione della generica retta che passa per (4,4) e trovare le intersezioni con il cerchio (mettendo a sistema le due equazioni). I punti di intersezione sono le soluzioni dell'equazione quadratica così ricavata. A seconda del fatto che il discriminante di questa equazione sia positivo, nullo o negativo si ottengono due, una o nessuna soluzione. Le rette tangenti sono quelle che corrispondono ad una sola soluzione: se ne trovano esattamente due.

Un metodo alternativo (ma di fatto equivalente) consiste nello scrivere in forma parametrica, grazie a (1.2.3), i punti del cerchio, che nel caso presente, in base a (1.2.2), è  $\mathbf{r}t = (1 + \cos t, 1 + \sin t)$ , ed il vettore dal centro del cerchio al punto generico su di esso (che evidentemente è  $(\cos t, \sin t)$ , e trovare i valori del parametro t per cui questo vettore sia perpendicolare al vettore dal punto (4, 4) al punto del cerchio  $\mathbf{r}t$ , cioè il vettore  $(-3 + \cos t, -3 + \sin t)$ .

ESERCIZIO 1.3.9. Scrivere l'equazione della retta tangente nell'origine al cerchio con centro in (2, 2) e passante per l'origine.

SVOLGIMENTO. La retta tangente è quella perpendicolare al raggio, cioè al segmento dal centro (1, 12) al punto di tangenza (0, 0). Questo raggio giace lungo la retta y = x, quindi la tangente è la sua perpendicolare y = -x.

ESERCIZIO 1.3.10. Scrivere l'equazione della tangente nell'origine al cerchio  $(x - 1)^2 + (y - 2)^2 = 5$ .

SVOLGIMENTO. Il raggio dal centro (1, 2) all'origine è  $\mathbf{r} = -(1, 2)$ . La retta tangente è ortogonale al raggio e passa per l'origine, quindi il suo vettore normale è  $\mathbf{r}$  e l'equazione è 2x - y = 0.

ESERCIZIO 1.3.11. Scrivere l'equazione del cerchio passante per l'origine e tangente alla retta x + 3y = 4 nel punto (1, 1).

SUGGERIMENTO. Il centro del cerchio si trova sulla retta perpendicolare a x+3y = 4 nel punto (1, 1), cioè sulla retta 3y-x = 2. Il quadrato della distanza dei punti di questa retta da (1, 1) è  $(x - 1)^2 + (y - 1)^2$ ; il fatto che il cerchio passi anche per l'origine equivale a richiedere che tale distanza sia uguale a quella dall'origine, cioè che si abbia  $(x - 1)^2 + (y - 1)^2 = x^2 + y^2$ . Il centro del cerchio verifica quindi questa equazione quadratica e l'equazione lineare 3y - x = 2.  $\Box$ 

ESERCIZIO 1.3.12. Scrivere l'equazione del cerchio passante per il punto (3,4) e tangente alle rette x + 2y = 1 e 3x - y = 2. Dal punto di intersezione di queste due rette, qual è l,angolo sotteso dal cerchio?

SUGGERIMENTO. Il centro del cerchio è equidistante dalle due rette e dal punto. Quindi lo si trova calcolando le proiezioni ortogonali di un generico punto (x, y) sulle due rette, trovandone le rispettive distanze ed uguagliandole fra loro ed alla distanza da (3, 4), cioè alla radice quadrata di  $(x - 3)^2 + (y - 4)^2$ .

L'angolo sotteso, ovviamente, è quello formato dalle due rette, ossia l'arcocoseno deel prodotto scalare dei loro versori normali (si faccia attenzione a prendere i versi concordi!).

#### CAPITOLO 2

# Vettori e geometria euclidea ed analitica in $\mathbb{R}^3$

Alcuni esempi geometrici nello spazio tridimensionale sono analoghi a quelli dati nel piano nel Capitolo 1, ma, anche quando la trattazione è pressoché identica parola per parola, per completezza la riportiamo di nuovo nel presente contesto.

#### **2.1.** L'equazione del piano in $\mathbb{R}^3$

• L'equazione cartesiana di un piano in tre dimensioni reali è

$$ax_1 + bx_2 + cx_3 = d \tag{2.1.1}$$

dove  $a, b, c \in d$  sono costanti reali.

1

Se poniamo  $\mathbf{n} = (a, b, c)$  e  $\mathbf{x} = (x_1, x_2, x_3)$ , l'equazione del piano si scrive

$$\mathbf{n} \cdot \mathbf{x} = d \,. \tag{2.1.2}$$

In questa forma l'equazione vale anche in qualsiasi dimensione, cioè per iperpiani di codimensione 1 in  $\mathbb{R}^n$ . Se d = 0, allora il piano passa per l'origine, ed il vettore **n** è perpendicolare ai punti del piano. Se invece  $d \neq 0$  allora il piano è formato dai vettori la cui proiezione sulla direzione di **n** vale  $d\frac{\mathbf{n}}{\|\mathbf{n}\|}$ , cioè è il piano parallelo a quello precedente a distanza  $\frac{d}{\|\mathbf{n}\|}$  in direzione **n**.

In particolare, se tre punti **u**, **q** e **r** non sono allineati, cioè non multipli uno dell'altro, allora sostituendo le loro componenti nell'equazione  $ax_1 + bx_2 + cx_3 = d$  otteniamo un sistema lineare di tre equazioni nelle incognite  $a, b, c \in d$ . Questo sistema ha un grado di incertezza, essendo un sistema con una incognita in più del numero di equazioni, ma per ogni valore di d ha un'unica soluzione. Quindi, a meno di multipli, c'è un'unica soluzione, che ci dà l'equazione cartesiana del piano.  $\tilde{E}$ del resto ovvio che, per ogni costante  $\alpha$  non nulla, le equazioni  $ax_1 + bx_2 + cx_3 = d e \alpha ax_1 + \alpha bx_2 + \alpha cx_3 = \alpha d$  descrivono lo stesso piano (si noti anche che i due vettori normali sono multipli uno dell'altro).

- L'equazione normale del piano si ottiene normalizzando il vettore perpendicolare in (2.1.2):  $\mathbf{n} \cdot \mathbf{x}/||\mathbf{n}|| = d/||\mathbf{n}||$ . La distanza dall'origine è quindi  $d/||\mathbf{n}||$ .
- L'equazione parametrica del piano in tre dimensioni che contiene le due rette passanti per un punto  $\mathbf{u}$  e dirette lungo i vettori rispettivamente  $\mathbf{v} \in \mathbf{w}$  è

$$\mathbf{p}(s,t) = \mathbf{u} + s\mathbf{v} + t\mathbf{w}$$

dove s, t sono due parametri reali. In particolare, se tre punti **u**, **q** e **r** non sono allineati, allora **q** – **u** e **r** – **u** non sono multipli uno dell'altro, e l'equazione parametrica

$$\mathbf{p}(s,t) = \mathbf{u} + s(\mathbf{q} - \mathbf{u}) + t(\mathbf{r} - \mathbf{u})$$

determina un piano, l'unico piano che contiene i tre punti.

ESEMPIO 2.1.1. (Il piano bisettore di un segmento, cioè equidistante da due punti.) Siano  $\mathbf{a}$  e  $\mathbf{b}$  due vettori distinti, e  $\mathbf{m} = \frac{1}{2}(\mathbf{a} + \mathbf{b})$ il loro punto medio, come nella Nota 1.3.1. Il piano equidistante da  $\mathbf{a}$  e  $\mathbf{b}$  è quello che taglia perpendicolarmente a metà il segmento che li congiunge, cioè è ortogonale a  $\mathbf{b} - \mathbf{a}$  e passa per  $\mathbf{m}$ . Quindi questo piano ha per vettore normale il vettore  $\mathbf{n} = \mathbf{b} - \mathbf{a}$ : la sua equazione è

$$\mathbf{x} \cdot \mathbf{n} = \mathbf{m} \cdot \mathbf{n} = \frac{1}{2} (\mathbf{a} + \mathbf{b}) \cdot (\mathbf{b} - \mathbf{a}) = \frac{1}{2} (\|\mathbf{a}\|^2 + \|\mathbf{b}\|^2).$$

ESERCIZIO 2.1.2. Scrivere l'equazione (cartesiana e parametrica) del piano che passa per i punti (0, 1, 2), (3, -1, 1), (1, 1, 3).

ESERCIZIO 2.1.3. Scrivere l'equazione del piano bisettore del segmento da (1,0,0) a (0,0,1).

ESERCIZIO 2.1.4. In quali punti il piano 2x + y + 3z = 4 tagloia gli assi coordinati?

SUGGERIMENTO. Per trovare l'intersezione con l'asse x basta porre y = z = 0 nell'equazione del piano, ed analogamente per le intersezioni con gli altri due assi.

#### 2.2. L'equazione della retta in $\mathbb{R}^3$

Anche qui, come prima, abbiamo due modi di scrivere l'equazione di una retta:

92

• Equazione parametrica: questo modo è identico a quello visto in due dimensioni. La retta che passa per il punto  $\mathbf{p}$  ed ha la direzione del vettore  $\mathbf{v}$  ha equazione

$$\mathbf{r}(t) = \mathbf{p} + t\mathbf{v}$$

 $\operatorname{con} t \in \mathbb{R}.$ 

Osserviamo che, se la retta passa per due punti  $\mathbf{p}$  e  $\mathbf{q}$ , allora la sua equazione parametrica è

$$\mathbf{r}(t) = \mathbf{p} + t(\mathbf{q} - \mathbf{p})$$

 $\operatorname{con} t \in \mathbb{R}.$ 

• Equazione cartesiana: Si noti che l'analogo tridimensionale dell'equazione cartesiana (1.1.3) della retta in  $\mathbb{R}^2$  dovrebbe essere

$$ax_1 + bx_2 + cx_3 = d$$

dove  $a, b, c \in d$  sono costanti reali, ma questa non è l'equazione di una retta in  $\mathbb{R}^3$ , bensí di un piano, come visto sopra in (2.1.1). Per determinare l'equazione cartesiana di una retta basta osservare che, in tre dimensioni, ogni retta si può interpretare come l'intersezione di due piani (non paralleli, e non univocamente determinati), e quindi invece di un'equazione come (2.1.1) abbiamo un sistema di due equazioni:

$$a_1x_1 + b_1x_2 + c_1x_3 = d_1$$
  
$$a_2x_1 + b_2x_2 + c_2x_3 = d_2.$$

Quindi l'equazione della retta è un sistema di due equazioni lineari, cioè una equazione vettoriale:

$$\begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} .$$
(2.2.1)

I vettori  $(a_1, b_1, c_1)$  e  $(a_2, b_2, c_2)$  sono perpendicolari ai due piani rispettivi, e poiché tali piani non sono paralleli i due vettori determinano un unico (a meno di multipli) terzo vettore perpendicolare ad entrambi, che quindi è il vettore di direzione della retta.

ESEMPIO 2.2.1. (La retta intersezione di due piani.) Ogni coppia di piani non paralleli definisce una ed una sola retta di intersezione. Se le equazioni dei due piani sono  $a_1x + b_1y + c_1z = d_1$  e  $a_2x + b_2y + c_2z =$  $d_1$  allora essi sono paralleli esattamente quando i loro vettori normali  $(a_1, b_1, c_1) \in (a_2, b_2, c_2)$  sono multipli uno dell'altro. In tal caso il sistema delle due equazioni dei piani è l'equazione cartesiana della retta.

Per scrivere l'equazione parametrica della retta occorre trovare il suo vettore direzionale. Come osservato più sopra, questo vettore  $\mathbf{v}$  è, a meno di multipli, quello che completa una terna ortogonale i cui primi due vettori sono i vettori perpendicolari ai due piani, quindi si ottiene dal procedimento di ortogonalizzazione di Gram-Schmidt (Sezione ??), o equivalentemente ma più velocemente calcolando il prodotto vettoriale dei due vettori (Sezione ??). A questo punto per trovare l'equazione parametrica della retta basta trovare un qualsiasi suo punto  $\mathbf{p}$ : l'equazione della retta è  $\mathbf{r}(t) = \mathbf{p} + t\mathbf{v}$ . Per trovare  $\mathbf{p}$  occorre trovare almeno una soluzione del sistema (2.2.1), e quindi, ad esempio, tramite il metodo di eliminazione di Gauss.

È interessante il caso in cui i due piani sono assegnati tramite due terne di punti che vi appartengono. Abbiamo visto come determinare le equazioni dei due piani, e quindi un vettore direzionale **v** della retta. Il caso più comodo in questo contesto è quello in cui le due terne di punti ne contengono uno in comune: allora questo punto **p** appartiene ad entrambi i piani e l'equazione parametrica della retta è  $\mathbf{r}(t) = \mathbf{p} + t\mathbf{v}$ .

# ESERCIZIO 2.2.2. (Il fascio di piani che contengono una retta assegnata.)

- (i) Sia **r** una retta espressa in forma parametrica,  $\mathbf{r}(t) = \mathbf{p} + t\mathbf{v}$ . Come si scrivono le equazioni dei piani  $\pi$  che la contengono?
- (*ii*) Ora esprimiamo **r** in forma cartesiana, come intersezione di due piani,  $\pi_0$  : { $(x_1, x_2, x_3)$  :  $a_0x_1 + b_0x_2 + c_0x_3 = d_0$ } e  $\pi_1$  : { $(x_1, x_2, x_3)$  :  $a_1x_1 + b_1x_2 + c_1x_3 = d_1$ }. Come si scrive ora il fascio di piani che contiene questa retta?

SVOLGIMENTO. La prima parte è facile: i piani che contengono la retta sono quelli che contengono un punto della retta, ad esempio  $\mathbf{p}$ , ed hanno vettore normale  $\mathbf{n}$  perpendicolare al vettore direzionale della retta, ossia  $\mathbf{v}$ . Scrivendo  $\mathbf{n} = (n_1, n_2, n_3)$ ,  $\mathbf{p} = (p_1, p_2, p_3)$  e  $\mathbf{x} = (x_1, x_2, x_3)$ troviamo l'equazione  $\langle \mathbf{n}, \mathbf{x} \rangle = \langle \mathbf{n}, \mathbf{p} \rangle$ , ovvero

$$n_1x_1 + n_2x_2 + n_3x_3 = n_1p_1 + n_2p_2 + n_3p_3.$$
 (2.2.2)

Per la seconda parte, basta prendere i vettori normali ai due piani, diciamo  $\mathbf{n}_0 = (a_0, b_0, c_0) \in \mathbf{n}_1 = (a_1, b_1, c_1)$ , e considerare i piani con vettore normale dato da una combinazione lineare di  $\mathbf{n}_0 \in \mathbf{n}_1$ : infatti i vettori normali dei piani appartenenti al fascio di piani che contengono  $\mathbf{r}$  generano il piano perpendicolare a  $\mathbf{r}$  e quindi sono combinazioni lineari di due qualsiasi indipendenti di essi. Allora, sia  $\lambda = (\lambda_0, \lambda_1)$  e consideriamo il vettore normale  $\mathbf{n}_{\lambda} = \lambda_0 \mathbf{n}_0 + \lambda_1 \mathbf{n}_1$ . Poniamo  $d_{\lambda} = \lambda_0 d_0 + \lambda_1 d_1$ e  $\mathbf{x} = (x_1, x_2, x_3)$ . Allora il piano che passa per  $\mathbf{r}$  ed ha  $\mathbf{n}_{\lambda}$  come vettore normale è quello con equazione  $\langle \mathbf{n}_{\lambda}, \mathbf{x} \rangle = d_{\lambda}$ . Infatti questo piano contiene la retta data, perché il sistema delle due equazioni lineari  $\langle \mathbf{n}_{\lambda}, \mathbf{x} \rangle = d_{\lambda}$  e  $\langle \mathbf{n}_1, \mathbf{x} \rangle = d_1$  è equivalente al sistema originale  $\langle \mathbf{n}_0, \mathbf{x} \rangle = d_0, \langle \mathbf{n}_1, \mathbf{x} \rangle = d_1$ , purché  $\mathbf{n}_0$  e  $\mathbf{n}_1$  siano linearmente indipendenti (come si vede subito sottraendo la seconda equazione dalla prima).

ESERCIZIO 2.2.3. (Il piano che contiene una retta assegnata ed è perpendicolare ad un'altra retta data.) Date due rette in forma parametrica,  $\mathbf{r}(t) = \mathbf{p} + t\mathbf{v} \in \mathbf{s}(t) = \mathbf{q} + t\mathbf{u}$ , mostrare che esiste un piano che contiene  $\mathbf{r}$  ed è perpendicolare a  $\mathbf{s}$  se e solo se i vettori direzionali  $\mathbf{v} \in \mathbf{u}$  sono ortogonali.

SVOLGIMENTO. Senza perdita di generalità possiamo supporre, a meno di una traslazione, che la retta  $\mathbf{r}$  passi per l'origine, ossia  $\mathbf{p} = \mathbf{o}$ : infatti la traslazione non cambia il vettore direzionale della retta. Esattamente per lo stesso motivo possiamo supporre che anche  $\mathbf{s}$  passi per l'origine: basta rimpiazzare  $\mathbf{s}$  con la sua parallela che passa per l'origine ed osservare che se un piano è perpendicolare a questa retta parallela è anche perpendicolare alla retta originale. Ora le due rette si incontrano nell'origine. Se i loro vettori direzionali  $\mathbf{v} \in \mathbf{u}$  sono ortogonali, allora il piano per l'origine con vettore normale  $\mathbf{u}$  contiene il vettore  $\mathbf{v}$  e quindi la retta  $\mathbf{r}(t) = t\mathbf{v}$ . Viceversa, ogni piano che contiene  $\mathbf{v}$  e quindi la retta  $\mathbf{r}(t) = t\mathbf{v}$  deve avere vettore normale ortogonale a  $\mathbf{v}$ . Un tale piano è perpendicolare a  $\mathbf{s}(t) = t\mathbf{u}$  se e solo se il suo vettore normale è multiplo di  $\mathbf{u}$ , e pertanto  $\mathbf{v}$  deve essere ortogonale a  $\mathbf{u}$ .

ESERCIZIO 2.2.4. I vettori  $\mathbf{v} = (1, 0, 0)$  e  $\mathbf{u} = (0, 1, 1)$  sono ortogonali. Trovare l'equazione del piano  $\pi$  che contiene la retta  $\mathbf{r}(t) = (0, 1, 0) + t\mathbf{v}$ ed è perpendicolare alla retta  $\mathbf{s}(t) = (0, 0, 1) + t\mathbf{u}$ .

SVOLGIMENTO. In base al precedente Esercizio 2.2.3, il vettore normale a  $\pi$  è **u**. Ioltre  $\pi$  deve contenere **r**, ovvero deve contenere il punto **p** (poiché  $\pi$  ha per vettore normale un vettore ortogonale a **v**, e quindi contiene **v**, quest'ultima asserzione equivale a dire che  $\pi$  contiene tutta

la retta  $\mathbf{r}$ , in base alla parte (*i*) dell'Esercizio 2.2.2). Pertanto conosciamo un punto contenuto in  $\pi$  ed il vettore normale di  $\pi$ : ora l'equazione cartesiana di  $\pi$  si trova come in (2.2.2).

#### 2.3. Proiezioni e distanze fra piani

ESEMPIO 2.3.1. (Proiezione ortogonale di un punto su un piano.) Il problema della proiezione ortogonale di un punto  $\mathbf{x} \in \mathbb{R}^3$  su un piano  $\pi$  si risolve in maniera analoga al caso bidimensionale trattato nell'Esempio 1.1.2. Ci sono quindi due modi:

- Tramite l'equazione parametrica della retta perpendicolare. Sia  $ax_1 + bx_2 + cx_3 = d$  l'equazione del piano  $\pi$ . Allora un vettore normale al piano è  $\mathbf{n} = (a, b, c)$ , e l'equazione parametrica della retta che passa per  $\mathbf{x}$  ed è perpendicolare a  $\pi$  è  $\mathbf{r}(t) =$   $\mathbf{x}+t\mathbf{n}$ . Imponendo che  $\mathbf{r}(t)$  soddisfi l'equazione  $ax_1+bx_2+cx_3 =$  d si trova il valore  $t_0$  del parametro t per cui  $\mathbf{r}(t)$  giace in  $\pi$ , quindi il piede  $\mathbf{p} := \mathbf{r}(t_0)$  della proiezione ortogonale di  $\mathbf{x}$  sul piano  $\pi$ .
- Tramite la decomposizione ortogonale di Gram-Schmidt. Come prima, si considera il vettore normale n = (a, b, c), ed ora lo si normalizza (per semplicità supponiamo che sia già di norma 1). Supponiamo dapprima che il piano passi per l'origine, cioè che si abbia d = 0. Allora questo piano è un sottospazio bidimensionale: in esso si costruisce una base ortonormale v<sub>1</sub>, v<sub>2</sub> tramite il procedimento di Gram-Schmidt. La terna v<sub>1</sub>, v<sub>2</sub>, n è una base ortonormale in R<sup>3</sup>. Ora la proiezione ortogonale p di x su π si ottiene da (??), sottraendo le componenti lungo π:

$$\mathbf{p} = \mathbf{x} - \mathbf{x} \cdot \mathbf{v}_1 \mathbf{v}_1 - \mathbf{x} \cdot \mathbf{v}_2 \mathbf{v}_2 \,.$$

Si osservi che non c'è bisogno di prendere una base ortonormale in  $\pi$ : qualunque base va bene, però le componenti di  $\mathbf{x}$  lungo questi vettori di base si scrivono in maniera più complicata, come nell'uguaglianza (??).

Ora veniamo al caso generale in cui il piano  $\pi$  non passa per l'origine, cioè  $d \neq 0$ :  $\pi$  passa a distanza |d| dall'origine. In questo caso il procedimento precedente ci dà la proiezione  $\mathbf{p}'$  di  $\mathbf{x}$  non su  $\pi$  ma sul piano  $\pi'$  ad esso parallelo che passa per l'origine. Ora la proiezione  $\mathbf{p}$  su  $\pi$  si ottiene spostandosi perpendicolarmente a  $\pi'$  di una distanza |d|:

$$\mathbf{p} = \mathbf{p}' + d\mathbf{n}$$

(nel secondo membro abbiamo preso la somma invece della differenza perché, se ad esempio a, b, c > 0 allora il piano  $\pi$  è spostato verso l'ottante positivo rispetto a  $\pi'$  se d > 0 e verso quello negativo (cioè con tutte e tre le coordinate negative) altrimenti. Quindi per spostarsi da  $\pi$  a  $\pi'$  bisogna muoversi nella direzione di  $\mathbf{n}$  se d > 0 e al contrario se d < 0.

In particolare, se un piano ha equazione  $\mathbf{n} \cdot \mathbf{x} = d$  (e quindi equazione normale  $\mathbf{n} \cdot \mathbf{x} / ||\mathbf{n}|| = d / ||\mathbf{n}||$ ), la distanza di un punto  $\mathbf{p}$  dal piano è

$$\left|\frac{\mathbf{n} \cdot \mathbf{p} - d}{\|\mathbf{n}\|}\right| \,. \tag{2.3.1}$$

Infatti, per quanto osservato sopra, questa è la differenza fra le distanze dall'origine del piano dato e del suo parallelo che passa per  $\mathbf{p}$ .

In particolare, l'equazione del piano con vettore normale  ${\bf n}$  che contiene un dato punto  ${\bf p}$  è

$$\mathbf{n} \cdot \mathbf{x} = \mathbf{n} \cdot \mathbf{p} \,. \tag{2.3.2}$$

ESERCIZIO 2.3.2. Trovare la distanza fra il punto (1, 1, 1) ed il piano x + 3y - z = 2.

ESERCIZIO 2.3.3. Trovare la distanza fra i due piani paralleli seguenti:

$$\begin{array}{rcl} x + 3y - 2z &=& 1 \\ x + 3y - 2z &=& 4 \,. \end{array}$$

г	
L	
L	

ESEMPIO 2.3.4. (Proiezione ortogonale di un punto su una retta in  $\mathbb{R}^3$ .) Anche questo problema si può risolvere in due modi equivalenti.

- Tramite l'equazione parametrica della retta perpendicolare.
- Tramite la decomposizione ortogonale di Gram-Schmidt.

ESERCIZIO 2.3.5. Trovare la distanza del punto (1, 2, 3) dalla retta intersezione dei due piani 2x + y - z = 0 e x - 2y + z = 1.

#### 2.4. Sfere

Il modo di descrivere sfere tridimensionali tramite equazioni quadratiche è identico a quello già presentato per i cerchi nel piano (Sezione 1.2). Lo ripetiamo in questo ambiente più generale.

La sfera di centro  $\mathbf{c}$  e raggio r > 0 è l'insieme dei punti  $\mathbf{x}$  a distanza da  $\mathbf{c}$  uguale a r, quindi l'equazione della sfera è

$$\|\mathbf{x} - \mathbf{c}\| = r$$

cioè

$$\sum_{i=1}^{3} (x_i - c_i)^2 = r^2.$$
(2.4.1)

Sviluppando i quadrati nell'equazione della sfera (2.4.1) otteniamo:

$$\sum_{i=1}^{3} x_i^2 + 2\sum_{i=1}^{3} c_i x_i + \sum_{i=1}^{3} c_i^2 - r^2 = 0.$$

Viceversa, consideriamo più generale l'equazione quadratica

$$\sum_{i=1}^{3} x_i^2 + \sum_{i=1}^{3} a_i x_i + b = 0.$$

Allora, ponendo  $c_i = \frac{a_i}{2} e r^2 = \sum_{i=1}^3 c_i^2 - b$ , questa equazione quadratica si riconduce a (2.4.1) se  $b \leq \sum_{i=1}^3 c_i^2$  (nel caso si abbia l'uguaglianza la sfera consiste di un solo punto), ed invece non ha soluzioni reali se  $b > \sum_{i=1}^3 c_i^2$ .

Se invece di equazioni del tipo  $\sum_{i=1}^{3} (x_i - c_i)^2 = r^2$  scegliamo tre coefficienti positivi  $b_i$  e consideriamo l'equazione

$$\sum_{i=1}^{3} b_i x_i^2 + 2 \sum_{i=1}^{3} c_i x_i + \sum_{i=1}^{3} c_i^2 - r^2 = 0,$$

possiamo riportare questa equazione alla precedente tramite un cambiamento di scala su ciascun asse. Quindi le soluzioni di questa equazione sono ellissi con gli assi principali paralleli agli assi coordinati. Tramite una rotazione ci riportiamo al caso generale di ellissi con gli assi principali disposti in altre orientazioni (ma ovviamente consistenti di una terna ortogonale, anche se non si incrociano nell'origine).

#### 2.5. Esercizi di geometria analitica in $\mathbb{R}^3$

ESERCIZIO 2.5.1. Siano  $\mathbf{p} = (0, 0, 1)$  e  $\mathbf{q} = (2, 2, 3)$ . Per quali valori delle costanti reali *a* e *d* la retta che passa per  $\mathbf{p} \in \mathbf{q}$  è parallela al piano ax + y + z + d = 0?

SVOLGIMENTO. Basta imporre che il vettore normale al piano sia ortogonale al vettore di direzione della retta, che è (2, 2, 2). Quindi l'equazione del piano è un multiplo di 2x + 2y + 2z + d = 0 (la costante dnon viene determinata, perché la condizione sul piano riguarda solo il suo vettore normale, non la distanza dall'origine: cambiando d il piano si trasforma in un altro piano parallelo al primo.) Pertanto la risposta è a = 1, d arbitrario.

In alternativa, ma con molti più calcoli non necessari, si [uò scrivere l'equazione della retta e metterla a sistema con quella del piano, poi imporre che non ci siano soluzioni.

ESERCIZIO 2.5.2. Scrivere l'equazione del piano parallelo all'asse z che interseca l'asse x nel punto (2, 0, 0) e l'asse y in (0, 1, 0).

SVOLGIMENTO. Il piano è parallelo all'asse z se e solo se il suo vettore normale è perpendicolare a  $\mathbf{e}_3 = (0, 0, 1)$ , e quindi del tipo (a, b, 0). Pertanto l'equazione del piano è della forma ax + by + d = 0. L'intersezione con l';asse x deve quindi essere in un punto (x, 0, 0) che soddisfa ax + d = 0: poiché si richiede che questa intersezione sia in (2, 0, 0) dobbiamo avere 2a + d = 0. Con lo stesso ragionamento per l'intersezione con l'asse y si ottiene b + d = 0. Da queste due equazioni ricaviamo 2a = b = -d: quindi il piano ha equazione ax + 2ay - 2a = 0. Poiché il piano non cambia se dividiamo tutti i coefficienti dell'equazione per la stessa costante, possiamo scrivere per il piano l'equazione x+2y-2=0.

ESERCIZIO 2.5.3. (*Piani che contengono una retta assegnata.*) Scrivere l'equazione dei piani che contengono la retta x = 2t, y = t, z = t + 1, dove t è una variabile reale.

SVOLGIMENTO. Il vettore direzionale della retta è  $\mathbf{d} = (2, 1, 1, )$ . I piani che contengono la retta hanno vettore normale ortogonale a  $\mathbf{d}$ , quindi sono quelli del tipo ax + by + cz + d = 0 dove  $(a, b, c) \perp (2, 1, 1)$ (cioè 2a + b + c = 0) che contengono un punto della retta, ad esempio il punto (0, 0, 1) (ottenuto ponendo t = 0 nell'equazione parametrica della retta). Dall'ultima condizione segue c + d = 0, cioè d = -c. Pertanto i piani richiesti soddisfano l'equazione ax + by + cz - c = 0con la condizione 2a + b + c = 0: una famiglia a due parametri di piani.

ESERCIZIO 2.5.4. (*Piani che contengono una retta data come intersezione di due piani fissati.*) Scrivere l'equazione dei piani che

contengono la retta intersezione dei piani x+y+z = 0 e x+2y+3z-1 = 0.

SVOLGIMENTO. Fissiamo un punto nell'intersezione dei due piani, cioè nella retta, diciamo  $\mathbf{p} = (-1, 1, 0)$  (ottenuto scegliendo z = 0 nelle equazioni dei due piani, e mettendole a sistema).

Non occorre trovare l'equazione parametrica della retta: il suo vettore direzionale **d** è perpendicolare ai vettori normali dei due piani assegnati, cioè ai vettori  $\mathbf{n}_1 = (1, 1, 1)$  e  $\mathbf{n}_2 = (1, 2, 3)$ . Ora, come nel precedente Esercizio 2.5.3, i piani che contengono la retta hanno vettore normale **n** ortogonale a **d**, e quindi appartenente al piano generato da  $\mathbf{n}_1 \in \mathbf{n}_2$  (perché in  $\mathbb{R}^3$  questo piano è il complemento ortogonale di **d**: ma come si risolverebbe lo stesso esercizio in  $\mathbb{R}^n$ ?). Pertanto  $\mathbf{n} = s\mathbf{n}_1 + t\mathbf{n}_2 = (s + t, s + 2t, s + 3t)$  per qualche s,  $t \in \mathbb{R}$ . Quindi i piani cercati sono la famiglia a due parametri di equazione

$$\mathbf{n} \cdot (x, y, z) = (s+t)x + (s+2t)y + (s+3t)z = \mathbf{n} \cdot \mathbf{p} = t$$

Ponendo s + t = u riscriviamo la famiglia a due parametri di equazioni nel seguente modo leggermente più semplice:

$$ux + (u+t)y + (u+2t)z - t = 0.$$

ESERCIZIO 2.5.5. (Piano che contiene un punto assegnato ed una retta data come intersezione di due piani fissati.) Scrivere l'equazione del piano che contiene il punto  $\mathbf{p} = (1, 1, -2)$  e la retta intersezione dei piani  $\sigma$  di equazione x + y + z = 0 e  $\tau$  di equazione x + 2y + 3z - 1 = 0. Si risolva lo stesso problema nel caso in cui, invece di  $\mathbf{p}$ , si prende il punto  $\mathbf{q} = (-1, 1, 0)$ .

SVOLGIMENTO. Dal precedente Esercizio 2.5.4 sappiamo che i piani che passano per la retta data verificano l'equazione

$$ux + (u+t)y + (u+2t)z - t = 0$$

dove  $u \in t$  sono due parametri reali. Fra questi piani, quindi, quello che passa anche per il punto (1, 1, -2) soddisfa u + (u+t) - 2(u+2t) - t = 0, cioè -5t = 0, ossia t = 0. L'equazione è ux + uy + uz = 0: poiché il piano rimane invariato se riscaliamo i coefficienti dell'equazione, essa si può riscrivere come z + y + z = 0.

Si noti che il punto **p** appartiene già al piano  $\sigma$ , e quindi il piano cercato è proprio  $\sigma$ . Invece, come già osservato nel precedente Esercizio 2.5.4, il punto  $\mathbf{q} = (-1, 1, 0)$  appartiene ad entrambi i piani  $\sigma \in \tau$ , e quindi alla retta data dalla loro intersezione. Perciò imporre che fra i vari piani che contengono tale retta se ne scelga uno che passa per

**q** non comporta nessuna restrizione: tutti lo fanno. Ed infatti, se si ripete il ragionamento precedente, imponendo ora che u e t siano tali che l'equazione ux + (u + t)y + (u + 2t)z - t = 0 sia soddisfatta dal punto (x, y, z) = (-1, 1, 0), otteniamo l'identità 0 = 0 invece che una equazione in u e t.

ESERCIZIO 2.5.6. Trovare l'equazione del piano che passa per la retta di equazione parametrica  $\mathbf{r}(t) = (1 + t, 2t, t - 1)$  ed è perpendicolare al piano x + y + 2z - 1 = 0

SVOLGIMENTO. Come nel precedente Esercizio ??, imponiamo che il vettore normale  $\mathbf{n} := (a, b, c)$  del piano cercato sia ortogonale al vettore direzionale  $\mathbf{d} = (1, 2, 1)$  della retta, e che passi per un suo punto, diciamo  $\mathbf{p} = (1, 0, -1)$ . Inoltre, ora richiediamo che  $\mathbf{n}$  sia anche ortogonale al vettore normale del piano x + y + 2z - 1 = 0, cioè a (1, 1, 2). Le condizioni imposte determinao  $\mathbf{n}$  come la soluzione (a meno di multipli) del sistema a + 2b + c = 0, a + b + 2c = 0, cioè a = c, b = -c. Quindi un vettore normale al piano cercato è  $\mathbf{n} = (1, -1, 1)$ , e l'equazione (2.3.2) di questo piano diventa x - y + z = 0.

ESERCIZIO 2.5.7. Trovare l'equazione della retta **r** che interseca le rette  $\mathbf{r}_1(t) = (1, t, 2t), \mathbf{r}_2(t) = (1, 2t, t)$  e passa per il punto  $\mathbf{p} = (1, 1, 1)$ .

SUGGERIMENTO. Il modo più rapido è di trovare l'equazione del piano che passa per la retta  $\mathbf{r}_1$  ed il punto  $\mathbf{p}$ , e da essa ricavare il punto  $\mathbf{q}$  di intersezione di questo piano con la retta  $\mathbf{r}_2$ . Ora la retta  $\mathbf{r}$  desiderata è quella che passa per i punti  $\mathbf{p} \in \mathbf{q}$ .

Assai più pedissequamente si può scrivere l'equazione della retta che passa per il generico punto (1, t, 2t) di  $\mathbf{r}_1$  ed il generico punto = (1, 2s, s)di  $\mathbf{r}_2(t)$  (basta usare il vettore direzionale  $\mathbf{d}_t s = (1, t, 2t) - (1, 2s, s)$ ), ed imporre che passi anche per  $\mathbf{p} = ((1, 1, 1).$ 

ESERCIZIO 2.5.8. Trovare l'equazione della retta **r** che interseca le rette  $\mathbf{r}_1(t) = (1, t, 2t), \ \mathbf{r}_2(t) = (1, 2t, t)$  ed è parallela alla retta  $\mathbf{r}_3(t) = (t, -t, 0).$ 

SUGGERIMENTO. Il fatto che  $\mathbf{r}$  e  $\mathbf{r}_3$  siano parallele equivale a dire che hanno lo stesso vettore direzionale, cioè  $\mathbf{d} = (1, -1, 0)$  (o equivalentemente un suo multiplo). Scegliamo un punto  $\mathbf{p} = \mathbf{r}_1(t_0)$  sulla retta  $\mathbf{r}_1$ , e consideriamo la retta  $\mathbf{v}$  di equazione parametrica  $\mathbf{v}(s) = \mathbf{p} + s\mathbf{d}$ . Per tutti i valori di  $t_0$  per cui questa retta  $\mathbf{s}$  interseca  $\mathbf{r}_2$ , essa dà una soluzione del problema proposto. Per trovare quando questo accade si procede in maniera analoga al precedente Esercizio 2.5.7: si determina l'equazione del piano che contiene  $\mathbf{r}_1$  e  $\mathbf{s}$  e si vede se questo piano ha intersezione con  $\mathbf{r}_2$ .

ESERCIZIO 2.5.9. Trovare l'equazione della retta uscente dall'origine che interseca la retta  $\mathbf{r}(t) = (1, t, 2t)$  ed è parallela al piano 2x + z = 0.

SUGGERIMENTO. La condizione di parallelismo al piano equivale a richiedere che il vettore direzionale **d** della retta cercata sia ortogonale al vettore normale del piano  $\mathbf{n} = (2, 0, 1)$ . Quindi abbiamo un sottospazio bidimensionale di tali vettori (l'ortogonale di  $\mathbf{n}$ ), che corrispondono ad un piano di rette uscenti dall'origine: la retta desiderata è quella fra queste che interseca  $\mathbf{r}$ , cioè quella il cui vettore **d** è diretto verso il punto di intersezione di questo piano con  $\mathbf{r}_2$  (cioè è un multiplo di tale punto).

ESERCIZIO 2.5.10. Trovare l'equazione della retta passante per (1, 3, 2)e perpendicolare al piano 2x - y + z - 2 = 0.

SVOLGIMENTO. Il vettore direzionale della retta deve essere ortogonale al piano, quindi multiplo del suo vettore normale (2, -1, 1). Pertanto la retta ha equazione parametrica  $\mathbf{r}(t) = (1, 3, 2) + t(2, -1, 1) = (1 + 2t, 3 - t, 2 + t)$ .

ESERCIZIO 2.5.11. Trovare l'equazione della retta passante per (1, 3, 2) e perpendicolare al piano 2x - y + z - 2 = 0.

SUGGERIMENTO. Ci si riporta al precedente Esercizio 2.5.10 osservando che laretta cercata è quella passante per (1, 3, 2) e parallela alla retta di intersezione dei due piani (Esempio 2.2.1).

ESERCIZIO 2.5.12. Trovare l'equazione della retta uscente dall'origine che interseca perpendicolarmente la retta  $\mathbf{r}(t) = (1 + t, 2t, -t)$ .

SUGGERIMENTO. La retta cercata ha vettore direzionale perpendicolare a quello di  $\mathbf{r}$ , cioè a (1, 2, -1). Questi vettori direzionali giacciono quindi nel piano ortogonale a (1, 2, -1). Fra questi, quello dsiderato è quello che punta verso l'intersezione della retta  $\mathbf{r}$  con questo piano, esattamente come nell'Esercizio 2.5.10.

ESERCIZIO 2.5.13. Trovare la distanza minima fra le rette  $\mathbf{r}(t) = (0, 0, t)$ e  $\mathbf{q}(u) = (u, u + 1, u + 2)$ .

SVOLGIMENTO. Il segmento che connette un punto generico della prima retta ad un punto della seconda ha lunghezza minima se e solo se è perpendicolare ad entrambe le rette. Allora prendiamo in esame la prima retta e consideriamo il piano ad essa ortogonale che passa per il suo punto generico  $\mathbf{r}(t)$ . Questo piano è ortogonale al suo vettore direzionale (0, 0, 1), e quindi la sua equazione (2.3.2) è  $x = (0, 0, 1) \cdot \mathbf{r}(t)$ , cioè z = t. Intersechiamo tale piano con la retta  $\mathbf{q}$  e troviamo il punto di intersezione  $\mathbf{q}(u) = (u, u + 1, u + 2)$  con u + 2 = t, cioè il punto  $\mathbf{q}(t-2) = (t-2, t-1, t)$ . Il segmento che unisce  $\mathbf{r}(t)$  con  $\mathbf{q}(t-2)$  è diretto come  $\mathbf{s}(t) := \mathbf{q}(t-2) - \mathbf{r}(t) = (t-2, t-1, 0)$ . Ora basta imporre che questo vettore sia ortogonale a  $\mathbf{q}$ , cioè al suo vettore direzionale  $\mathbf{d} = (1, 1, 1)$ . Ma  $\mathbf{s}(t) \cdot \mathbf{d} = 2t - 3$ , da cui si ricava  $t = \frac{3}{2}$ . La distanza minima richiesta quindi è  $\|\mathbf{s}(\frac{3}{2})\| = \frac{1}{\sqrt{2}}$ .

ESERCIZIO 2.5.14. Trovare l'equazione della sfera con centro in  $\mathbf{c} = (1, 2, 3)$  e tangente al piano x + y + 2z - 1 = 0.

SUGGERIMENTO. Il raggio dal centro della sfera al punto di tangenza deve essere ortogonale al piano, quindi diretto come il suo vettore normale  $\mathbf{n} = (1, 1, 2)$ . Perciò questo raggio giace lungo la retta  $\mathbf{r}(t) = \mathbf{c} + t\mathbf{n}$ . Intersecando la retta  $\mathbf{r}$  con il piano tangente troviamo ora il punto di tangenza  $\mathbf{t}$ , e la norma  $\|\mathbf{t} - \mathbf{c}\|$ , ossia la sua distanza dal centro  $\mathbf{c}$ , è il raggio della sfera.

ESERCIZIO 2.5.15. Trovare l'equazione della sfera tangente nell'origine al piano  $\sigma$  di equazione x + y + z = 0 e tangente anche al piano  $\tau$  di equazione x + y + 2z - 1 = 0.

SUGGERIMENTO. Il centro della sfera deve giacere sulla retta  $\mathbf{r}$  uscente dall'origine e perpendicolare a  $\sigma$ , cioè diretta come il suo vettore normale  $\mathbf{p} = (1, 1, 1)$ . Normalizziamo  $\mathbf{p}$  ed otteniamo il versore normale  $\mathbf{n} = \left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right)$ . Quindi abbiamo l'equazione parametrica  $\mathbf{r}(t) = t\mathbf{n}$ . Poiché  $\mathbf{n}$  ha norma 1, la distanza di  $\mathbf{r}(t)$  dall'origine è |t|. calcoliamo allora, come nell'Esempio 2.3.1, la proiezione ortogonale  $\mathbf{p}(t)$  del punto  $\mathbf{r}(t)$  sul piano  $\tau$ . La sfera è tangente a  $\tau$  se e solo se la distanza dal suo centro al punto di tangenza su  $\tau$  è la stessa di quella dal centro al punto di tangenza su  $\sigma$  (ossia l'origine). Pertanto, per trovare il centro della sfera, basta imporre che il centro sia il punto  $\mathbf{r}(t)$  tale che  $\|\mathbf{r}(t) - \mathbf{p}(t)\| = \|\mathbf{r}(t)\| = |t|$ . Dopo di che, il raggio della sfera è  $\|\mathbf{r}(t)\| = |t|$ .

#### 2.6. Raggi riflessi e rifratti

Varie procedure in Computer Graphics richiedono di tracciare raggi (cioè rette parametriche) nello spazio e farli riflettere in maniera speculare su superficie. Il problema, espresso in termini del vettore direzionale  $\mathbf{p}$  del raggio, si riduce al seguente. Dato un versore  $\mathbf{p}$  ed un piano  $\pi$  che passa per l'origine con versore normale  $\mathbf{n}$ , come si determina il versore  $\mathbf{v}$  riflesso speculare di  $\mathbf{p}$  rispetto a  $\mathbf{n}$ ? Per ragioni di semplicità di programmazione e velocità di esecuzione del codice, vorremmo trovare la risposta solo in termini di prodotti scalari (somme di prodotti di coordinate, rapidi da calcolare) invece che tramite espressioni trigonometriche.

La proiezione del versore  $\mathbf{p}$  sul versore normale  $\mathbf{n}$  vale  $\cos \theta \mathbf{n}$ , dove  $\theta$  è l'angolo di incidenza del raggio sul piano  $\pi$ . Perciò la componente di  $\mathbf{p}$  ortogonale a  $\mathbf{n}$  vale  $\mathbf{s} = \mathbf{p} - \cos \theta \mathbf{n}$ . Si osservi che questo vettore  $\mathbf{s}$  è quello che verifica l'identità  $\mathbf{p} + \mathbf{s} = \cos \theta \mathbf{n} = (\mathbf{p} \cdot \mathbf{n})mbn$ , cioè porta da  $\mathbf{p}$  al piede della sua proiezione lungo  $\mathbf{n}$ .



FIGURA 1. Versore riflesso

È quindi chiaro che si ha

 $\mathbf{v} = \mathbf{p} + 2\mathbf{s} = 2\cos\theta\mathbf{n} - \mathbf{p} = 2(\mathbf{p}\cdot\mathbf{n})\mathbf{n} - \mathbf{p}$ 

Talvolta i raggi di luce, anziché riflettersi sulla superficie di un materiale, vi si rifrangono dentro. Determiniamo l'espressione del versore rifratto dalla legge di Snell della rifrazione, che asserisce che il raggio incidente viene deviato in maniera da soddisfare la seguente relazione fra l'angolo  $\theta_i$  che esso forma con la normale alla superficie (nel nostro caso un piano  $\pi$  e l'analogo angolo  $\theta_t$  formato dal raggio rifratto, cioè trasmesso al secondo materiale:

$$\eta_i \sin \theta_i = \eta_t \sin \theta_t$$

dove  $\eta_i \in \eta_i$  sono i rispettivi indici di rifrazione dei due materiali, definiti come il rapporto tra la velocità della luce nel vuoto e nel materiale: essi variano con la lunghezza d'onda della luce. Il vuoto ha quindi indice di rifrazione 1; i materiali hanno indice di rifrazione più elevato. Calcoliamo il versore direzionale **t** del raggio rifratto. Per prima cosa, determiniamo il versore **m** perpendicolare al versore normale **n** che giace nel piano generato da **n** e dal versore della direzione di incidenza **i**, e punta verso il lato opposto di **i** rispetto a **n**. La proiezione di **i** lungo **n** vale  $\cos \theta_i$  **n**. Perciò il vettore  $\cos \theta_i$  **n** - **i** è diretto parallelamente al piano  $\pi$ , e quindi è perpendicolare a **n**. Poiché **n** e **i** hanno norma 1 (sono versori!) e  $\cos \theta_i$  **n** è ortogonale a **n** e quindi anche a  $\cos \theta_i$  **n**, per il teorema di Pitagora la norma di  $\cos \theta_i$  **n** - **i** vale  $\sqrt{1 - \cos^2 \theta_i} = \sin \theta_i$ (il seno è positivo perché l'angolo  $\theta_i$  è compreso fra 0 e  $\pi/2$ . Quindi, normalizzando, troviamo un vettore **m** =  $(\cos \theta_i$  **n** - **i**)/sin \theta\_i.



FIGURA 2. Offset trasversale del raggio incidente

Ora, per la legge di Snell, il versore del raggio trasmesso è

$$\mathbf{t} = \sin \theta_t \mathbf{m} - \cos \theta_t \mathbf{n} = \frac{\sin \theta_t}{\sin \theta_i} (\cos \theta_i \mathbf{n} - \mathbf{i}) - \cos \theta_t \mathbf{n}.$$

D'altra parte, sempre per la legge di Snell, si ha sin $\theta_t/\sin\theta_i=\eta_i/\eta_t$ . Quindi, ponendo $\eta_\tau=\eta_i/\eta_t$ , otteniamo

$$\mathbf{t} = (\eta_\tau \cos \theta_i - \cos \theta_t) \mathbf{n} - \eta_\tau \mathbf{i}.$$



FIGURA 3. Il versore rifratto

D'altra parte  $\cos \theta_i = \mathbf{n} \cdot \mathbf{i}$ , e

$$\cos \theta_t = \sqrt{1 - \sin^2 \theta_t} = \sqrt{1 - \eta_\tau^2 \sin^2 \theta_i}$$
$$= \sqrt{1 - \eta_\tau^2 (1 - \cos^2 \theta_i)} = \sqrt{1 - \eta_\tau^2 (1 - (\mathbf{n} \cdot \mathbf{i})^2)}.$$

Abbiamo così ricavato  ${\bf t}$  unicamente in termini di prodotti scalari:

$$\mathbf{t} = \left(\eta_{\tau} \mathbf{n} \cdot \mathbf{i} - \sqrt{1 - \eta_{\tau}^2 (1 - (\mathbf{n} \cdot \mathbf{i})^2)}\right) \mathbf{n} - \eta_{\tau} \mathbf{i}$$

106
## CAPITOLO 3

## Spazi proiettivi

## 3.1. Lo spazio proiettivo

DEFINIZIONE 3.1.1. (Quoziente rispetto ad una relazione di equivalenza.) Sia X uno spazio vettoriale (a dimensione finita) e ~ una relazione di equivalenza su X: la classe di equivalenza con rappresentante  $\mathbf{x} \in \mathbf{X} \ e \ \mathbf{y} \sim \mathbf{x}$ . L'insieme delle classi lateraloi di X si chiama lo spazio quoziente  $\mathbf{X}/\sim$ .

Ad esempio, l'equivalenza per dilatazione identifica due vettori non nulli in  $\mathbb{R}^n$  se sono multipli uno dell'altro. Le classi di equivalenza sono le rette passanti per l'origine tolta l'origine stessa, che possiamo pensare parametrizzati dai versori modulo il segno, cioè dai punti della sfera unitaria con punti antipodali identificati (le direzioni dall'origine verso l'infinito). Se si preferisce separare i segni, cioè avere una relazione di equivalenza le cui classi siano parametrizzate dai punti della sfera, si devono considerare equivalenti solo vettori multipli l'uno dell'altro per scalari positivi.

Sommando i vettori di una retta con quelli di un'altra otteniamo una figura piana, non una retta, quindi la relazione di equivalenza non rispetta la somma; ovviamente non rispetta la moltiplicazione per scalari, perché moltiplicando per 0 usciamo dalle classi di equivalenza. Però rispetta la moltiplicazione per scalari non nulli.

NOTA 3.1.2. (**Proiezione stereografica.**) Una parte delle classi di equivalenza della relazione di equivalenza per dilatazione (cioè le rette uscenti dall'origine, scavate dell'origine stessa) sono in corrispondenza biunivoca con i punti di una sfera tangente all'origine in  $\mathbb{R}^n$ , detta sfera stereografica. Infatti, se indichiamo con  $\sigma$  l'iperpiano (cioè il sottospazio a dimensione n-1) di tangenza, le classi di equivalenza date dalle rette che non giacciono su  $\sigma$  sono in corrispondenza biunivoca con i punti della sfera eccetto il punto di tangenza, ed anche con i punti dell'altro iperpiano  $\tau$  tangente a questa sfera e parallelo a  $\sigma$ . Ad esempio, se la sfera ha diametro di lunghezza 1 e se per  $\sigma$  scegliamo l'iperpiano { $\mathbf{x} :$  $x_n = 0$ }, si ha  $\tau = {\mathbf{x} : x_n = 1}$ . In tal modo si crea una corrispondenza fra i punti di quest'ultimo iperpiano (che è una copia di  $\mathbb{R}^{n-1}$ ) e quelli

### 3. SPAZI PROIETTIVI

della sfera stereografica tolta l'origine: questa corrispondenza si chiama proiezione stereografica. Si noti che nella proiezione stereografica punti della sfera che si muovono verso l'origine sono associati a punti di  $\tau$  che si muovono verso l'infinito di  $\mathbb{R}^{n-1}$ . Il caso a dimensione due è illustrato nella Figura 1.



FIGURA 1. Proiezione stereografica in due dimensioni

Si osservi che, in dimensione superiore a 2, le rette *orizzontali*, cioè che giacciono in  $\sigma$ , costituiscono (tolta l'origine) differenti classi di equivalenza per dilatazione, che identificano differenti direzioni versi l'infinito, e si parametrizzano ovviamente con una sfera a dimensione n-1 (nel caso di  $\mathbb{R}^3$ , con un cerchio, il cerchio di raggio, ad esempio, 1 nel piano di base  $\sigma = \{\mathbf{x} : x_n = 0\}$ . Invece, sulla sfera stereografica, cè un unico punto sul piano di base  $\sigma$ , e quindi la proiezione stereografica non separa le direzioni verso l'infinito sul piano di tangenza.

A differenza dell'esempio precedente, siamo interessati ad equivalenze che separino le direzioni all'infinito. Chiariamo meglio questo concetto, prendendo ad esempio il piano. Consideriamo le rette nel piano, e dichiariamo due di esse equivalenti se sono parallele: possiamo pensare che le classi di equivalenza individuino le *direzioni all'infinito*. Questo significa pensare la direzione del vettore  $\mathbf{x} = (x_1, x_2)$  come la pendenza della retta formata da tutti i multipli di  $\mathbf{x}$ . Questa pendenza è data dal rapporto  $\frac{x_1}{x_2}$  (che può essere infinito, il che accade quando la retta è l'asse  $x_1 = 0$ ).

DEFINIZIONE 3.1.3. *(Spazio proiettivo.)* Vogliamo aggiungere ai punti di  $\mathbb{R}^2$  un ulteriore insieme di *punti all'infinito*, o forse dovremmo dire

108

direzioni all'infinito, parametrizzato dai punti di un cerchio, come spiegato nella Nota 3.1.2. A questo scopo prendiamo in esame una variante dell'ambiente geometrico tridimensionale introdotto in quella Nota per la proiezione stereografica, con il piano di base  $\sigma = \{\mathbf{x} : x_0 = 0\}$ ed il suo traslato  $\tau = \{\mathbf{x} : x_0 = 1\}$ . Introduciamo su  $\mathbb{R}^3 \setminus \{\mathbf{0}\}$  una relazione di equivalenza che identifichi i punti di  $\tau$  con i punti di uno spazio vettoriale bidimensionale  $\mathbb{R}^2$ , ed i punti del cerchio unitario (o di un qualsiasi raggio non nullo) con centro l'origine in  $\sigma$  con i punti all'infinito di  $\mathbb{R}^2$ .

Questa relazione di equivalenza è nient'altro che l'equivalenza per dilatazioni non nulle già introdotta all'inizio di questa Sezione: due punti in  $\mathbb{R}^3$  diversi dall'origine, diciamo  $(x_0, x_1, x_2)$  e  $(x'_0, x'_1, x'_2)$  sono equivalenti se esiste uno scalare  $\lambda \neq 0$  tale che

$$(x'_0, x'_1, x'_2) = (\lambda x_0, \lambda x_1, \lambda x_2)$$
.

In particolare, se  $x_0 \neq 0$ , cioè se il vettore non si trova nel piano di base  $\sigma$ , allora ogni classe di equivalenza ha un rappresentante nel piano  $\tau$ , ed uno solo, dato dal punto  $(1, \frac{x_1}{x_0}, \frac{x_2}{x_0})$ ; invece per la classe di equivalenza di un punto del piano di base, cioè del tipo  $\mathbf{v} = (0, x_1, x_2)$ , si può scegliere il rappresentante dato dal punto del cerchio unitario in  $\sigma$  con la stessa direzione di  $\mathbf{v}$ , cioè dal versore  $\mathbf{v}/||\mathbf{v}||$ .

Lo spazio quoziente di  $\mathbb{R}^3 \setminus \{0\}$  rispetto a questa relazione di equivalenza, cioè l'insieme di tutte le classi di equivalenza si indica con  $\mathbb{P}^2$ . Come già osservato, non costituisce uno spazio vettoriale, ed infatti l'operazione di somma non passa al quoziente, perché dipende dalla scelta dei rappresentanti, e quindi non è ben definita sulle classi di equivalenza.

Più precisamente:

ESEMPIO 3.1.4. La classe di equivalenza per dilatazione del vettore  $s(x_0, x_1, x_2) + t(x'_0, x'_1, x'_2)$  dipende dalla scelta degli scalari *s* e *t*, a meno che  $(x_0, x_1, x_2)$  e  $(x'_0, x'_1, x'_2)$  siano multipli non nulli l'uno dell'altro. Infatti, basta osservare che per s = 0 la classe di equivalenza è quella di  $(x'_0, x'_1, x'_2)$ , mentre per t = 0 è quella di  $(x_0, x_1, x_2)$ .

In tal modo lo spazio proiettivo  $\mathbb{P}^2$  parametrizza i punti al finito ed all'infinito grazie a tre coordinate  $(x_0, x_1, x_2)$ , dette coordinate omogenee perché in esse le equazioni delle rette, che in  $\mathbb{R}^2$  sono del tipo  $a_1x_1 + a_2x_2 + a_0 = 0$ , diventano (grazie all'equivalenza per dilatazioni non nulle) equazioni omogenee, del tipo  $a_0x_0 + a_1x_1 + a_2x_2 = 0$ . I punti all'infinito, come abbiamo visto, sono geometricamente in corrispondenza biunivoca con quelli del cerchio unitario nel piano  $x_0 = 0$ , ma

### 3. SPAZI PROIETTIVI

(sempre grazie all'equivalenza per dilatazioni non nulle) sono anche in corrispondenza con i punti  $(x_0, x_1, x_2)$  che verificano l'equazione  $x_0 = 0$ , la quale è l'equazione di un piano in tre dimensioni (il piano  $x_0 = 0$ , appunto), ovvero di una retta in due dimensioni.

Analogamente si tratta il caso della geometria proiettiva in n + 1dimensioni reali ( $\mathbb{P}^n(\mathbb{R})$ ), o anche, più in generale, complesse ( $\mathbb{P}^n(\mathbb{C})$ ).

Per comodità nell'uso degli indici, d'ora in avanti permuteremo ciclicamente le coordinate ed indicheremo i punti al finito con  $(x_1, x_2, \ldots, x_{n-1}, 1)$ e quelli all'infinito con  $(x_1, x_2, \ldots, x_{n-1}, 0)$ . Quindi l'iperpiano dei punti al finito in  $\mathbb{P}^{n-1}$  ora ha equazione  $x_n = 1$  e quello dei punti all'infinito  $x_n = 0$ .

## 3.2. Trasformazioni proiettive

Consideriamo il gruppo lineare  $GL_n(\mathbb{R})$  delle matrici reali invertibili a dimensione n. Ora vogliamo prendere in esame la forma matriciale degli operatori lineari sullo spazio proiettivo  $\mathbb{P}^{n-1}$ . Per questo scopo fissiamo una scelta privilegiata di rappresentanti nelle classi di equivalenza, quella legata alla rappresentazione stereografica introdotta nella precedente Sezione 3.1: per i rappresentanti delle classi al finito scegliamo  $(x_1, x_2, \ldots, x_{n-1}, 1)$ , mentre per le classi all'infinito  $(x_1, x_2, \ldots, x_{n-1}, 0)$ .

Come visto nella Sezione 3.1, per le classi in  $\mathbb{P}^{n-1}$  usiamo n coordinate *omogenee*, definite univocamente solo a meno di dilatazioni, rammentandoci però del fatto che in  $\mathbb{P}^{n-1}$  bastano n-1 parametri per identificare le classi. Infatti l'insieme delle classi al finito è isomorfo a  $\mathbb{R}^{n-1}$ , e ad esse si devono aggiungere quelle all'infinito, che formano una circonferenza in  $\mathbb{R}^{n-1}$ , o se si preferisce un iperpiano: l'iperpiano  $x_0 = 0$ . Questo iperpiano, si parametrizza con soli n-2 parametri (ad esempio n-2 variabili cartesiane  $x_1, \ldots, x_{n-2}$ , oppure, se lo si pensa come una circonferenza, n-2 angoli di Eulero). Per evitare ambiguità notazionali, in questa Sezione denotiamo in termini di coordinate il vettore  $\mathbf{x} \in \mathbb{R}^n$  con

$$\left(\begin{array}{c} x_1\\ \vdots\\ x_n \end{array}\right)$$

e la sua classe di equivalenza per dilatazione, cioè l'elemento di  $\mathbb{P}^{n-1}$ di cui **x** è rappresentante, con

$$\left[\begin{array}{c} x_1\\ \vdots\\ x_n \end{array}\right]$$

Le matrici in  $GL_n(\mathbb{R})$  agiscono sui vettori, invece che sulle classi di equivalenza. Però, per linearità, se due vettori  $\mathbf{x}$  e  $\mathbf{y}$  sono multipli uno dell'altro,  $\mathbf{x} = \lambda \mathbf{y}$ , ogni matrice M in  $GL_n(\mathbb{R})$  verifica  $M\mathbf{x} = \lambda M\mathbf{y}$ . Quindi M preserva le classi di equivalenza, e perciò l'azione di  $GL_n(\mathbb{R})$ su  $\mathbb{R}^n$  induce una azione ben definita di  $GL_n(\mathbb{R})$  sullo spazio proiettivo  $\mathbb{P}^{n-1}$ . Però l'azione di matrici diverse può essere identica: per via del fatto che la scelta del rappresentante di una classe è determinata solo a meno di multipli, per ogni  $\lambda \neq 0$  reale l'azione di M coincide con quella di  $\lambda M$ . Pertanto, viste come operatori sullo spazio proiettivo, Me  $\lambda M$  si identificano. Questo porta a definire correttamente il gruppo di operatori lineari che agiscono sullo spazio proiettivo in termini di equivalenza sotto dilatazione, come segue.

DEFINIZIONE 3.2.1. (Gruppo lineare proiettivo.) Il gruppo lineare proiettivo  $PGL_n(\mathbb{R})$  è il gruppo i cui elementi sono le classi di equivalenza di  $GL_n(\mathbb{R})$  sotto la relazione di equivalenza che identifica due matrici  $M_1$  e  $M_2$  se esiste  $\lambda \in \mathbb{R}, \lambda \neq 0$  tale che  $M_1 = \lambda M_2$ .

Le operazioni di prodotto e di inverso in  $GL_n(\mathbb{R})$  (prodotto righe per colonne) rispettano le classi di equivalenza, perché il prodotto di multipli di due matrici è un multiplo della matrice prodotto, e quindi le operazioni di gruppo su  $GL_n(\mathbb{R})$  inducono operazioni analoghe ben definite su  $PGL_n(\mathbb{R})$ .

DEFINIZIONE 3.2.2. (Quoziente sotto dilatazione dello spazio delle matrici reali.) Analogamente alla Definizione 3.2.1, introduciamo lo spazio vettoriale  $M_n(\mathbb{R})$  delle matrici  $n \times n$  a coefficienti reali, ewd il suo quoziente  $PM_n(\mathbb{R})$  modulo l'equivalenza per dilatazione positiva, cioè modulo il sottoinsieme { $\lambda I$ }, con  $\lambda > 0$  e I la matrice identità.

NOTAZIONE 3.2.3. In analogia a quanto fatto per  $\mathbb{P}^n$ , indichiamo nel modo seguente gli elementi di  $PGL_n(\mathbb{R})$  e  $PM_n(\mathbb{R})$  a partire dalle matrici in  $GL_n(\mathbb{R})$  e  $PM_n(\mathbb{R})$  che li rappresentano: l'elemento di  $PGL_n(\mathbb{R})$ o  $PM_n(\mathbb{R})$  associato alla matrice

$$\left(\begin{array}{ccc} m_{11} & \cdots & m_{1n} \\ & \ddots & \\ m_{n1} & \cdots & m_{nn} \end{array}\right)$$

si scrive con le parentesi quadre:

$$\left(\begin{array}{ccc}m_{11}&\cdots&m_{1n}\\&\ddots\\&\\m_{n1}&\cdots&m_{nn}\end{array}\right)$$

### 3. SPAZI PROIETTIVI

ESEMPIO 3.2.4. Per semplificare la notazione, restringiamo in questo esempio la nostra attenzione al caso dello spazio proiettivo tridimensionale  $\mathbb{P}^4$ , con coordinate omogenee x, y, z, w (questo è il caso di interesse nelle applicazioni della geometria proiettiva alla Computer Graphics, che vedremo in seguito nella Sezione 4.2. Consideriamo una classe con un rappresentante al finito, cioè del tipo (x, y, z, 1). Sia  $M \in PM_4(\mathbb{R})$ , ed indichiamo con lettere maiuscole le coordinate dei rappresentanti della classe in cui M manda (x, y, z, 1). La notazione quindi diventa:

$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} = \begin{pmatrix} m_{11} & \cdots & m_{1n} \\ & \ddots & \\ m_{n1} & \cdots & m_{nn} \end{pmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} .$$
(3.2.1)

Si osservi che  $M \in PM_4(\mathbb{R})$  può mandare una classe al finito, come [x, y, z, 1] sia in un'altra classe al finito, sia in una classe all'infinito, del tipo [x, y, z, 0]. Ad esempio, la matrice identità manda ciascuna classe in sé stessa, mentre la (classe di equivalenza dell) matrice di proiezione sui punti all'infinito,

$$\left(\begin{array}{rrrrr} 1 & 0 & 0 & 0 \\ & \ddots & & \\ 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right)$$

manda [x, y, z, 1] in [x, y, z, 0], ed infine, se  $\lambda \neq 0$ ,

manda [x, y, z, 0] in  $[x, y, z, \lambda z]$ , che, almeno per  $z \neq 0$ , è una classe al finito.

## CAPITOLO 4

## Trasformazioni affini

## 4.1. Moti rigidi in $\mathbb{R}^n$ immersi in trasformazioni lineari di $\mathbb{R}^{n+1}$

L'insieme delle trasformazioni lineari di  $\mathbb{R}$  in sé è l'insieme dei funzionali lineari su  $\mathbb{R}$ , e quindi, per il Teorema di rappresentazione di Riesz, è uno spazio vettoriale isomorfo a  $\mathbb{R}$ : ogni  $a \in \mathbb{R}$  corrisponde alla trasformazione lineare su  $\mathbb{R}$  data dalla dilatazione  $x \mapsto ax$  (una costante a è infatti la più generale matrice  $1 \times 1$ ).

Consideriamo la famiglia (anzi il gruppo) data da tutte le composizioni di queste trasformazioni lineari (le dilatazioni) e le traslazioni di  $\mathbb{R}$ , cioè le applicazioni (non lineari perché non preservano l'origine!) di  $\mathbb{R}$  in sé definite da  $x \mapsto x + b$ , dove  $b \in \mathbb{R}$ . È facile costruire esempi che mostrano che questo è un gruppo non commutativo. Esso non consiste quindi di matrici  $1 \times 1$ : però ora mostriamo che i suoi elementi possono essere rappresentati come matrici  $2 \times 2$ , cioè di applicazioni lineari su  $\mathbb{R}^2$ .

Infatti:

LEMMA 4.1.1. Immergiamo la retta reale come una retta nello spazio bidimensionale  $\mathbb{R}^2$ , ad esempio come la retta  $\{x_2 = 1\}$ , parallela all'asse delle ascisse ad ordinata 1. Allora esiste una applicazione lineare di  $\mathbb{R}^2$ in sé che preserva questa retta, e su di essa agisce traslandone l'origine (cioè il punto (0, 1) e dilatando gli intervalli di un fattore a.

DIMOSTRAZIONE. Per prima cosa troviamo quali operatori lineari su  $\mathbb{R}^2$  preservano la retta  $\{x_2 = 1\}$ . Scriviamo questa condizione come una condizione sulla matrice A di un tale operatore nella base canonica. La condizione è che per ogni  $s \in \mathbb{R}$  esista  $t \in \mathbb{R}$  tale che

$$A\left(\begin{array}{c}s\\1\end{array}\right) = \left(\begin{array}{c}t\\1\end{array}\right)$$

In particolare, se si sceglie s = 0, si vede che A manda il secondo vettore della base canonica in un vettore con ordinata 1, e quindi la seconda colonna di A è del tipo (b, 1) per qualche  $b \in \mathbb{R}$ . D'altra parte, la differenza fra due vettori sulla retta  $\{x_2 = 1\}$  è un vettore proporzionale a  $\mathbf{e}_1 = (1,0)$ , e quindi questo vettore viene mandato in un multiplo (a,0) di  $\mathbf{e}_1$ . Pertanto la prima colonna della matrice A è del tipo (a,0) per qualche  $a \in \mathbb{R}$ . Riassumendo si ha:

$$A = \left(\begin{array}{cc} a & b \\ 0 & 1 \end{array}\right) \ .$$

Osserviamo che questa matrice manda (s, 1) in (as+b, 1), e quindi trasla l'origine della retta  $\{x_2 = 1\}$  di una quantità b, e dilata gli intervalli di questa retta di un fattore a.

In tre dimensioni il problema corrispondente è quello di quali operatori lineari agiscono preservando il piano  $x_3 = 1$  e traslandone l'origine di un vettore *orizzontale*  $(v_1, v_2, 0)$ . Esattamente lo stesso argomento di prima porta ora alla conclusione seguente.

PROPOSIZIONE 4.1.2. Le trasformazioni lineari di  $\mathbb{R}^3$  in  $\mathbb{R}^3$  che lasciano invariante il piano  $\{x_3 = 1\}$  e traslano il punto (0, 0, 1) (l'origine del piano affine) nel punto  $(v_1, v_2, 1)$  sono precisamente quelle la cui espressione matriciale (nella base canonica) è triangolare a blocchi, del tipo

$$A = \left(\begin{array}{cc} a & b & v_1 \\ c & d & v_2 \\ 0 & 0 & 1 \end{array}\right)$$

con a, b, c,  $d \in \mathbb{R}$ .

Ora riscriviamo queste considerazioni per  $\mathbb{R}^n$ .

DEFINIZIONE 4.1.3. (*Trasformazioni affini.*) Una trasformazione di  $\mathbb{R}^n$  in sé che si ottiene componendo una traslazione con una trasformazione lineare si chiama una *trasformazione affine*.

PROPOSIZIONE 4.1.4. Se identifichiamo  $\mathbb{R}^n$  con l'iperpiano di equazione  $x_{n+1} = 1$  in  $\mathbb{R}^{n+1}$ , le trasformazioni affini di  $\mathbb{R}^n$  corrispondono alle trasformazioni lineari di  $\mathbb{R}^{n+1}$  la cui matrice (nella base canonica) ha forma triangolare a blocchi del tipo

$$A_{n+1} = \begin{pmatrix} a_{11} & \dots & a_{1n} & v_1 \\ & \ddots & & \\ a_{n1} & \dots & a_{nn} & v_n \\ 0 & \dots & 0 & 1 \end{pmatrix}$$

dove il vettore  $(v_1, v_2, \ldots, v_n)$  è il punto in cui viene traslata l'origine di  $\mathbb{R}^n$ .

Una trasformazione affine T è identificata da dove manda l'origine e dove manda i vettori di una base. Supponiamo che mandi l'origine in in  $\mathbf{v}$  ed il vettore  $\mathbf{e}_j$  della base canonica in  $\mathbf{u}_j$ , per  $j = 1, \ldots, n$ . Allora T è identificata anche dal vettore  $\mathbf{v}$  immagine dell'origine e dai vettori applicati a  $\mathbf{v}$  dati da  $\mathbf{v}_j = \mathbf{e}_j - \mathbf{v}$ . Più precisamente:

DEFINIZIONE 4.1.5. (Spazio affine, vettori applicati e trasformazioni affini.) Lo spazio affine reale X a dimensione n è lo spazio  $\mathbb{R}^n$  con una scelta di origine o da cui calcolare le coordinate, cioè è la coppia  $\mathbf{o} \times \mathbb{R}^n \subset \mathbb{R}^{2n}$ ; il vettore  $\mathbf{o} \in \mathbb{R}^n$  si chiama il punto di applicazione dei vettori. Il generico elemento  $(\mathbf{o}, \mathbf{v})$  di X si chiama vettore applicato al punto o. Per evitare ambiguità fra vettori e vettori applicati, denotiaomo il vettore applicato  $(\mathbf{o}, \mathbf{v})$  con  $\overrightarrow{\mathbf{ov}}$ .

COROLLARIO 4.1.6. Una trasformazione affine T su X sposta l'origine in una nuova origine  $\mathbf{o}'$ , ed i vettori applicati a  $\mathbf{o}$  in vettori applicati a  $\mathbf{o}'$ ; essa è univocamente determinata dalla nuova origine  $\mathbf{o}'$  (che esprime la traslazione che è stata applicata all'origine iniziale  $\mathbf{o}$ ) e dalle immagini dei vettori di una base applicati a  $\mathbf{o}$ , che sono vettori applicati a  $\mathbf{o}'$ .

NOTA 4.1.7. **Cautela:** poichè una trasformazione affine sposta l'origine, essa manda vettori applicati all'origine in vettori applicati altrove (lo spostamento essendo dato dalla componente di traslazione). Perciò non è più vero che la matrice che la rappresenta ha per colonne l'immagine dei vettori canonici di base applicati all'origine : bisogna traslarli perchè siano applicati alla nuova origine. La forma che la matrice assume viene presentata nel prossimo enunciato. □

COROLLARIO 4.1.8. (i) La traslazione di  $\mathbb{R}^n$  che manda l'origine in  $\mathbf{v}$  (e quindi che sposta gli assi coordinati in assi a questi paralleli, ossia manda i vettori  $\mathbf{e}_i$  della base canonica in  $\mathbf{e}_i + \mathbf{v}$ ) ha per matrice

$$A = \begin{pmatrix} 1 & 0 & \dots & 0 & v_1 \\ & \ddots & & \\ 0 & 0 & \dots & 1 & v_n \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix}$$

Osserviamo che, quando consideriamo lo spazio affine come immerso in  $\mathbb{R}^{n+1}$ , l'i-simo vettore della base canonica diventa  $\mathbf{e}_i = (0, \dots, 0, 1, 0, \dots, 0, 1)$  con il primo 1 alla componente i-sima, e la matrice A manda questo vettore in  $\mathbf{w}_i :== \mathbf{e}_i +$  $\mathbf{v} = (v_1, v_2, \dots, v_{i-1}, v_i + 1, v_{i+1}, \dots, v_n, 1)$ . La colonna i-sima della matrice è quindi il vettore  $\mathbf{e}_i = \mathbf{w}_i - \mathbf{v}$ . (ii) Più in generale, la trasformazione affine che manda l'origine in  $\mathbf{o}' \equiv \mathbf{v}$  ed il vettore  $\mathbf{e}_i$  della base canonica in  $\mathbf{w}_i$ , cioè il vettore applicato  $\overrightarrow{\mathbf{oe}_j}$  nel vettore applicato  $\overrightarrow{\mathbf{vw}_j} = \mathbf{w}_j - \mathbf{v}$  (per  $j=1,\ldots,n)$  è

$$A = \begin{pmatrix} w_{11} - v_1 & w_{21} - v_1 & \dots & w_{n1} - v_1 & v_1 \\ & & \ddots & & \\ w_{1n} - v_n & w_{2n} - v_n & \dots & w_{nn} - v_n & v_n \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix}$$

dove abbiamo posto  $w_{ij} = (\mathbf{w}_i)_j$ , la j-sima componente del vettore  $\mathbf{w}_i$ . Si noti che le colonne della sottomatrice  $n \times n$ in alto a sinistra di A sono date dalle coordinate dei vettori applicati  $\overrightarrow{\mathbf{vw}}_i$  immagini dei vettori canonici di base (che si possono pensare applicati all'origine)  $\mathbf{e}_{\mathbf{j}} = \overrightarrow{\mathbf{oe}_{\mathbf{j}}}$ . La colonna *i-sima della matrice è quindi il vettore*  $\mathbf{w}_i - \mathbf{v}$ .

DIMOSTRAZIONE. La cui verifica è immediata: basta applicare la trasformazione affine ai vettori canonici di base  $\mathbf{e}_i$ .

Ora abbiamo:

COROLLARIO 4.1.9. (Cambio di riferimento affine.) Sia  $\{\overrightarrow{\mathbf{oe}}_{i}, j =$  $1, \ldots, n$  una base di vettori applicati a **o**, e { $\overline{\mathbf{vw}_j}$ ,  $j = 1, \ldots, n$ } una base di vettori applicati a  $\mathbf{v} \equiv \mathbf{o}'$ . Indichiamo con  $x_i$  le componenti di un vettore  $\vec{\mathbf{ox}}$  applicato ad  $\mathbf{o}$  nella prima base, e con  $x'_i$  le componenti del vettore applicato a  $\mathbf{v}$  che si ottiene dallo stesso punto spaziale  $\mathbf{x}$ , cioè il vettore applicato  $\vec{\mathbf{vx}}$ : in altre parole, sia

 $\overrightarrow{\mathbf{ox}} = \sum_{i=1}^{n} x_i \overrightarrow{\mathbf{oe}_i}$ 

е

$$\overrightarrow{\mathbf{o'x}} = \sum_{i=1}^{n} x'_i \overrightarrow{\mathbf{o'w}_i}$$
  
Allora le nuove coordinate affini  $(x'_1, \ldots, x'_n)$  sono l'immagine delle  
coordinate  $(x_1, \ldots, x_n)$  sotto la matrice inversa di quella del Corol-  
lario 4.1.8, cioè si ha

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ 1 \end{pmatrix} = \begin{pmatrix} w_{11} - v_1 & w_{21} - v_1 & \dots & w_{n1} - v_1 & v_1 \\ & & \ddots & & \\ w_{1n} - v_n & w_{2n} - v_n & \dots & w_{nn} - v_n & v_n \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix} \begin{pmatrix} x'_1 \\ x'_2 \\ \vdots \\ x'_n \\ 1 \end{pmatrix} .$$

116

La matrice inversa che realizza la trasformazione di coordinate si chiama la matrice del cambiamento di riferimento affine.

NOTA 4.1.10. Poiché le trasformazioni lineari mandano parallelogrammi in parallelogrammi, e parallelepipedi in parallelepipedi, lo stesso è vero per ogni trasformazione affine, in quanto essa differisce da una trasformazione lineare per una traslazione. Ad esempio, una trasformazione del piano che manda il quadrato Q di vertici (0,0), (1,0), (0,1), (1,1) nel quadrilatero di vertici (1,3), (2,2), (-1,-1), (3,2) non è affine, perché il quadrilatero non è un parallelogramma.

Viceversa, dato un parallelogramma  $P_2$  nel piano o un parallelepipedo  $P_3$  in  $\mathbb{R}^3$ , esiste sempre una trasformazione lineare che manda il cubo unitario del rispettivo spazio (rispettivamente  $\mathbb{R}^2$  o  $\mathbb{R}^3$ ) in un traslato di  $P_2$  (rispettivamente,  $P_3$ ) che ha un vertice nell'origine (infatti i vertici del cubo contigui all'origine formano una base nel rispettivo spazio, per cui esiste una trasformazione lineare che li manda nei vertici contigui all'origine in  $P_2$  (rispettivamente  $P_3$ ), e la regola del parallelogramma asserisce che i vertici restanti vengono mandati da questa trasformazione nei corrispondenti altri vertici del parallelogramma o parallelepipedo.

Quindi esiste sempre una trasformazione affine che manda il cubo unitario in un qualsiasi parallelogramma (in  $\mathbb{R}^2$ ) o parallelepipedo (in  $\mathbb{R}^3$ ).

### 4.2. Esempio: spostamento di una macchina da ripresa

Per praticità, e per il suo significato geometrico, formuliamo questo esempio in tre dimensioni. Vogliamo trovare la forma matriciale della rototraslazione in tre dimensioni senza deflessione laterale del piano verticale (o *tilt*): questa trasformazione è quella che determina il cambiamento degli assi del sistema di riferimento quando un osservatore si sposta senza inclinare lateralmente il piano verticale della macchina da ripresa, e quindi è essenziale in Computer Graphics. In effetti, sarebbe naturale identificare l'asse laterale con l'asse x, l'asse di osservazione con l'asse y, e come terzo asse (di elevazione) l'asse z. In realtà, è tradizione in Computer Graphics scegliere come asse y quello verticale ed invece riservare l'asse z è per la profondità (per una motivazione di questa scelta apparentemente stravagante di coordinate si veda nel seguito la Sezione 6.4). Quindi qui ci conformiamo a questa abitudine: il versore della base canonica che viene spostato nel nuovo versore di osservazione è il terzo versore canonico  $e_3$ . Cominciamo con il determinare la rotazione richiesta: questo è il termine lineare  $3 \times 3$  della trasformazione affine che vogliamo trovare. dopo incorporeremo anche la traslazione per ottenere l'intera trasformazione affine  $4 \times 4$ .

LEMMA 4.2.1. Denotiamo come sempre con  $\mathbf{e}_1$ ,  $\mathbf{e}_2$ ,  $\mathbf{e}_3$  i vettori della base canonica in  $\mathbb{R}^3$ . Per ogni  $\mathbf{w} \neq \mathbf{0}$  prefissato in  $\mathbb{R}^3$ , la trasformazione lineare di  $\mathbb{R}^3$  che manda  $\mathbf{e}_3$  in un versore  $\mathbf{w}$  ed  $\mathbf{e}_1$  in un vettore che giace sul piano di base { $x_3 = 0$ } ha per matrice la matrice ortonormale

$$A = \left(\begin{array}{ccc} a & b & w_1 \\ c & d & w_2 \\ 0 & e & w_3 \end{array}\right)$$

dove a, b, c, d,  $e \in \mathbb{R}$ . In particolare, se la trasformazione manda la base canonica in una base ortogonale  $\mathbf{u}$ ,  $\mathbf{v}$ ,  $\mathbf{w}$  destrorsa (ossia con con orientamento concorde a quello della base canonica) con il primo versore sul piano di base, le colonne della matrice sono proporzionali a quelle della matrice ortogonale

$$B = \begin{pmatrix} -w_2 & -w_1w_3 & w_1 \\ w_1 & -w_2w_3 & w_2 \\ 0 & w_1^2 + w_2^2 & w_3 \end{pmatrix}$$

Normalizzando i vettori colonna della matrice B e rammentando che  $\mathbf{w}$  è già scelto di norma 1, si ottiene la forma esplicita della matrice ortogonale richiesta:

$$A = \begin{pmatrix} -\frac{w_2}{\sqrt{w_1^2 + w_2^2}} & -\frac{w_1 w_3}{\sqrt{w_1^2 + w_2^2}} & w_1 \\ \frac{w_1}{\sqrt{w_1^2 + w_2^2}} & -\frac{w_2 w_3}{\sqrt{w_1^2 + w_2^2}} & w_2 \\ 0 & \sqrt{w_1^2 + w_2^2} & w_3 \end{pmatrix}$$

DIMOSTRAZIONE. La prima espressione è immediata perché le colonne della matrice sono le immagini dei vettori canonici di base, ed  $\mathbf{e}_1$  viene mandato in un vettore con terza componente nulla. La seconda segue dal fatto che un vettore del tipo (a, b, 0) ortogonale a  $(w_1, w_2, w_3)$ è multiplo di  $(-w_2, w_1, 0)$ , e che questi due vettori perpendicolari determinano, a meno di multipli, un solo terzo vettore perpendicolare ad entrambi, che è  $(-w_1w_3, -w_2w_3, w_1^2 + w_2^2)$  (si veda la Sezione ? sul prodotto vettoriale).

Se la base  $\mathbf{u}$ ,  $\mathbf{v}$ ,  $\mathbf{w}$  nel Lemma 4.2.1 è ortonormale, la matrice ortogonale che ne risulta si esprime più chiaramente in termini di angoli anziché di coordinate. A questo scopo introduciamo gli angoli di Eulero come segue. DEFINIZIONE 4.2.2. (Angoli di Eulero.) Ogni vettore  $\mathbf{x} \in \mathbb{R}^3$  di norma  $\rho > 0$  si esprime in termini degli angoli  $\phi$  di latitudine e  $\theta$  di longitudine (detti angoli di Eulero) mediante la seguente trasformazione di coordinate:

$$\begin{aligned} x &= \rho \cos \theta \sin \varphi \\ y &= \rho \sin \theta \sin \varphi \\ z &= \rho \cos \varphi \,, \end{aligned}$$

 $\operatorname{con}\, 0 \leqslant \theta < 2\pi, \, 0 \leqslant \varphi \leqslant \pi.$ 



FIGURA 1. Angoli di Eulero

COROLLARIO 4.2.3. Sia **u**, **v**, **w** una base ortonormale destrorsa in  $\mathbb{R}^3$ , e scriviamo il vettore **w** in termini degli angoli di Eulero: **w** =  $(\cos \theta \sin \varphi, \sin \theta \sin \varphi, \cos \varphi)$ . Allora la matrice ortonormale A del Lemma 4.2.1 diventa

$$A = \begin{pmatrix} -\sin\theta & \cos\theta\cos\varphi & \cos\theta\sin\varphi\\ \cos\theta & \sin\theta\cos\varphi & \sin\theta\sin\varphi\\ 0 & -\sin\varphi & \cos\varphi \end{pmatrix}$$

oppure la matrice che si ottiene da questa cambiando i segni dei coefficienti delle prime due colonne (il che corrisponde a ruotare la base ortonormale di un angolo  $\pi$  intorno all'asse **w**).

### 4. TRASFORMAZIONI AFFINI

DIMOSTRAZIONE. Il vettore **u** deve giacere nel piano  $\{z = 0\}$ , quindi la condizione di ortogonalità con  $\mathbf{w} = (\cos \theta \sin \varphi, \sin \theta \sin \varphi, \cos \varphi)$ equivale a richiedere che **u** sia proporzionale a  $(-\sin \theta \sin \varphi, \cos \theta \sin \varphi, 0)$ . Osserviamo che la norma di quest'ultimo vettore è  $\sqrt{\sin^2 \varphi} = |\sin \varphi|$ , e  $\sin \varphi \ge 0$  perché  $0 \le \varphi \le \pi$ . Ne segue che  $\mathbf{u} = \pm (-\sin \theta, \cos \theta, 0)$ . (Nota: anche senza applicare la definizione di prodotto scalare euclideo, il fatto che il vettore **u** dipenda solo da  $\theta$  e non da  $\varphi$  è ovvio. Infatti esso è perpendicolare a **w** e giace sul piano di base, quindi se  $\varphi$ varia il vettore **w** ruota intorno all'asse generato da **u**: perciò **u** rimane invariante sotto questa rotazione, cioè non dipende da  $\varphi$ . Inoltre esso giace nel piano di base ed è perpendicolare a **w** la cui proiezione sul piano di base, data da  $(\cos \theta \sin \varphi, 0)$ , una volta normalizzata diventa  $(\cos \theta, \sin \theta, 0)$ . Quindi è chiaro che deve essere

$$\mathbf{u} = \pm (-\sin\theta, \cos\theta, 0). \tag{4.2.1}$$

Questo mostra quanto sia geometricamente più evidente ragionare con gli angoli di Eulero invece che con le coordinate cartesiane.)

A questo punto, il terzo vettore  $\mathbf{v}$  della terna ortonormale destrorsa si ottiene, come prima, calcolando il prodotto vettore  $\mathbf{w} \times \mathbf{u}$ . Il risultato è

$$\mathbf{v} = \pm (\cos\theta\cos\varphi, \sin\theta\cos\varphi, -\sin\varphi).$$

dove, affinché la terna sia ortonormale destrorsa, il segno deve essere lo stesso che in (4.2.1). (Nota: anche questo risultato è geometricamente evidente, perché **v**, essendo perpendicolare a **u** che è equatoriale, deve giacere nello stesso piano verticale in cui giace **w**, e quindi essere del tipo  $\mathbf{v} = (\cos\theta\sin\varphi', \sin\theta\sin\varphi', \cos\varphi')$  per qualche angolo  $\varphi' \in [0, \pi]$ . Ma siccome **v** è anche perpendicolare a **w**, che ha una inclinazione  $\varphi$  rispetto al piano equatoriale, la sua inclinazione rispetto al piano equatoriale, la sua inclinazione rispetto al piano equatoriale.

A questo punto la matrice della trasformazione affine dello spostamento della macchina da ripresa si ottiene immediatamente dal Corollario 4.1.8 (*ii*), ed è la seguente:

PROPOSIZIONE 4.2.4. La trasformazione affine di  $\mathbb{R}^3$  che trasla l'origine in  $\mathbf{v}$  e manda la base canonica in una base ortonormale di versori applicati  $\overrightarrow{\mathbf{vu}}$ ,  $\overrightarrow{\mathbf{vv}}$ ,  $\overrightarrow{\mathbf{vw}}$ , con primo versore parallelo al piano di base, agisce su  $\mathbb{R}^3$  come la restrizione all'iperpiano { $x_4 = 1$ } in  $\mathbb{R}^4$  della

120

trasformazione lineare su  $\mathbb{R}^4$  la cui matrice è

$$A = \begin{pmatrix} -\frac{w_2}{\sqrt{w_1^2 + w_2^2}} - v_1 & -\frac{w_1 w_3}{\sqrt{w_1^2 + w_2^2}} - v_2 & w_1 - v_3 & v_1 \\ \frac{w_1}{\sqrt{w_1^2 + w_2^2}} - v_1 & -\frac{w_2 w_3}{\sqrt{w_1^2 + w_2^2}} - v_2 & w_2 - v_3 & v_2 \\ -v_1 & \sqrt{w_1^2 + w_2^2} - v_2 & w_3 - v_3 & v_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Se questa matrice viene espressa in termini di angoli di Eulero, come nel Corollario 4.2.3, essa diventa

$$A = \begin{pmatrix} \sin \theta - v_1 & \cos \theta \cos \varphi - v_2 & \cos \theta \sin \varphi - v_3 & v_1 \\ \cos \theta - v_1 & \sin \theta \cos \varphi - v_2 & \sin \theta \sin \varphi - v_3 & v_2 \\ -v_1 & -\sin \varphi - v_2 & \cos \varphi - v_3 & v_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

## CAPITOLO 5

## Quaternioni e matrici di rotazione

In questo capitolo introduciamo un nuovo spazio vettoriale munito di una operazione di prodotto (associativa e distributiva rispetto alla somma, però non commutativa), che generalizza il campo dei numeri complessi, ed utilizziamo la sua operazione di prodotto per rappresentare in maniera computazionalmente efficiente le matrici di rotazione su  $\mathbb{R}^3$ . Per prima cosa, descriviamo in modo appropriato l'azione di una rotazione in  $\mathbb{R}^3$  intorno al proprio asse.

### 5.1. Espressione delle rotazioni in forma assiale

Consideriamo un operatore di rotazione R su  $\mathbb{R}^3$ . Ogni tale operatore ha un asse di rotazione: chiamiamo **n** il versore di tale asse (è tradizione orientarlo in maniera tale che la rotazione avvenga in senso antiorario intorno a **n** quando visto guardando verso l'origine a partire dal punto terminale del versore, ma qui non è essenziale). Vediamo come R opera su un generico vettore **r** scomponendo **r** in una componente assiale  $\mathbf{r}_{\parallel}$  ed una trasversale  $\mathbf{r}_{\perp}$ : ossia,

$$\mathbf{r}_{\shortparallel} = \langle \mathbf{r}, \mathbf{n} 
angle \mathbf{n}$$
  
 $\mathbf{r}_{\perp} = \mathbf{r} - \langle \mathbf{r}, \mathbf{n} 
angle \mathbf{n}$ 

(qui e nel resto del Capitolo usiamo la notazione  $\langle \mathbf{r}, \mathbf{n} \rangle \mathbf{n}$  invece che  $\mathbf{r} \cdot \mathbf{nn}$ per motivi di leggibilità). L'operatore R lascia invariata la componente assiale  $\mathbf{r}_{\parallel}$ ,

$$R\mathbf{r}_{\parallel} = \mathbf{r}_{\parallel}$$

e ruota la componente trasversale  $\mathbf{r}_{\perp}$  nel piano ortogonale a **n**. Esprimiamo dunque  $R\mathbf{r}_{\perp}$  nella base di questo piano formata da  $\mathbf{r}_{\perp}$  e da un vettore in questo piano ad esso ortogonale , ossia il prodotto vettore

$${f v}={f n} imes{f r}={f n} imes{f r}_{ot}$$
 ,

Osserviamo che  $\|\mathbf{v}\| = \|\mathbf{r}\| = \|R\mathbf{r}_{\perp}\|$ . Pertanto, se denotiamo con  $\theta$  l'angolo di rotazione, abbiamo

$$R\mathbf{r} = \cos\theta \,\mathbf{r}_{\perp} + \sin\theta \,\mathbf{v}$$
.

Pertanto

$$R\mathbf{r} = R\mathbf{r}_{\parallel} + R\mathbf{r}_{\perp} = \mathbf{r}_{\parallel} + \cos\theta \,\mathbf{r}_{\perp} + \sin\theta \,\mathbf{v}$$
$$= \langle \mathbf{r}, \mathbf{n} \rangle \mathbf{n} + \cos\theta \,(\mathbf{r} - \langle \mathbf{r}, \mathbf{n} \rangle \mathbf{n}) + \sin\theta \,\mathbf{v}$$
$$= \cos\theta \,\mathbf{r} + (1 - \cos\theta) \langle \mathbf{r}, \mathbf{n} \rangle \mathbf{n} + \sin\theta \,\mathbf{n} \times \mathbf{r} \,. \quad (5.1.1)$$

# 5.2. Rotazioni in $\mathbb{R}^2$ , numeri complessi ed estensione a tre dimensioni

Ora sviluppiamo in forma intuitiva l'idea di estensione dei numeri complessi ad uno spazio quadridimensionale dotato di una moltiplicazione ed utile a rappresentare le rotazioni in  $\mathbb{R}^3$ , come accennato all'inizio del capitolo. Riconsideriamo anzitutto le rotazioni bidimensionali in termini di numeri complessi. La rotazione di un angolo  $\theta$ intorno all'origine di  $\mathbb{R}^2$  si identifica con l'angolo  $\theta$ , o equivalentemente con il punto  $e^{i\theta}$  del cerchio unitario complesso. La moltiplicatività dell'esponenziale,  $e^{i(\theta_1+\theta_2)} = e^{i\theta_1} e^{i\theta_2}$ , assicura che il prodotto di due di queste matrici di rotazione (ossia la loro composizione) si associa al prodotto dei due corrispondenti numeri complessi (in altre parole, che la mappa dalle matrici di rotazione ai numeri complessi di modulo uno è un omomorfismo moltiplicativo, e quindi un isomorfismo - una mappa biunivoca che conserva le operazioni di prodotto - del gruppo delle matrici di rotazione su  $\mathbb{R}^2$  nel gruppo dei numeri complessi di modulo 1). Vogliamo fare la stessa cosa per le matrici di rotazione su  $\mathbb{R}^3$ . Questo gruppo, però, non è commutativo, e quindi dobbiamo costruire una estensione del campo complesso la cui operazione di prodotto non sia commutativa. Invece che aggiungere una dimensione immaginaria a  $\mathbb{R}$  per formare i numeri complessi x + iy, ora dobbiamo aggiungere tre dimensioni immaginarie, ossia una copia di  $\mathbb{R}^3$ , per formare una somma diretta  $\mathbb{H} = \mathbb{R} \oplus \mathbb{R}^3$  di uno spazio reale unidimensionale ed uno spazio *immaginario* a dimensione tre sul campo  $\mathbb{R}$ : quindi abbiamo bisogno di tre unità immaginarie, che chiamiamo  $i, j \in k$ . È opportuno considerare queste unità immaginarie come tre vettori: li denotiamo con i, j, k, e denotiamo il versore dell'asse reale con 1. (Anche se qui il prodotto scalare non viene impiegato, la visualizzazione geometrica viene facilitata dal pensare questi quattro vettori come versori ortogonali in  $\mathbb{R}^4$ .) Come nel caso dei numeri complessi, ora dovremmo scrivere espressioni come x + iy + jz + kw, oppure, in forma vettoriale,  $x + y\mathbf{i} + z\mathbf{j} + w\mathbf{k}$ . Però troveremo opportuno far agire sullo spazio H matrici, ovviamente a dimensione quattro, e saremo interessati in quelle matrici che preservano il sottospazio tridimensionale immaginario (ad esempio, visualizzeremo

124

### 5.3. QUATERNIONI

in questa maniera le matrici di rotazione su  $\mathbb{R}^3$ ). È conveniente considerare queste sottomatrici tridimensionali alla stregua del blocco tre per tre che rappresenta la componente lineare di una trasformazione affine a dimensione 4. Per questo preferiamo riordinare i quattro versori nell'ordine **i**, **j**, **k**, **1**: quindi scriveremo ix + jy + kz + w oppure  $x\mathbf{i} + y\mathbf{j} + z\mathbf{k} + w$ . Ovviamente stiamo per definire una operazione di somma con la proprietà commutativa, quindi in queste espressioni l'ordine è inessenziale, ma quando consideriamo azioni di matrici vogliamo che la quarta coordinata sia quella reale (che identifica i multipli di **1**).

## 5.3. Quaternioni

DEFINIZIONE 5.3.1. (Quaternioni.) Un quaternione  $\mathbf{q}$  è una espressione del tipo

$$\mathbf{q} = (\mathbf{q}_v, q_w) = (q_x, q_y, q_z, q_w) = iq_x + jq_y + kq_z + q_w, \quad (5.3.1)$$

dove  $q_x$ ,  $q_y$ ,  $q_z$  e  $q_w$  sono numeri reali,  $\mathbf{q}_v \in \mathbb{R}^3$  ed i simboli *i*, *j*, *k* soddisfano le proprietà seguenti:

$$i^{2} = j^{2} = k^{2} = -1$$
 (5.3.2)  
 $jk = -kj = i$   
 $ki = -ik = j$   
 $ij = -ji = k$ ,

Equivalentemente, lo spazio  $\mathbb{H}$  dei quaternioni può essere definito come lo spazio delle combinazioni lineari di quattro vettori **i**, **j**, **k**, **1** muniti della tabella di moltiplicazione (5.3.2).

Il vettore  $\mathbf{q}_v = (q_x, q_y, q_z)$  è chiamato parte immaginaria del quaternione  $\mathbf{q}$ , mentre  $q_w$  è detto parte reale.

Se estendiamo per linearità l'operazione di somma dalle singole componenti, lo spazio dei quaternioni diventa uno spazio vettoriale a dimensione 4 sul campo  $\mathbb{R}$ , che denotiamo con  $\mathbb{H}$ . Estendendo anche la moltiplicazione in maniera che rispetti la proprietà associativa e distributiva rispetto alla somma, muniamo lo spazio  $\mathbb{H}$  di una operazione di moltiplicazione.

In base a questa definizione, l'operazione di moltiplicazione tra due quaternioni  $\mathbf{q} \in \mathbf{r}$  soddisfa la tabella moltiplicativa che presentiamo in (5.3.3). Dalla tabella si vede che la moltiplicazione tra quaternioni non gode della proprietà commutativa.

## 5.3.1. Proprietà e definizioni.

Definizioni.

• *Moltiplicazione*: Denotando con  $\times$  il prodotto vettoriale in  $\mathbb{R}^3$  e con  $\cdot$  il prodotto scalare euclideo, abbiamo

$$\mathbf{qr} = (iq_x + jq_y + kq_z + q_w)(ir_x + jr_y + kr_z + r_w) = i(q_yr_z - q_zr_y + r_wq_x + q_wr_x) + j(q_zr_x - q_xr_z + r_wq_y + q_wr_y) + k(q_xr_y - q_yr_x + r_wq_z + q_wr_z) + q_wr_w - q_xr_x - q_yr_y - q_zr_z = (\mathbf{q}_v \times \mathbf{r}_v + r_w\mathbf{q}_v + q_w\mathbf{r}_v, q_wr_w - \mathbf{q}_v \cdot \mathbf{r}_v).$$
(5.3.3)

Definiamo ora la somma tra quaternioni, il coniugato, la norma e l'elemento identità.

• Somma:

$$\mathbf{q} + \mathbf{r} = (\mathbf{q}_v, q_w) + (\mathbf{r}_v, r_w) = (\mathbf{q}_v + \mathbf{r}_v, q_w + r_w).$$
(5.3.4)

• Coniugato:

$$\mathbf{q}^* = (\mathbf{q}_v, q_w)^* = (-\mathbf{q}_v, q_w).$$
 (5.3.5)

• Norma:

$$\|\mathbf{q}\|^{2} = \mathbf{q}\mathbf{q}^{*} = \mathbf{q}^{*}\mathbf{q} = \mathbf{q}_{v}\cdot\mathbf{q}_{v} + q_{w}^{2}$$
$$= q_{x}^{2} + q_{y}^{2} + q_{z}^{2} + q_{w}^{2}.$$
(5.3.6)

Si osservi che questa definizione fornisce la norma sui quaternioni, la quale peraltro coincide con la norma sullo spazio euclideo  $\mathbb{R}^4$  a cui i quaternioni sono isomorfi come spazio vettoriale.

• Elemento identità (per moltiplicazione):

$$\mathbf{i} = (\mathbf{0}, 1) \tag{5.3.7}$$

Per ogni elemento di norma non nulla esiste il corrispettivo elemento inverso rispetto all'operazione di moltiplicazione, ovvero il reciproco:

• Reciproco:

$$\mathbf{q}^{-1} = \frac{1}{n(\mathbf{q})}\mathbf{q}^* \tag{5.3.8}$$

Nella formula precedente per il reciproco compare la moltiplicazione per uno scalare, che è definita nel modo ovvio seguente:

• Moltiplicazione per scalare:

$$s\mathbf{q} = (\mathbf{0}, s)(\mathbf{q}_v, q_w) = (s\mathbf{q}_v, sq_w)$$
  
$$\mathbf{q}s = (\mathbf{q}_v, q_w)(\mathbf{0}, s) = (s\mathbf{q}_v, sq_w).$$
 (5.3.9)

126

Vediamo quindi che la moltiplicazione con scalare è commutativa, quindi:

$$s\mathbf{q} = \mathbf{q}s = (s\mathbf{q}_v, sq_w)$$
.

*Proprietà*. Dalle precedenti definizioni seguono le proprietà che ora elenchiamo.

• Proprietà del coniugato:

$$(\mathbf{q}^*)^* = \mathbf{q}$$
  
 $(\mathbf{q} + \mathbf{r})^* = \mathbf{q}^* + \mathbf{r}^*$   
 $(\mathbf{q}\mathbf{r})^* = \mathbf{r}^*\mathbf{q}^*.$  (5.3.10)

• Proprietà della norma:

$$|\mathbf{q}^*\| = \|\mathbf{q}\| \tag{5.3.11a}$$

$$\|\mathbf{qr}\| = \|\mathbf{q}\| \|\mathbf{r}\|.$$
 (5.3.11b)

• Proprietà della moltiplicazione:

– Bilinearità

$$\mathbf{p}(s\mathbf{q} + t\mathbf{r}) = s\mathbf{p}\mathbf{q} + t\mathbf{p}\mathbf{r}$$
  
(s\mathbf{p} + t\mathbf{q})\mathbf{r} = s\mathbf{p} + t\mathbf{q}\mathbf{r}. (5.3.12)

– Associatività

$$\mathbf{p}(\mathbf{qr}) = (\mathbf{pq})\mathbf{r} \,. \tag{5.3.13}$$

## • Quaternioni unitari

Consideriamo un quaternione  $\mathbf{q} = (\mathbf{q}_v, q_w)$  unitario, ossia tale che  $\|\mathbf{q}\| = 1$ . Grazie a (5.3.6) un quaternione è unitario se e solo se può essere scritto come

$$\mathbf{q} = (\mathbf{u}_{\mathbf{q}}\sin\omega, \cos\omega) \tag{5.3.14}$$

dove  $\mathbf{u}_{\mathbf{q}}$  è un versore in  $\mathbb{R}^3$ :  $\|\mathbf{u}_{\mathbf{q}}\|^2 = 1$ . Infatti, se  $\mathbf{q}$  è del tipo in (5.3.14),

$$\|\mathbf{q}\|^2 = \|(\mathbf{u}_{\mathbf{q}}\sin\omega,\cos\omega)\|^2 = (\mathbf{u}_{\mathbf{q}}\cdot\mathbf{u}_{\mathbf{q}})\sin^2\omega + \cos^2\omega$$
$$= \sin^2\omega + \cos^2\omega = 1,$$

e viceversa, se  $\mathbf{u}_{\mathbf{q}}$  è unitario, allora (5.3.14) segue dall'ultima espressione in (5.3.11) per la norma.

## 5.4. Rotazioni in $\mathbb{R}^3$ e coniugazione di quaternioni

Poiché la moltiplicazione di quaternioni è non commutativa, l'operazione di coniugazione,

$$\Omega_{\mathbf{q}}\mathbf{p}=\mathbf{q}\mathbf{p}\mathbf{q}^{-1},$$

è non banale. Inoltre, grazie alla moltiplicatività della norma (5.3.11b), se **q** è un quaternione unitario l'operatore lineare  $\Omega_{\mathbf{q}}$  preserva la norma, e quindi è un operatore ortogonale sullo spazio dei quaternioni  $\mathbb{H}$ pensato come spazio vettoriale reale a dimensione 4. Si noti che, se **q** è un quaternione unitario,  $\mathbf{q}^{-1}$  coincide con il coniugato **q** in base a (5.3.8), e quindi  $\Omega_{\mathbf{q}} = \mathbf{q}\mathbf{p}\mathbf{q}^*$ .

TEOREMA 5.4.1. (Coniugazione di quaternioni e rotazioni in  $\mathbb{R}^3$ .) Sia **p** un punto in uno spazio tridimensionale pensato come una classe di equivalenza per dilatazione in  $\mathbb{R}^4$  grazie alle sue coordinate omogenee  $(p_x, p_u, p_z, p_w)$ . Identifichiamo **p** con il quaternione

$$\mathbf{p} = \left( (p_x, p_y, p_z), p_w \right) = \left( \mathbf{p}_v, p_w \right).$$

Sia q un quaternione non nullo. Allora:

- (i) L'operatore di coniugazione  $\Omega_{\mathbf{q}}\mathbf{p} = \mathbf{q}\mathbf{p}\mathbf{q}^{-1}$  trasforma  $\mathbf{p} = (\mathbf{p}_v, p_w)$  in un quaternione  $\mathbf{p}' = (\mathbf{p}'_v, p_w) = (p'_x, p'_y, p'_z, p_w)$ , tale che  $\|\mathbf{p}_v\| = \|\mathbf{p}'_v\|$ , e quindi, ristretto al sottospazio tridimensionale dei quaternioni immaginari puri  $\mathbf{p}_v, 0$ ), è un operatore unitario (ossia di rotazione).
- (ii) Qualsiasi multiplo reale, diverso da zero, di q effettua la stessa trasformazione, e quindi la rotazione di punti tridimensionali, vista come coniugazione rispetto ad un opportuno quaternione, è compatibile con la rappresentazione di quei punti come classe di equivalenza di vettori in R<sup>4</sup>, ossia non dipende dalla scelta dei rappresentanti della classe di equivalenza.
- (iii) Se il quaternione  $\mathbf{q}$  è unitario e come in (5.3.14) lo scriviamo  $\mathbf{q} = (\mathbf{u}_{\mathbf{q}} \sin \omega, \cos \omega)$ , allora  $\Omega_{\mathbf{q}}$ , ristretto al sottospazio tridimensionale dei quaternioni immaginari puri, è l'operatore di rotazione antioraria di angolo  $2\omega$  attorno all'asse  $\mathbf{u}_{\mathbf{q}}$ . Qui la rotazione si intende antioraria quando vista da un osservatore orientato come il vettore  $\mathbf{u}_{\mathbf{q}}$  che guarda verso l'origine.

DIMOSTRAZIONE. Osserviamo anzitutto che la parte (*ii*) è banale, in quanto l'inversa di  $s\mathbf{q}$  è  $\mathbf{q}^{-1}s^{-1}$ , e la moltiplicazione con scalare gode della proprietà commutativa. Perciò  $(s\mathbf{q})\mathbf{p}(s\mathbf{q})^{-1} = s\mathbf{q}\mathbf{p}\mathbf{q}^{-1}s^{-1} =$  $\mathbf{q}\mathbf{p}\mathbf{q}^{-1}ss^{-1} = \mathbf{q}\mathbf{p}\mathbf{q}^{-1}$ . Possiamo quindi assumere  $\mathbf{q}$  come un quaternione unitario senza perdita di generalità. Per un quaternione unitario  $\mathbf{q}, \mathbf{q}^{-1} = \mathbf{q}^*$ ; possiamo quindi scrivere  $\mathbf{q}\mathbf{p}\mathbf{q}^{-1}$  come  $\mathbf{q}\mathbf{p}\mathbf{q}^*$ . Dimostriamo la parte (i). La parte reale di un qualsiasi quaternione, Re  $\mathbf{q}$ , può essere estratta usando la formula  $2 \operatorname{Re} \mathbf{q} = \mathbf{q} + \mathbf{q}^*$ . Consideriamo  $2 \operatorname{Re}(\mathbf{qpq}^*) = \mathbf{qpq}^* + (\mathbf{qpq}^*)^* = \mathbf{qpq}^* + \mathbf{qp}^*\mathbf{q}^*$ . Visto che la moltiplicazione tra quaternioni è bilineare, possiamo scrivere l'ultimo membro come  $\mathbf{q}(\mathbf{p} + \mathbf{q}^*)\mathbf{q}^* = 2\mathbf{q}\operatorname{Re}\mathbf{pq}^* = 2\operatorname{Re}\mathbf{p}$ . Quindi  $\mathbf{q}$  coniuga  $\mathbf{p} = (\mathbf{p}_v, p_w)$  in  $\Omega_{\mathbf{q}}\mathbf{p} = \mathbf{p}' = (\mathbf{p}'_v, p_w)$ , preservando la parte reale di  $\mathbf{p}$ . Inoltre l'operazione di moltiplicazione mantiene la norma perché la norma è moltiplicativa (si veda (5.3.11b)) e  $\mathbf{q}$  è unitario: quindi  $\|\mathbf{p}\| = \|\mathbf{p}'\|$ . Infine, poiché  $p_w$  resta inalterata,  $\|\mathbf{p}_v\| = \|\mathbf{p}'_v\|$ .

Per ultimo dimostriamo (*iii*). Abbiamo visto che si può scegliere  $\mathbf{q} = (\mathbf{q}_v, q_w)$  quaternione *unitario*. Sia  $\mathbf{p}$  un quaternione *immaginario* puro. Dall'ultima identità della regola di moltiplicazione (5.3.3) si verifica facilmente che

$$\Omega_{\mathbf{q}}\mathbf{p} = \left( (q_w^2 - \mathbf{q}_v \cdot \mathbf{q}_v)\mathbf{p}_v + 2(\mathbf{q}_v \cdot \mathbf{p}_v)\mathbf{q}_v + 2q_w(\mathbf{q}_v \times \mathbf{p}_v), 0 \right),$$

e poiché 
$$\mathbf{q} = \mathbf{u}_{\mathbf{q}} \sin \omega, \cos \omega$$
) abbiamo  

$$\Omega_{\mathbf{q}} \mathbf{p} = ((\cos^2 \theta - \sin^2 \theta) \mathbf{p}_v + 2\sin^2 \theta (\mathbf{p}_v \cdot \mathbf{u}_{\mathbf{q}}) \mathbf{u}_{\mathbf{q}} + 2\cos \theta \sin \theta \, \mathbf{u}_{\mathbf{q}} \times \mathbf{p}_v), 0$$

$$= (\cos 2\theta \, \mathbf{p}_v + (1 - \cos 2\theta) (\mathbf{p}_v \cdot \mathbf{u}_{\mathbf{q}}) \mathbf{u}_{\mathbf{q}} + \sin 2\theta \, \mathbf{u}_{\mathbf{q}} \times \mathbf{p}_v, 0).$$

ESEMPIO 5.4.2. Calcoliamo il quaternione unitario associato alla rotazione determinata di un angolo  $\omega$  in senso antiorario rispetto all'asse x, o y, o z.

Dalla parte (*iii*) del Teorema 5.4.1 si trova che, per ciascun versore canonico di base  $\mathbf{e}_1$ ,  $\mathbf{e}_2$ ,  $\mathbf{e}_3$ , il quaternione è  $\mathbf{q}_i = (\sin \omega \, \mathbf{e}_i, \, \cos \omega)$ .

### 5.4.1. Composizione di rotazioni e prodotto di quaternioni.

Vediamo come la composizione di rotazioni si associa al prodotto di quaternioni. Consideriamo due quaternioni unitari,  $\mathbf{q}_1 \in \mathbf{q}_2$ , ed un vettore  $\mathbf{p} \in \mathbb{R}^3$  trasformato in un quaternione mediante le sue coordinate omogenee,  $\mathbf{p} = (p_x, p_y, p_z, p_w)$ . In base al Teorema 5.4.1, l'operazione

## $\mathbf{q}_2(\mathbf{q}_1\mathbf{p}\mathbf{q}_1^*)\mathbf{q}_2^*$

ristretta al sottospazio tridimensionale dei quaternioni puramente immmaginari coincide la composizione delle rotazioni associate a  $\mathbf{q}_1$  ed a  $\mathbf{q}_2$ , in questo ordine. Ponendo  $\mathbf{q} = \mathbf{q}_2 \mathbf{q}_1$  la formula precedente può essere scritta come

 $\mathbf{qpq}^*$ .

Riassumendo,

COROLLARIO 5.4.3. La mappa  $\Omega$  è un omomorfismo dal gruppo moltiplicativo  $\mathbb{H}$  al gruppo delle matrici ortogonali reali su  $\mathbb{R}^4$ :

5.4.2. Matrice di rotazione in termini di quaternioni. Ora troviamo la forma matriciale dell'operatore  $\Omega_{\mathbf{q}}$  su  $\mathbb{R}^4 \sim \mathbb{H}$  dato dalla coniugazione con il quaternione  $\mathbf{q}$ . Ritorniamo, per maggiore generalità, al caso di un quaternione  $\mathbf{q}$  non necessariamente unitario.

Poiché la moltiplicazione fra quaternioni è bilineare, possiamo esprimere questa operazione sui quaternioni (pensati come vettori in  $\mathbb{R}^4$ ) tramite matrici a dimensione 4, spezzandola nella moltiplicazione sulla sinistra, **qp**, e la moltiplicazione sulla destra, **pq**<sup>\*</sup>.

Scriviamo  $\mathbf{L}^{q}\mathbf{p}$  la moltiplicazione sulla sinistra,  $\mathbf{p} \mapsto \mathbf{q}\mathbf{p}$ , con  $\mathbf{q} = (q_x, q_y, q_z, q_w) = (\mathbf{q}_v, q_w)$ . Segue immediatamente dalla regola di moltiplicazione (5.3.3) che la matrice associata all'operatore lineare  $\mathbf{L}^{q}$  è

$$\mathbf{L}^{q} = \begin{pmatrix} q_{w} & -q_{z} & q_{y} & q_{x} \\ q_{z} & q_{w} & -q_{x} & q_{y} \\ -q_{y} & q_{x} & q_{w} & q_{z} \\ -q_{x} & -q_{y} & -q_{x} & q_{w} \end{pmatrix}$$

Scriviamo ora la moltiplicazione sulla destra,  $\mathbf{p} \mapsto \mathbf{p}\mathbf{q}^*$  come  $\mathbf{R}^{q^*}\mathbf{p}$ , dove l'operatore  $\mathbf{R}^{q^*}$  viene espresso, grazie alla regola di moltiplicazione, come

$$\mathbf{R}^{q^{*}} = \begin{pmatrix} q_{w} & -q_{z} & q_{y} & -q_{x} \\ q_{z} & q_{w} & -q_{x} & -q_{y} \\ -q_{y} & q_{x} & q_{w} & -q_{z} \\ q_{x} & q_{y} & q_{x} & q_{w} \end{pmatrix}$$

Ora un calcolo elementare ma tedioso mostra che la matrice  $\mathbf{M}^{q}$  associata all'operatore di coniugazione  $\Omega_{q}$  è

$$\mathbf{M}^{\mathbf{q}} = \mathbf{L}^{q} \mathbf{R}^{q^{*}}$$

$$= \begin{pmatrix} q_{w}^{2} + q_{x}^{2} - q_{y}^{2} - q_{z}^{2} & 2(q_{x}q_{y} - q_{w}q_{z}) & 2(q_{x}q_{z} + q_{w}q_{y}) & 0 \\ 2(q_{x}q_{y} + q_{w}q_{z}) & q_{w}^{2} - q_{x}^{2} + q_{y}^{2} - q_{z}^{2} & 2(q_{y}q_{z} - q_{w}q_{x}) & 0 \\ 2(q_{x}q_{z} - q_{w}q_{y}) & 2(q_{y}q_{z} + q_{w}q_{x}) & q_{w}^{2} - q_{x}^{2} - q_{y}^{2} + q_{z}^{2} & 0 \\ 0 & 0 & 0 & q_{w}^{2} + q_{x}^{2} + q_{y}^{2} + q_{z}^{2} \end{pmatrix}$$

Semplificando otteniamo

$$\mathbf{M}^{\mathbf{q}} = \begin{pmatrix} \|\mathbf{q}\|^2 - 2(q_y^2 + q_z^2) & 2(q_x q_y - q_w q_z) & 2(q_x q_z + q_w q_y) & 0\\ 2(q_x q_y + q_w q_z) & \|\mathbf{q}\|^2 - 2(q_x^2 + q_z^2) & 2(q_y q_z - q_w q_x) & 0\\ 2(q_x q_z - q_w q_y) & 2(q_y q_z + q_w q_x) & \|\mathbf{q}\|^2 - 2(q_x^2 + q_y^2) & 0\\ 0 & 0 & 0 & \|\mathbf{q}\|^2 \end{pmatrix}$$

Nel caso il quaternione **q** sia unitario si ha una ulteriore semplificazione:

$$\mathbf{M}^{\mathbf{q}} = \begin{pmatrix} 1 - 2(q_y^2 + q_z^2) & 2(q_x q_y - q_w q_z) & 2(q_x q_z + q_w q_y) & 0\\ 2(q_x q_y + q_w q_z) & 1 - s(q_x^2 + q_z^2) & 2(q_y q_z - q_w q_x) & 0\\ 2(q_x q_z - q_w q_y) & 2(q_y q_z + q_w q_x) & 1 - 2(q_x^2 + q_y^2) & 0\\ 0 & 0 & 0 & 1 \end{pmatrix}$$
(5.4.1)

Questo dimostra la prima parte del seguente risultato:

COROLLARIO 5.4.4. (Matrici di rotazione espresse in termini di quaternioni.) Consideriamo la matrice di rotazione (ossia ortogonale reale con determinante +1)  $\mathbf{M}^q$  dell'operatore di coniugazione  $\Omega_{\mathbf{q}}$ dove  $\mathbf{q}$  è un quaternione unitario: essa ha la forma espressa in (5.4.1). Viceversa, ogni matrice di rotazione su  $\mathbb{R}^3$ , espressa nel modo seguente in forma di matrice affine a dimensione quattro,

$$\mathbf{M}^{\mathbf{q}} = \begin{pmatrix} m_{00} & m_{01} & m_{02} & 0\\ m_{10} & m_{11} & m_{12} & 0\\ m_{20} & m_{21} & m_{22} & 0\\ 0 & 0 & 0 & 1 \end{pmatrix} ,$$

è associata al quaternione unitario  $\mathbf{q} = (q_x, q_y, q_z q_w)$  dato da

$$q_{w} = \pm \frac{1}{2} \sqrt{m_{00} + m_{11} + m_{22} + 1} = \pm \frac{1}{2} \sqrt{\operatorname{tr}(\mathbf{M}^{\mathbf{q}})}$$

$$q_{x} = \frac{m_{21} - m_{12}}{4q_{w}}$$

$$q_{y} = \frac{m_{02} - m_{20}}{4q_{w}}$$

$$q_{z} = \frac{m_{10} - m_{01}}{4q_{w}}.$$
(5.4.2)

DIMOSTRAZIONE. Dobbiamo solo dimostrare la seconda parte dell'enunciato, ossia la ricostruzione del quaternione a partire dalla matrice. Questo significa invertire la prima parte dell'enunciato, ossia ricavare il quaternione dall'espressione (5.4.1). Da questa espressione si vede subito che la traccia (ossia la somma dei coefficienti diagonali) della matrice è  $4(1 - q_x^2 - q_y^2 - q_z^2)$ . Poiché **q** è un quaternione unitario,  $q_x^2 + q_y^2 + q_z^2 + q_w^2 = 1$  e quindi

$$\operatorname{tr}\left(\mathbf{M}^{\mathbf{q}}\right) = 4q_w^2 \,.$$

Questo prova la prima identità in (5.4.2). Esaminando ancora (5.4.1) si vede che  $m_{21} = 2(q_wq_x + q_yq_z)$  e  $m_{12} = 2(q_yq_z - q_wq_x)$ , da cui segue la seconda identità in (5.4.2). Le restanti due identità si provano allo stesso modo.

## 132 5. QUATERNIONI E MATRICI DI ROTAZIONE

Esprimere le matrici di rotazione in termini di quaternioni è una notevole opportunità di ridurre la mole di calcoli in Computer Graphics, dove le rotazioni intervengono al cambiare della posizione dell'osservatore (come accade continuamente durante le animazioni). Esewguire il calcolo tramite moltiplicazione di quaternioni è numericamante vantaggioso, ma soprattutto questa procedura è facile da implementare in hardware, dal momento che la moltiplicazione di due quaternioni è lineare nelle coordinate di entrambi: l'implementazione in hardware permette di eseguirla in maniera velocissima. Pertanto può essere utile svolgere i seguenti esercizi.

ESERCIZIO 5.4.5. Rivedere il precedente esempio 5.4.2 e ritrovarne il risultato mediante il Corollario 5.4.4.

ESERCIZIO 5.4.6. Calcolare il quaternione unitario associato alla rotazione determinata dagli angoli di Eulero  $\theta \in \phi$  introdotti nella Definizione 4.2.2.

ESERCIZIO 5.4.7. Calcolare il quaternione associato alla componente di rotazione della matrice affine di rototraslazione dello spostamento della macchina da ripresa, calcolata nel Corollario 4.2.3.

## CAPITOLO 6

## Trasformazioni prospettiche

Questo capitolo presenta vari tipi di trasformazioni prospettiche (dette anche *assonometriche*). Queste trasformazioni si suddividono in due categorie: proiezione centrale, se ci sono fasci di rette parallele che dopo la proiezione convergono verso opportuni punti di fuga, oppure proiezioni parallele, se questo non succede. Nel caso delle proiezioni parallele, i fasci di rette parallele rimangono paralleli, e la proiezione è determinata dalla scelta di un piano di proiezione e dalla direzione di proiezione. Se tale direzione è perpendicolare al piano di proiezione si dice che la proiezione è ortogonale (o ortografica), altrimenti obliqua. In generale il piano di proiezione non contiene l'origine, e quindi le proiezioni prospettiche non fissano l'origine e pertanto non sono operazioni lineari. Vedremo che la proiezione centrale si rappresenta con matrici proiettive, le altre con matrici affini. La proiezione centrale è quella più videorealistica, utilizzata in Computer Graphics, e la studiamo per prima (un caso particolarmente semplice nella Sezione 6.1, il caso generale nelle Sezioni  $6.4 \, \mathrm{e} \, 6.5$ ). Rispetto alle proiezioni parallele, trattate sistematicamente nella Sezione 6.6, la proiezione ortogonale è anticipata alla Sezione 6.2, in modo da poter fornire una versione unificata della forma matriciale delle proiezioni centrale ed ortogoonale nella Sezione 6.3. Sarebbe possibile estendere questa trattazione unificata anche alle altre proiezioni parallele, ma lasciamo questo tedioso compito al lettore.

Nel calcolare le matrici delle trasformazioni prospettiche dobbiamo mettere in guardia il lettore che la Computer Graphics ha una tradizione assai peculiare, quella di far agire le matrici sui vettori non da sinistra ma da destra. La ragione storica di ciò è che i primordi della Computer Graphics furono sviluppati non da matematici, bensì da studiosi a cui pareva strano che se due operatori  $A \in B$  agiscono su un vettore **p** in questo ordine allora si debba avere che la composizione dei due uno dopo l'altro si debba scrivere  $BA\mathbf{p} = B(A\mathbf{p})$  invece che  $AB\mathbf{p}$ . Per rovesciare l'ordine con cui i due simboli si succedono sulla carta, questi studiosi preferirono scrivere l'azione da destra, in modo che la composizione diventasse  $\mathbf{p}AB = (\mathbf{p}A)B$ . Per nostra fortuna a questo punto il lettore avrà studiato la matematica e compreso la ragione dell'ordine naturale, e gli sarà facile capire gli articoli di Computer Graphics, nei quali purtroppo l'ordine è opposto.

## 6.1. Prospettiva centrale, proiezione standard

Riprendendo in esame l'Esempio 3.2.4, osserviamo anzitutto, a scanso di malintesi, che anche quando i rappresentanti delle classi in  $\mathbb{P}^n$ si scelgono del tipo  $(x_1, x_2, \ldots, x_{n-1}, 1)$ , la forma della matrice in  $PGL_n(\mathbb{R})$  associata ad una trasformazione prospettica non ha necessariamente l'ultima riga  $(0, \ldots, 0, 1)$ , come invece avviene per le matrici delle trasformazioni affini (Proposizione 4.1.4). Infatti questa sarebbe la forma giusta per l'azione sui punti al finito se la trasformazione mandasse l'iperpiano  $\{x_n = 1\}$  in sé, come appunto avviene per le trasformazioni affini di  $\mathbb{R}^{n-1}$  quando considerate come trasformazioni lineari in  $\mathbb{R}^n$  che lasciano invariante tale iperpiano: ma ciò non avviene per le trasformazioni proiettive, per le quali l'azione sui vettori non è neppure definita, visto che i rappresentanti delle classi dell'equivalenza proiettiva possono essere dilatati (e quindi uscire dal suddetto piano) senza che l'azione ne risenta. Abbiamo visto un esempio concreto alla fine dell'Esempio 3.2.4, ed esattamente in (3.2.2), che riprendiamo in esame nella prossima Sezione.

Come è consuetudine in Computer Graphics, chiamiamo  $x \in y$  le coordinate orizzontale e verticale, e z la profondità, orientata in modo da aumentare dall'origine verso l'osservatore. Il piano di visuale è parallelo agli assi  $x \in y$  e quindi perpendicolare all'asse z, diciamo in posizione z = d (qui stiamo supponendo che l'osservatore non si trovi verticalmente sopra l'origine, cioè sul piano x-y: in genere, in Computer Graphics, l'osservatore è poco al di sopra dell'asse z, nel senso che la sua posizione ha z positivo grande, y non molto grande e positivo, e x = 0). Si osservi che non stiamo richiedendo che il punto **p** abbia valore positivo di z: anzi, se invece che modellare un osservatore che guarda una scena stessimo modellando una camera oscura come un cubo con un piccolo foro nell'origine disposto nel semispazio  $z \leq 0$ , il piano di visuale, cioè in questo caso il piano della pellicola, passerebbe attraverso l'interno del cubo, e quindi avrebbe d negativo: in tal caso l'immagine creata dalla proiezione sarebbe ribaltata rispetto alla scena reale, perché i raggi si incrociano nel passare tutti attraverso l'origine. Con questa scelta di coordinate, la proiezione sul piano di visuale porta gli assi  $x \in y$  della scena tridimensionale su rispettivi assi orizzontale e verticale in tale piano: quindi i nomi delle coordinate sono quelli naturali per la compatibilità, perché se immaginiamo che questo sia il piano del monitor, è naturale chiamare questi assi del monitor  $x \in y$ , rispettivamente.

Fissiamo ora in due modi diversi i parametri della proiezione prospettica della prospettiva centrale. Il primo modo è quello più naturale se si pensa di aver fissato una volta per tutte la posizione d del piano di visuale. In tal caso, collochiamo per ora l'osservatore nell'origine: cioè, l'origine è il *centro di proiezione*. Tratteremo nella prossima Sezione 6.4 il caso generale in cui il centro di proiezione è generico (non necessariamente l'origine).

La proiezione manda il generico punto  $\mathbf{q} = (x, y, z)$  sul punto determinato sul piano di visuale dall'intersezione con la retta che passa per l'origine e per il punto  $\mathbf{q}$ . Riconosciamo in questo modo di procedere l'analogia con il principio della proiezione stereografica nella geometria proiettiva (Sezione 3.1).

La proiezione avviene quindi tramite una similitudine, cioè una proporzione: il punto  $\mathbf{q}$  viene mandato in



FIGURA 1. Proiezione della prospettiva centrale: piano di visuale in $\boldsymbol{z}=\boldsymbol{d}$ 

Possiamo riformulare questo risultato in termini di trasformazioni prospettiche: la trasformazione prospettica si scrive in termini di una (classe di equivalenza di) matrice  $M_d^{(c)}$  in  $PM_4(\mathbb{R})$  come in (3.2.1) dell'Esempio 3.2.4. Il punto **p** corrisponde ad un punto proiettivo al finito [x, y, z, 1], e si ha

$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} = M_d^{(c)} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{pmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} .$$
(6.1.1)

Quindi  $[X, Y, Z, W] = [x, y, z, \frac{z}{d}]$ , e la quarta coordinata omogenea vale W = z/d. Perciò come rappresentante della classe immagine

[X, Y, Z, W]

possiamo scegliere il consueto rappresentante stereografico

$$\left(\frac{X}{W}, \frac{Y}{W}, \frac{Z}{W}, 1\right),\,$$

che corrisponde in  $\mathbb{R}^3$  al punto  $(x_p, y_p, z_p) = \left(\frac{x}{z/d}, \frac{y}{z/d}, d\right).$ 

Ora veniamo al secondo modo utile di fissare i parametri prospettici. Poniamo il centro di prospettiva non più nell'origine, bensì in (0, 0, -d), ed il piano di visuale in z = 0 (è consuetudine in Computer Graphics, per semplificare il processo di trasformazione da coordinate tridimensionali nel piano di visuale in  $\mathbb{R}^3$  a coordinate bidimensionali del monitor, collocare il piano di visuale in  $\{z = 0\}$ ). In tal modo, quando facciamo crescere la distanza d), il piano di visuale non si sposta, e possiamo più agevolmente confrontare i risultati della trasformazione prospettica).

Lo stesso argomento di proporzionalità adottato prima ora porta alle seguenti equazioni:

$$\frac{x_p}{d} = \frac{x}{z+d}$$

$$\frac{y_p}{d} = \frac{y}{z+d}$$

$$z_p = 0$$
(6.1.2)

da cui si ricava

$$x_p = \frac{x}{1 + \frac{z}{d}}$$
$$y_p = \frac{y}{1 + \frac{z}{d}}$$
$$z_p = 0.$$



FIGURA 2. Proiezione della prospettiva centrale: piano di visuale in z = 0

Pertanto ora la "matrice"  $M_d^{(c)}$  della trasformazione prospettica è la classe di equivalenza, cioè l'elemento in  $PM_4(\mathbb{R})$ , che ha per rappresentante la matrice

$$M_d^{(c)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/d & 1 \end{pmatrix} .$$
(6.1.3)

Poiché gli elementi di  $PM_4(\mathbb{R})$  sono classi di equivalenza per dilatazione, possiamo scegliere un altro rappresentante della stessa classe, dilatando quello appena scritto in modo da eliminare il denominatore. Così si ottiene la seguente forma della matrice della trasformazione prospettica:

$$M_d^{(c)} = \begin{pmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & d \end{pmatrix} \quad . \tag{6.1.4}$$

Si noti che questa forma corrisponde nel modo più naturale alle equazioni (6.1.2) della trasformazione.

ESERCIZIO 6.1.1. Consideriamo la prospettiva centrale con centro di proiezione ubicato sull'asse z al punto z = -d e piano di proiezione  $\{z = 0\}$ , la quale porta alla matrice di proiezione prospettica (6.1.3). Sia S una superficie emisferica di raggio 1 nel semispazio  $\{x \ge 0\}$  con

centro nel punto (0, 0, 2), e C un cilindro di raggio  $\frac{1}{2}$  avente per asse centrale la retta  $\{y = 0, z = 2\}$ . Sia  $J = S \cap C$ . Calcolare l'immagine prospettica di J sul piano di proiezione.

SVOLGIMENTO. Anzitutto determiniamo J. L'equazione di S è  $x^2 + y^2 + (z-2)^2 = 1, x \ge 0$ . L'equazione di C è  $y^2 + (z-2)^2 = \frac{1}{4}$ . Quindi i punti di J sono tutti e soli quelli che soddisfano le seguenti equazioni e disequazioni:

$$x \ge 0$$
  
$$x^{2} + y^{2} + (z - 2)^{2} = 1$$
  
$$y^{2} + (z - 2)^{2} = \frac{1}{4}$$

Se ne ricava

$$x^{2} = \frac{3}{4}$$
$$x \ge 0$$
$$y^{2} + (z - 2)^{2} = \frac{1}{4}$$

ossia  $x = \sqrt{3}/2$ ,  $y^2 + (z - 2)^2 = \frac{1}{4}$ . Si tratta, ovviamente, di una circonferenza sul piano  $x = \sqrt{3}/2$ , che parametrizziamo nel modo seguente:

$$x = \sqrt{3}/2 \tag{6.1.5}$$

$$y = \frac{1}{2}\cos t \tag{6.1.6}$$

$$z = 1 + \frac{\sin t}{2} \tag{6.1.7}$$

dove l'angolo t varia fra 0 e  $2\pi$ . Scriviamo i punti di J in termini di coordinate omogenee (x, y, z, w) con quarta coordinata w = 1 (scelta del rappresentante stereografico per punti al finito), ed applichiamo la matrice di proiezione prospettica (6.1.3). Analogamente a (6.1.1), da

$$(6.1.3) = 6.1.5$$
 si ottiene

$$\begin{bmatrix} X\\Y\\Z\\W \end{bmatrix} = M_d^{(c)} \begin{bmatrix} x\\y\\z\\1 \end{bmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 1/d & 1 \end{pmatrix} \begin{bmatrix} \sqrt{3}/2\\\frac{1}{2}\cos t\\1+\frac{1}{2}\sin t\\1 \end{bmatrix}$$
(6.1.8)
$$= \begin{bmatrix} \sqrt{3}/2\\\frac{1}{2}\cos t\\0\\1+\frac{1+\frac{1}{2}\sin t}{d} \end{bmatrix} .$$
(6.1.9)

Ora riportiamo in coordinate omogenee con quarta componente 1 il punto proiettivo (X, Y, Z, W) così calcolato, dividendo per il valore di W: si ottiene così un altro rappresentante della classe di equivalenza proiettiva [X, Y, Z, W], e precisamente il punto (x, y, z, 1) dove

$$x = \frac{d}{d+1+\frac{1}{2}\sin t} \frac{\sqrt{3}}{2}$$
$$y = \frac{d\cos t}{2(d+1+\frac{1}{2}\sin t)}$$
$$z = 0.$$

Quindi la curva immagine di J sul piano di proiezione  $\{z = 0\}$ , munito delle coordinate  $x \in y$ , ha la seguente equazione:

$$x = \frac{\sqrt{3d}}{2d + 2 + \sin t}$$
$$y = \frac{d\cos t}{2d + 2 + \sin t}$$

Si osservi che, se facciamo tendere  $d = -\infty$  (ossia se passiamo alla proiezione ortogonale sul piano  $\{z = 0\}$ , si ottiene, come previsto,  $x = \sqrt{3}/2$ ,  $y = \frac{1}{2}\cos t$ : al variare di t questo punto immagine si muove sul segmento  $x = \sqrt{3}/2$ ,  $-1/2 \leq y \leq 1/2$ , che è esattamente la proiezione ortogonale dell'intersezione fra sfera e cilindro, perché essa è una circonferenza nello spazio tridimensionale che giace sul piano  $x = \sqrt{3}/2$ , ha componente y del centro uguale a 0 e raggio 1/2.  $\Box$ 

NOTA 6.1.2. (Punto di fuga della proiezione standard.) È interessante osservare che, in questo caso particolare della proiezione prospettica centrale, che si chiama la *proiezione standard*, il piano di visuale è ortogonale alla direzione dall'osservatore all'origine. Consideriamo un fascio di rette perpendicolare al piano di visuale: nella proiezione standard nella forma appena sviluppata, si tratta del fascio delle rette parallele all'asse z. Chiamiamo  $\mathbf{r_0} = (x_0, y_0, 0)$  il punto in cui una tale retta **r** interseca il piano  $\{z = 0\}$ : allora la equazione parametrica di **r** è  $\mathbf{r}(t) = \mathbf{r_0} + t\mathbf{e_3}$ . Scegliamo  $\mathbf{r_0}$  nel piano di proiezione  $\{z = 0\}$ , diciamo  $\mathbf{r_0} = (a, b, 0)$ , e poniamo  $\mathbf{r}(t) = (x(t), y(t), z(t)) =$ (a, b, t). Associamo ai punti della retta **r** i rappresentanti speciali delle loro classi di equivalenza proiettiva, ossia, in coordinate omogenee,  $\mathbf{R}(t) = (x(t), y(t), z(t), 1) = (a, b, t, 1)$ ; analogamente, scriviamo **S** per la corrispondente estensione quadridimensionale di ogni altra retta **s**. Quando applichiamo la trasformazione trovata in (6.1.4), la retta **R** viene trasformata nella retta **S** = T**R** di equazioni parametriche

$$\mathbf{S}(t) = M_d^{(c)} \mathbf{R}(t) = \begin{pmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & d \end{pmatrix} \begin{pmatrix} a \\ b \\ t \\ 1 \end{pmatrix} = \begin{pmatrix} da \\ db \\ 0 \\ t+d \end{pmatrix}.$$

Riconduciamo il risultato alla consueta espressione stereografica delle classi di equivalenza proiettive, ossia con i rappresentanti speciali di ultima coordinata 1, rinormalizzando con la divisione per t + d. Si ottiene

$$\mathbf{S}(t) = \left(\frac{da}{t+d}, \frac{db}{t+d}, 0, 1\right). \tag{6.1.10}$$

La retta immagine tridimensionale  $\mathbf{s}$ , che si ottiene considerando le prime tre componenti dopo questa normalizzazione, è

$$\mathbf{s}(t) = \left(\frac{da}{t+d}, \frac{db}{t+d}, 0\right) \,. \tag{6.1.11}$$

Osserviamo che essa giace, come deve essere, nel piano  $\{z = 0\}$  (il piano di visuale), e quando  $t \to \infty$  tende all'origine, quali che siano a e b. In altre parole, il fascio di rette parallele perpendicolare al piano di visuale viene trasformato nel fascio delle semirette radiali nel piano  $\{z = 0\}$ . L'origine è quindi il *punto di fuga* prospettico di questo fascio di rette.

Osserviamo che l'origine è esattamente il punto ottenuto sommando al centro di proiezione il versore direzionale del fascio di rette moltiplicato per la distanza d fra il centro di proiezione ed il piano di visuale. Se avessimo scelto un centro di proiezione diverso da (0, 0, -d), avremmo potuto ripetere lo stesso calcolo della trasformazione prospettica, che svolgeremo in dettaglio nella prossima Sezione 6.4. A partire da esso, riprenderemo in esame e generalizzeremo questo risultato nella Sezione 6.5.

Per finire, consideriamo come è fatta la proiezione prospettica di una retta che attraversa il piano di visuale, riconsiderando la retta immagine s del fascio ortogonale al piano di visuale,  $\mathbf{r}(t) = (a, b, t)$ , che è stata calcolata in (6.1.11):  $\mathbf{s}(t) = \left(\frac{da}{t+d}, \frac{db}{t+d}, 0\right)$ . Facciamo variare il parametro t in modo che il punto  $\mathbf{r}(t)$  passi dal semispazio anteriore all'osservatore posto nel centro di prospettiva (0, 0, -d) a quello posteriore, ossia da t > -d a t < -d. Quando t decresce verso -d il punto  $\mathbf{s}(t)$  si muove nel piano di visuale radialmente fuori dall'origine (ossia il punto di fuga) verso l'infinito: più precisamente, verso il punto all'infinito di questo piano le cui coordinate proiettive sono (a, b, 0). Quando t diventa inferiore a -d e continua a decrescere, il punto  $\mathbf{s}(t)$ salta dalla parte opposta, sulla stessa retta radiale ma sulla semiretta opposta (come succede ai raggi che passano per una lente quando la scena e l'osservatore sono ai lati opposti dell'obiettivo, che è il centro di proiezione), e si avvicina dall'infinito al punto di fuga. Quindi una retta che attraversa il piano di visuale ha immagine che va all'infinito con un salto. Questo fatto crea una distorsione prospettica assai drastica: si provi ad immaginare l'immagine prospettica del cubo unitario quando l'osservatore si trova dentro il cubo! Per quato motivo, in Computer Graphics, il centro di prospettiva, ossia l'osservatore, si trova sempre da un lato del piano di visuale e la scena osservata dall'altro lato. 

ESEMPIO 6.1.3. (Trasformazione prospettica standard del cubo unitario.) Consideriamo il cubo Q i cui vertici sono  $\{\pm \mathbf{e_1} + \pm \mathbf{e_2} + \pm \mathbf{e_3}\}$ . È improprio chiamare unitario questo cubo, perché ha lato 2, ma possiamo sempre dividere per due alla fine se proprio vogliamo, e quindi procediamo con questa scelta, che evita i denominatori. Consideriamo le classi di equivalenza proiettiva dei vettori  $\mathbf{e_i}$ , scrivendo ad esempio i loro rappresentanti standard come  $\pm \mathbf{E_1}^{\pm} = (\pm 1, 0, 0, 1)$ ,  $\mathbf{E_2}^{\pm} = (0, \pm 1, 0, 1), \mathbf{E_3}^{\pm} = (0, 0, \pm 1, 1)$ . Applichiamo agli  $\mathbf{E_i}$  la matrice  $M_d^{(c)}$  in (6.1.4) e scriviamo  $\mathbf{V_i}^{\pm} = M_d^{(c)} \mathbf{E_i}^{\pm}$ . Si ottiene  $\mathbf{V_i}^{\pm} = \mathbf{E_i}^{\pm}$ per i = 1 e 2,  $\mathbf{eV_3}^{\pm} = (0, 0, 0, d \pm 1)$ .

Rinormalizzando per ritornare ai rappresentanti proiettivi standard con 1 alla quarta componente, e considerando solo le prime tre componenti del risultato per ritrovare i vettori tridimensionali trasformati  $\mathbf{v_i}$ , e scrivendo M per la trasformazione prospettica tridimensionale in tal modo ottenuta da  $M_d^{(c)}$ , si verifica subito che si ha

T

$$T(1,1,1) = \frac{d}{d+1} (1,1,0)$$
$$T(1,1,-1) = \frac{d}{d-1} (1,1,0)$$
$$T(1,-1,1) = \frac{d}{d+1} (1,-1,0)$$
$$T(1,-1,-1) = \frac{d}{d-1} (1,-1,0)$$
$$T(-1,1,1) = \frac{d}{d+1} (-1,1,0)$$
$$T(-1,1,-1) = \frac{d}{d-1} (-1,1,0)$$
$$T(-1,-1,1) = \frac{d}{d+1} (-1,-1,0)$$
$$(-1,-1,-1) = \frac{d}{d-1} (-1,-1,0)$$

Consideriamo allora i quattro vertici anteriori visti dall'osservatore (ovvero i punti  $(\pm 1, \pm 1, -1)$ : non dimentichiamo che l'osservatore è in (0, 0, -d), sull'asse z negativo). Abbiamo appena visto che essi vengono compressi di un fattore 1/(d-1), mentre i quattro vertici posteriori di un fattore più grande, 1/(d+1). Questa è la compressione prospettica. Qui abbiamo supposto d > 1. Se 0 < d < 1 allora d-1 < 0 ed i vertici anteriori sono in realtà alle spalle dell'osservatore, e vengono quindi scambiati di segno, ossia ribaltati, come abbiamo visto alla fine della Nota 6.1.2. Se poi d = 1, allora i vertici anteriori sono ai lati dell'osservatore, e la proiezione prospettica non è definita su di essi (le semirette da essi al centro di proiezione non passano per il piano di visuale  $\{z = 0\}$ , restano tutte nel piano  $\{z = -1\}$ .

### 6.2. Proiezione prospettica ortogonale (o ortografica)

La proiezione ortogonale (detta anche ortografica) è la trasformazione prospettica data dalla proiezione perpendicolare al piano di visuale, che potrebbe essere un piano che non contiene l'origine, quindi non un sottospazio vettoriale ma un suo traslato (ed in tal caso l'origine non può essere preservata dalla proiezione, che pertanto non è una applicazione lineare). Però manteniamo la scelta di localizzazione del piano di visuale fatta alla fine della precedente Sezione 6.4, e quindi il piano di visuale passa per l'origine: è il piano {z = 0}, quindi un sottospazio. La
proiezione ortogonale all'asse z è quindi la trasformazione che manda il punto  $\mathbf{p} = (x, y, z)$  nel punto di tale piano ottenuto ponendo uguale a zero la componente z, cioè (x, y, 0). In coordinate omogenee stiamo ponendo



FIGURA 3. Proiezione della prospettiva ortogonale sul piano z = 0

Pertanto la "matrice" prospettica ora è

$$M^{(o)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} .$$
 (6.2.1)

NOTA 6.2.1. Si osservi che la matrice  $M^{(o)}$  si ottiene facendo tendere dad infinito nell'espressione per  $M_d^{(c)}$  (si veda anche come, nell'Esempio 6.1.3, il trasformato prospettico del cubo unitario in  $\mathbb{R}^3$  tende al cubo unitario nel piano di proiezione  $\mathbb{R}^2$  quando  $d \to \infty$ , e quindi tende alla proiezione ortogonale del cubo). In effetti questo fatto è naturale, perchè i raggi prospettici in questo tipo di prospettiva sono tutti paralleli all'asse di proiezione (nel nostro caso l'asse z, e quindi non convergono verso un centro di proiezione (punto di fuga) al finito, bensì verso un punto di fuga all'infinito (la direzione proiettiva dell'asse z, cioè, in coordinate omogenee, [0, 0, 1, 0]). Questo equivale a dire che la distanza d fra il centro di proiezione e l'origine tende ad infinito.

# 6.3. Un'unica matrice per prospettiva centrale e ortogonale

Nelle due trasformazioni prospettiche viste nelle Sezioni precedenti, la matrice della prospettiva centrale  $M_d^{(c)}$  è quella che si applica quando il centro di proiezione è a distanza  $d < \infty$  dal piano di visuale, mentre la matrice della proiezione ortogonale  $M^o$  si applica quando il centro di proiezione è all'infinito. È possibile unificare questi due casi nelle seguente formulazione più generale di una trasformazione prospettica con un unico punto di fuga (al finito o all'infinito).

Sia  $\mathbf{p} = (x, y, z)$  un punto generico da trasformare prospetticamente, sia  $\mathbf{c}$  il centro di proiezione, ubicato in un punto arbitrario dello spazio, e collochiamo il piano di visuale in posizione  $z = z_p$  (questa è la situazione abituale della Computer Graphics, nella quale il piano di visuale è ortogonale all'asse z). Consideriamo il punto  $\mathbf{b}_p = (0, 0, z_p)$ dove questo piano interseca l'asse z, e sia  $q = \|\mathbf{c} - \mathbf{b}_p\|$  e **d** il versore  $\mathbf{d} = \frac{1}{q}(\mathbf{c} - \mathbf{b}_p)$ . Come prima, denotiamo con  $\mathbf{p}' := \mathbf{p}_p = (x_p, y_p, z_p)$  il punto sul piano di visuale ottenuto proiettando prospetticamente (con centro in **c**) il generico punto  $\mathbf{p} = (x, y, z)$ . Allora  $\mathbf{p}_p$  appartiene alla retta di equazione parametrica

$$\mathbf{r}(t) = (x'(t), y'(t), z'(t)) = \mathbf{c} + t(\mathbf{p} - \mathbf{c})$$
 (6.3.1)

per qualche  $0 \leq t \leq 1$ .

Usando il fatto che  $\mathbf{c} = \mathbf{b}_p + q\mathbf{d}$ ricaviamo da (6.3.1) che i punti del segmento parametrico verificano

$$x'(t) = qd_x + (x - qd_z)t$$
  

$$y'(t) = qd_x + (y - qd_z)t$$
  

$$z'(t) = z_p + qd_z + (z - (z_p + qd_z))t$$

Ora ricaviamo  $\mathbf{p}_p$  ponendo  $z' = z_p$  nell'ultima uguaglianza. In tal modo si ottiene il valore appropriato di t:

$$t = \frac{z_p - (z_p + qd_z)}{z - (z_p + qd_z)},$$



FIGURA 4. La proiezione prospettica: caso generale di punto di fuga al finito

che sostituito nelle uguaglianze precedenti dà

$$\begin{aligned} x_p &= \frac{x - z \frac{d_x}{d_z} + z_p \frac{d_x}{d_z}}{1 + \frac{z_p - z}{qd_z}} \\ y_p &= \frac{y - z \frac{d_y}{d_z} + z_p \frac{d_y}{d_z}}{1 + \frac{z_p - z}{qd_z}} \\ z_p &= z_p \frac{1 + \frac{z_p - z}{qd_z}}{1 + \frac{z_p - z}{qd_z}} = \frac{-z \frac{z_p}{qd_z} + \frac{z_p (1 + qd_z)}{qd_z}}{1 + \frac{z_p - z}{qd_z}} \end{aligned}$$

Queste uguaglianze equivalgono alla forma seguente della trasformazione prospettica  $M \in PM_4(\mathbb{R})$ :

$$\begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} = M \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix},$$

 $\operatorname{con}$ 

$$M = \begin{pmatrix} 1 & 0 & -\frac{d_x}{d_z} & z_p \frac{d_x}{d_z} \\ 0 & 1 & -\frac{d_y}{d_z} & z_p \frac{d_y}{d_z} \\ 0 & 0 & -\frac{z_p}{qd_z} & z_p + \frac{z_p^2}{qd_z} \\ 0 & 0 & -\frac{1}{qd_z} & 1 + \frac{z_p}{qd_z} \end{pmatrix}$$

Si osservi che, scegliendo  $\mathbf{d} = (0, 0, -1)$ , otteniamo

- $M = M_d^{(c)}$  per  $z_p = d$  e q = d (Sezione 6.4),  $M = M_d^{(c)}$  per  $z_p = 0$  e q = d (Sezione 6.4),  $M = M^{(o)}$  per  $z_p = 0$  e  $q = \infty$  (Sezione 6.2),.

#### 6.4. Forma matriciale generale della prospettiva centrale

Nella precedente Sezione 6.3 abbiamo visto la forma della matrice prospettica quando il centro di prospettiva è arbitrario ma, come di consuetudine in Computer Graphics, il piano di visuale è ortogonale all'asse z. Anche se questa è la consuetudine, in una animazione nella quale l'osservatore si sposta può essere necessario spostare anche il piano di visuale al fine di vedere parti interessanti della scena. Si pensi ad esempio ad un videogioco nel quale l'osservatore (ossia il centro di proiezione) si muove in un labirinto, o anche in un appartamento dove attraversa una porta, ed ha bisogno di guardare a sinistra e a destra per vedere se ci sono nemici: mentre attravesra la porta, i suoi lati sinistro e destro giacciono nel piano di visuale o nelle sue immediate vicinanze, e quindi (Esempio 6.1.3) non sono visibili (sono mandati all'infinito dalla proiezione) oppure sono estremamente distorti. È come se l'osservatore guardasse con la coda dell'occhio: per vedere bene deve gitrare la testa, o, nel nostro caso, il piano di visuale. In questa Sezione deriviamo la forma generale della trasformazione prospettica centrale nella quale anche il piano di visuale è scelto arbitrariamente.

Cominciamo con un caso particolare che viola il nostro precedente proposito (legato al confronto con la prospettiva ortogonale) di collocare il centro di prospettiva in (0, 0, -d), ed invece lo ricolloca nell'origine: ma siccome alla fine dovremo traslarlo ad un punto arbitrario, questa scelta ormai non comporta svantaggi, ed anzi rende più semplice effettuare in seguito la traslazione.

**PROPOSIZIONE 6.4.1.** (Prospettiva centrale con centro nell'origine e piano di proiezione arbitrario.) La trasformazione prospettica con centro nell'origine e piano di visuale P a distanza  $d_0$  dall'origine e conversore normale  $\mathbf{n} = (n_x, n_y, n_z)$  è espressa, in coordinate proiettive

146

(ovvero omogenee), dalla matrice

$$M_{\mathbf{0}} = \begin{pmatrix} d_0 & 0 & 0 & 0 \\ 0 & d_0 & 0 & 0 \\ 0 & 0 & d_0 & 0 \\ n_x & n_y & n_z & 0 \end{pmatrix}$$

DIMOSTRAZIONE. Poiché il centro di proiezione è l'origine, la trasformazione manda il punto generico  $\mathbf{p} = (x, y, z)$  in

$$\mathbf{p}' = \alpha \mathbf{p} \in P \,. \tag{6.4.1}$$

Osserviamo che la distanza del piano P dall'origine è, a parte il segno,  $d_0 = \mathbf{p}' \cdot \mathbf{n}$ . Ora, il fattore di compressione  $\alpha$  è il rapporto fra questa distanza e la distanza fra  $\mathbf{p}$  e l'origine proiettata lungo la direzione normale al piano P: ossia,  $\alpha = d_0/\mathbf{p} \cdot \mathbf{n} = d_0/(n_x x + n_y y + n_z z)$ . La divisione per  $n_x x + n_y y + n_z z$  non è altro che la divisione prospettica per la distanza (in questo caso la distanza dall'origine proiettata lungo la direzione normale al piano P) se, in coordinate omogenee, si scrive la matrice  $M_0$  come nell'enunciato.  $\Box$ 

Ora trattiamo il caso generale in cui anche il centro di prospettiva è arbitrario. A questo scopo è utile la seguente Nota, che mette in rilievo la mancanza di linearità delle trasformazioni proiettive pensate come trasformazioni dello spazio tridimensionale visto in coordinate omogenee quadridimensionali.

NOTA 6.4.2. Siano  $\mathbf{p} = (p_x, p_y, p_z) \in \mathbb{R}^3$ ,  $\mathbf{q} = (q_x, q_y, q_z)$ ,  $\mathbf{r} = (r_x, r_y, r_z)$ e si consideri la classe di equivalenza proiettiva  $\overline{\mathbf{p}}$  di  $(p_x, p_y, p_z, 1)$ , che d'ora in poi chiameremo la *classe di equivalenza di*  $\mathbf{p}$ . Se due matrici  $M_1 \in M_2$  a quattro dimensioni mandano, rispettivamente  $(p_x, p_y, p_z, 1)$ in  $(q_x, q_y, q_z, a) \in (r_x, r_y, r_z, b)$ , con  $a \neq -b$ , allora la matrice  $M_1 + M_2$  manda  $(p_x, p_y, p_z, 1)$  in  $(q_x + r_x, q_y + r_y, q_z + r_z, a + b)$ , e quindi le corrispondenti trasformazioni proiettive mandano la classe  $\overline{\mathbf{p}}$  nella classe di  $(\mathbf{q} + \mathbf{r})/(a + b)$ . In particolare, se le due matrici  $M_1 \in M_2$ hanno la stessa ultima riga, e quindi a = b, allora la matrice proiettiva la cui azione sulle classi di equivalenza manda la classe di  $(p_x, p_y, p_z, 1)$ nella somma delle classi di  $(q_x, q_y, q_z, 1) \in (r_x, r_y, r_z, 1)$  è la matrice con la stessa ultima riga di  $M_1 \in M_2$ .  $\Box$ 

NOTA 6.4.3. (Il significato geometrico del fattore di distanza.) Abbiamo ripetutamente incontrato, ad esempio nella Proposizione 6.4.1, il termine  $d_0 = \mathbf{p}' \cdot \mathbf{n}$ , dove  $\mathbf{p}'$  è un generico punto del piano di proiezione *P*. Chiaramente,  $|d_0|$  è la distanza fra *P* e l'origine, ed il segno di  $d_0$  è positivo se il versore normale **n** di P punta nel semispazio opposto all'origine. Nella suddetta Proposizione, l'origine è il centro di proiezione, e quindi la consuetudine della Computer Graphics è che la scena giaccia nel semispazio opposto, per evitare drastiche distorsioni prospettiche, come osservato alla fine dell'Esempio 6.1.3. Quindi in questo caso  $d_0$  è positivo e misura la distanza dall'origine al piano. Nei prossimi enunciati sposteremo il centro di proiezione dall'origine ad un punto arbitrario **c** ed introdurremo la costante  $d_1 = (\mathbf{c}, \mathbf{n})$ . Chiaramente,  $d_1$  è la distanza fra i piani paralleli a P che passano per l'origine e per **c**. Chiamiamo questi due piani paralleli  $P_0$  e  $P_c$ , rispettivamente. Il segno di  $d_1$  è positivo se il centro di proiezione **c** giace in quello dei due semispazi determinati da  $P_0$  verso cui punta il versore normale **n** (qualche volta si parla del *semispazio positivo* determinato dalla scelta del versore normale).

Infine, introdurremo la quantità  $d = d_0 - d_1$ , che ha un ruolo significativo nella matrice della prospettiva centrale nel caso generale. Il significato geometrico di questa costante è comprensibile se consideriamo due casi. Il primo caso è quello in cui il piano P ed il centro cgiacciono in semispazi opposti rispetto a  $P_0$ . In tal caso i segni di  $d_0$ e  $d_1$  sono opposti, e quindi d, a parte il segno, misura la somma delle distanze fra  $P \in P_0$  e fra  $P_0 \in P_c$ , ossia la distanza fra P ed il centro di proiezione **c** (esattamente come succedeva per  $d_0$  nel caso della Proposizione 6.4.1). Nel secondo caso, i segni di  $d_0$  e  $d_1$  sono uguali, e quindi d misura la differenza delle suddette distanze. Però ora il piano  $P_c$  giace fra  $P \in P_0$ , e quindi d misura nuovamente, a parte il segno, la distanza fra P ed il centro di proiezione c. Il segno di d è positivo o negativo a seconda della scelta di verso del versore  $\mathbf{n}$ , e più precisamente è positivo se il centro di proiezione (ossia l'osservatore) sta nel semispazio negativo deteminato dal piano di visuale P. 

TEOREMA 6.4.4. (Prospettiva centrale con centro e piano di proiezione arbitrari.) Consideriamo la trasformazione prospettica con centro in un punto  $\mathbf{c} = (c_x, c_y, c_z)$  e piano di visuale P con versore normale  $\mathbf{n} = (n_x, n_y, n_z)$  a distanza con segno  $d_0$  dall'origine (ossia  $d_0 = \mathbf{p_0} \cdot \mathbf{n}$  per qualsiasi punto  $\mathbf{p_0} \in P$ ). Sia  $d_1$  la distanza con segno fra i piani paralleli a P che passano per l'origine e per  $\mathbf{c}$ , rispettivamente (ossia  $d_1 = \mathbf{c} \cdot \mathbf{n}$ ), e sia  $d = d_0 - d_1$ . Allora la trasformazione è espressa, in coordinate proiettive, dalla matrice  $M_{\mathbf{c}}$ , che in questo  $Teorema \ scriviamo \ brevemente \ M, \ data \ da$ 

$$M = M_{\mathbf{c}} = \begin{pmatrix} d + c_x n_x & c_x n_y & c_x n_z & -c_x d_0 \\ c_y n_x & d + c_y n_y & c_y n_z & -c_y d_0 \\ c_z n_x & c_z n_y & d + c_z n_z & -c_z d_0 \\ n_x & n_y & n_z & -d_x \end{pmatrix} .$$

DIMOSTRAZIONE. Potremmo svolgere la dimostrazione riportando il centro di proiezione all'origine mediante una traslazione, applicando allora la trasformazione prospettica la cui matrice è stata determinata nella Proposizione 6.4.1 ed infine ritornando indietro con la traslazione opposta. Questi calcoli li svolgeremo nel successivo Esercizio 6.4.5 qui invece preferiamo sviluppare il calcolo diretto in analogia alla dimostrazione della Proposizione 6.4.1.

Poiché ora la trasformazione prospettica proietta radialmente verso **c** il punto generico  $\mathbf{p} = (x, y, z)$  in un punto  $\mathbf{p}'$ , l'identità (6.4.1) diventa

$$\mathbf{p}' - \mathbf{c} = \alpha(\mathbf{p} - \mathbf{c}) \,. \tag{6.4.2}$$

Pertanto  $\mathbf{n} \cdot (\mathbf{p}' - \mathbf{c}) = \alpha \mathbf{n} \cdot (\mathbf{p} - \mathbf{c})$ , da cui

$$\alpha = \frac{\mathbf{n} \cdot \mathbf{p}' - \mathbf{n} \cdot \mathbf{c}}{\mathbf{n} \cdot \mathbf{p} - \mathbf{n} \cdot \mathbf{c}} = \frac{d_0 - d_1}{\mathbf{n} \cdot \mathbf{p} - d_1} = \frac{d}{\mathbf{n} \cdot \mathbf{p} - d_1}, \quad (6.4.3)$$

perché, per definizione,  $d_0 = \mathbf{n} \cdot \mathbf{p}'$ ,  $d_1 = \mathbf{n} \cdot \mathbf{c}$  e  $d = d_0 - d_1$ . Riscriviamo la trasformazione prospettica (6.4.2) come

$$\mathbf{p}' = T\mathbf{p} = \mathbf{c} + \alpha(\mathbf{p} - \mathbf{c}) = \alpha\mathbf{p} + (1 - \alpha)\mathbf{c}$$

ovvero  $T = T_1 + T_2$ , dove  $T_1\mathbf{p} = \alpha\mathbf{p}$  e  $T_2\mathbf{p} = (1 - \alpha)\mathbf{c}$ . Consideriamo le trasformazioni proiettive  $\widetilde{T}_1$  e  $\widetilde{T}_2$  indotte da  $T_1$  e  $T_2$  pensate in coordinate omogenee. Scriviamo le matrici (a dimensione 4)  $M_1$  e  $M_2$ associate alle trasformazioni proiettive  $\widetilde{T}_1$  e  $\widetilde{T}_2$ . È chiaro da (6.4.3) che la prima di esse è

$$M_1 = \begin{pmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & d & 0 \\ n_x & n_y & n_z & -d_1 \end{pmatrix}$$

Per scrivere  $M_2$  osserviamo che da (6.4.3) segue

$$1 - \alpha = 1 - \frac{d}{\mathbf{n} \cdot \mathbf{p} - d_1} = \frac{\mathbf{n} \cdot \mathbf{p} - d_0}{\mathbf{n} \cdot \mathbf{p} - d_1}.$$

Pertanto 
$$\widetilde{T}_2$$
 diventa  
 $\widetilde{T}_2(x, y, z, 1) = ((1 - \alpha)c_x, (1 - \alpha)c_y, (1 - \alpha)c_z, 1)$   
 $\sim ((\mathbf{n} \cdot \mathbf{p} - d_0)c_x, (\mathbf{n} \cdot \mathbf{p} - d_0)c_y, (\mathbf{n} \cdot \mathbf{p} - d_0)c_z, \mathbf{n} \cdot \mathbf{p} - d_1)$   
 $= ((n_x x + n_y y + n_z z)c_x - d_0c_x, (n_x x + n_y y + n_z z)c_y - d_0c_y, (n_x x + n_y y + n_z z)c_z - d_0c_z, n_x x + n_y y + n_z z - d_1),$ 

e quindi

$$M_{2} = \begin{pmatrix} c_{x}n_{x} & c_{x}n_{y} & c_{x}n_{z} & -c_{x}d_{0} \\ c_{y}n_{x} & c_{y}n_{y} & c_{y}n_{z} & -c_{y}d_{0} \\ c_{z}n_{x} & c_{z}n_{y} & c_{z}n_{z} & -c_{z}d_{0} \\ n_{x} & n_{y} & n_{z} & -d_{1} \end{pmatrix}$$

L'enunciato ora segue dalla Nota 6.4.2.

ESERCIZIO 6.4.5. Come annunciato all'inizio della dimostrazione del Teorema 6.4.4, ora diamo una formulazione alternativa di quella dimostrazione riportando all'origine il centro di proiezione **c** mediante la traslazione  $\Lambda(x, y, z) = (x - c_x, y - c_y, z - c_z)$ , poi applicando la trasformazione prospettica con centro l'origine della Proposizione 6.4.1 ed infine ritornando indietro con la traslazione opposta  $\Lambda^{-1}$ .

SVOLGIMENTO. La matrice della trasformazione affine associata alla traslazione  $\Lambda$  è

$$\Lambda = \begin{pmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 1 & -c_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

e la sua inversa ovviamente è la traslazione opposta,

$$\Lambda^{-1} = \begin{pmatrix} 1 & 0 & 0 & c_x \\ 0 & 1 & 0 & c_y \\ 0 & 0 & 1 & c_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

.

Per scrivere la matrice  $M_0$  abbiamo bisogno della distanza con segno fra il piano di proiezione e l'origine *dopo aver applicato la traslazione*  $\Lambda$ . Nella Proposizione 6.4.1 questa distanza, chiaramente, era  $d_0 = \mathbf{n} \cdot \mathbf{p}'$ , dove  $\mathbf{p}'$  è un punto qualsiasi del piano di proiezione. La traslazione manda  $\mathbf{p}'$  in  $\mathbf{p}' - \mathbf{c}$ , e quindi  $d_0$  in

$$\mathbf{n} \cdot \mathbf{p}' - \mathbf{c} = \mathbf{n} \cdot \mathbf{p}' - \mathbf{n} \cdot \mathbf{c} = d_0 - d_1 = d_0$$

150

con la notazione del precedente Teorema 6.4.4. Pertanto, la matrice della Proposizione 6.4.1 ora diventa

$$M_{\mathbf{0}} = \begin{pmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & d & 0 \\ n_x & n_y & n_z & 0 \end{pmatrix}$$

È ora immediato verificare che la composizione di matrici  $\Lambda^{-1} M_0 \Lambda$  (si badi all'ordine!) è esattamente la matrice  $M_c$  del Teorema 6.4.4.

# 6.5. Punti di fuga della prospettiva centrale

Abbiamo introdotto il concetto di punto di fuga per la proiezione standard nella Nota 6.1.2. Rammentiamo che si tratta del punto in cui si incontrano le immagini sotto la trasformazione prospettica di un fascio di rette parallele, in quel caso ortogonali al piano di visuale, e quindi orientate secondo il versore normale dell'azze z. Ora, grazie al Teorema 6.4.4, possiamo estendere quel risultato al caso della prospettiva centrale generale, per un fascio di rette parallele diretto secondo un versore arbitrario  $\mathbf{u} = (u_x, u_y, u_z)$ , non necessariamente ortogonale al piano di visuale P. Come prima, sia  $\mathbf{c} = (c_x, c_y, c_z)$  il centro di proiezione, e  $\mathbf{n}$  il versore normale a P. Per qualsiasi punto  $\mathbf{p_0} = (x_0, y_0, z_0) \in P$ , la distanza (con segno) di P dall'origine è  $d_0 = \mathbf{n} \cdot \mathbf{p_0}$ , e la distanza (con segno) fra i piani paralleli a P che passano, rispettivamente, per l'origine e per  $\mathbf{c}$  è  $d_1 = \mathbf{n} \cdot \mathbf{c}$ . Poniamo ancora  $d = d_0 - d_1$ .

TEOREMA 6.5.1. Chiamiamo  $\mathbf{r}$  le rette parallele dirette secondo il versore  $\mathbf{u}: \mathbf{r}(t) = \mathbf{r}^{\mathbf{0}} + t\mathbf{u}$ , con  $\mathbf{r}_{\mathbf{0}}$  arbitrario. Se il fascio non è diretto parallelamente al piano di visuale, ossia se  $\langle \mathbf{n}, \mathbf{u} \rangle \neq 0$ , allora la proiezione della prospettiva centrale generale del Teorema 6.4.4 manda questo fascio di rette in un insieme di rette  $\mathbf{s}$  radiali rispetto al (ossia che si incontrano nel) punto di fuga

$$\mathbf{f}_{\mathbf{u}} = \mathbf{c} + \frac{d}{\langle \mathbf{n}, \mathbf{u} \rangle} \mathbf{u} \,.$$

Altrimenti, se  $\langle \mathbf{n}, \mathbf{u} \rangle = 0$ , il fascio di rette rimane parallelo anche dopo la trasformazione prospettica, e non c'è alcun punto di incontro (per queste direzioni diciamo che il punto di fuga è all'infinito).

DIMOSTRAZIONE. Scriviamo  $\mathbf{r}(t) = (x(t), y(t), z(t))$ , e la retta immagine  $\mathbf{s}(t) = (x'(t), y'(t), z'(t))$ . Sia T la trasformazione della prospettiva centrale con piano di proiezione P e centro di proiezione **c**: essa agisce in coordinate omogenee tramite la matrice  $M_{\mathbf{c}}$  del Teorema 6.4.4. Scriviamo rispettivamente  $\mathbf{R}(t)$ ,  $\mathbf{S}(t)$  le classi di equivalenza proiettiva  $[\mathbf{r}(t)]$ ,  $[\mathbf{s}(t)]$ , ossia in coordinate omogenee. Allora la trasformazione  $T\mathbf{r} = \mathbf{s}$  diventa  $M_{\mathbf{c}}\mathbf{R}(t) = \mathbf{S}(t)$ . Scegliamo i rappresentanti speciali della classe di equivalenza omogenea  $\mathbf{R}(t)$  dati da (x(t), y(t), z(t), 1). L'azione di  $M_{\mathbf{c}}$  manda questo vettore in un rappresentante di  $\mathbf{S}$  che scriviamo (x'', y'', z'', g). Dal Teorema 6.4.4 ora abbiamo

$$\begin{aligned} x''(t) &= (d + c_x n_x)(r_x^0 + t u_x) + c_x n_y (r_y^0 + t u_y) + c_x n_z (r_z^0 + t u_z) - c_x d_0 \\ y''(t) &= c_y n_x (r_x^0 + t u_x) + (d + c_y n_y) (r_y^0 + t u_y) + c_y n_z (r_z^0 + t u_z) - c_y d_0 \\ z''(t) &= c_z n_x (r_x^0 + t u_x) + c_z n_y (r_y^0 + t u_y) + (d + c_z n_z) (r_z^0 + t u_z) - c_z d_0 \\ g &= n_x (r_x^0 + t u_x) + n_y (r_y^0 + t u_y) + n_z (r_z^0 + t u_z) - d_1 . \end{aligned}$$

Da qui passiamo ai valori tridimensionali delle componenti di  $\mathbf{s}(t)$  dividendo x'', y'', z'' per g e poi facciamo tendere t ad infinito per trovare il punto di fuga  $\mathbf{f}_{\mathbf{u}}$ . Il limite esiste se il limite del denominatore non si annulla. Osserviamo che, senza perdita di generalità, si può assumere che il punto  $\mathbf{r}^{\mathbf{0}}$  appartenga al piano P: infatti, se il fascio di rette non è parallelo a P, ossia se  $\langle \mathbf{n}, \mathbf{u} \rangle \neq 0$ , allora ogni retta del fascio ha un (unico) punto di intersezione con P e scegliamo  $\mathbf{r}^{\mathbf{0}}$  come questo punto di intersezione; se invece il fascio di rette è parallelo a P, possiamo scegliere per  $\mathbf{r}(t)$  una retta che giace in P, ed allora ogni suo punto  $\mathbf{r}^{\mathbf{0}}$ sta in P. Con questa scelta di  $\mathbf{r}^{\mathbf{0}}$  sappiamo che  $\langle \mathbf{n}, \mathbf{r}^{\mathbf{0}} \rangle$  è la distanza  $d_1$  di P dall'origine. Ora vediamo dall'espressione di g che il limite per  $t \to \infty$  è zero se e solo se  $\langle \mathbf{n}, \mathbf{u} \rangle \neq 0$ . In tal caso il limite è esattamente l'espressione dell'enunciato. Se invece  $\langle \mathbf{n}, \mathbf{u} \rangle = 0$  il termine in t viene moltiplicato per zero e scompare, quindi le rette trasformate non si incontrano per alcun valore di t.  $\square$ 

NOTA 6.5.2. La determinazione del punto di fuga di un fascio di rette in direzione  $\mathbf{u}$ , ricavata nel Teorema 6.5.1, si può ottenere senza calcoli con un argomento geometrico alternativo ed illuminante, che ora presentiamo.

Per prima cosa si osserva che il punto di fuga deve esistere al finito se il fascio non è parallelo al piano di proiezione, perchè in tal caso la proiezione centrale diminuisce le distanze di punti che si muovono parallelamente allontanandosi dal piano di proiezione, ossia restando a distanza costante prima della proiezione (questo fatto geometricamente evidente equivale alla divisione prospettica per la distanza).

In secondo luogo, l'immagine della proiezione prospettica centrale è il piano di proiezione P. In particolare, il punto di fuga del fascio di rette parallele a **u** è un punto di P. In questo fascio basta allora considerare

la retta che passa per il centro di proiezione  $\mathbf{c}$ , ossia  $\mathbf{r}(t) = \mathbf{c} + t\mathbf{u}$ : il punto di fuga del fascio è l'intersezione di questa retta con P (esso esiste se e solo se  $\mathbf{u}$  non è un vettore parallelo a P: se invece lo è, allora le rette del fascio sono tutte parallele al piano P, e poiché  $\mathbf{c}$  si sceglie fuori di questo piano la retta  $\mathbf{s}$  non lo interseca: in tal caso si dice che il punto di fuga è all'infinito, come abbiamo visto nella dimostrazione del precedente Teorema 6.5.1 e vedremo geometricamente nella successiva Nota 6.5.3).

Se il fascio non è parallelo a P, mostriamo in dettaglio ciò che abbiamo già accennato alla fine della dimostrazione del precedente Teorema 6.5.2, ossia che il punto  $\mathbf{f_u}$  giace in P. Sia come sempre d la distanza fra  $\mathbf{c} \in P$ . Poiché  $\mathbf{u}$  ha norma 1, il valore di t per il quale la retta  $\mathbf{s}(t) = \mathbf{c} + t\mathbf{u}$  interseca P è la lunghezza  $d_{\mathbf{u}}$  del segmento da  $\mathbf{c}$  a Pnella direzione di  $\mathbf{u}$ . La lunghezza della proiezione di questa segmento sulla direzione normale a P (quella del versore  $\mathbf{n}$ ) è quindi d: pertanto  $d_{\mathbf{u}} = d_1/\mathbf{n} \cdot \mathbf{u} = d_1/\cos\theta$ , dove  $\theta$  è l'angolo sotteso da  $\mathbf{u} \in \mathbf{n}$ . In altre parole, il punto di fuga è precisamente  $\mathbf{f_u} = \mathbf{c} + \mathbf{u} d/\cos\theta$ , come trovato con i calcoli della dimostrazione del Teorema 6.5.1, e giace in P.

NOTA 6.5.3. È interessante vedere come cambia la posizione del punto di fuga al variare del versore **u** del fascio nella sfera unitaria. Se **u** varia nella sfera mantenendo costante la propria deviazione angolare  $\theta$  da **n** (ossia se si muove sul parallelo  $C_{\theta}$  rispetto al polo **n**), allora il punto di fuga descrive anch'esso una circonferenza nello spazio, ed esattamente il parallelo  $C_{\theta}$  della sfera con centro **c** e raggio  $d/\cos\theta$ , che tende ad infinito quando  $\theta$  tende a  $\pi/2$ . Questo è vero fin tanto che  $\theta \neq \pi/2$ , ossia purché **u** non sia perpendicolare a **n**, ovvero sull'equatore. Il caso di **u** sull'equatore è il caso limite in cui il raggio di queste circonferenze diverge, e diciamo allora che i punti di fuga sono all'infinito: in tal caso le rette rimangono parallele anche dopo la trasformazione prospettica.

Quindi punti di fuga a distanza piccola dal centro di proiezione si ottengono quando il fascio si avvicina ad essere perpendicolare al piano di proiezione (ossia l'angolo  $\theta$  vicino a zero, **u** vicino ad essere allineato con **n**), ed in aggiunta quando d è piccolo. Per quanto visto nella Nota 6.4.3, l'ultima condizione significa che la distanza fra il centro di proiezione (ovvero l'osservatore) ed il piano di proiezione P è piccola, ossia la prospettiva è accentuata. In altre parole, punti di fuga ravvicinati corrispondono a un osservatore vicino alla scena, con l'accentuazione (o distorsione) prospettica tipica di ciò che si ottiene quando si fotografa una scena da vicino. In questo caso, perché si riesca a riprendere la scena nella sua interezza, bisogna usare un grandangolo, e questa distorsione prospettica si chiama appunto distorsione grandangolare.  $\hfill \Box$ 

Dal Teorema 6.5.1 ottieniamo immediatamente il risultato seguente. Si noti che la terza identità è coerente con quella trovata nella Nota 6.1.2.

COROLLARIO 6.5.4. Scriviamo  $\mathbf{f}_i$  invece che  $\mathbf{f}_{\mathbf{e}_i}$  (i=1,2,3) per i punti di fuga principali della prospettiva centrale, ossia i punti di fuga dei fasci di rette paralleli, rispettivamente, ai tre versori canonici di base. Essi sono, rispettivamente,

$$\mathbf{f_1} = (c_x + \frac{d}{n_x}, c_y, c_z) \tag{6.5.1a}$$

$$\mathbf{f_2} = (c_x, c_y + \frac{d}{n_y}, c_z)$$
 (6.5.1b)

$$\mathbf{f_3} = (c_x, c_y, c_z + \frac{d}{n_z}).$$
 (6.5.1c)

In queste identità, si noti che, per ogni coppia di versori canonici, i corrispondenti punti di fuga principali hanno componente lungo il restante versore uguale a quella del centro di proiezione: ad esempio, i punti di fuga principali nelle direzioni degli assi  $x \, e \, y$  hanno la stessa altezza sul piano  $\{z = 0\}$  del centro di proiezione, e così via ruotando gli indici.

ESERCIZIO 6.5.5. Mostrare che i punti di fuga principali, se sono punti al finito, sono esattamente le intersezioni con il piano di proiezione delle tre semirette uscenti dal centro di proiezione  $\mathbf{c}$  nelle direzioni dei tre versori canonici di base.

ESERCIZIO 6.5.6. (i) Un pittore sceglie la prospettiva per un quadro nel quale, oltre ad altri dettagli, dipinge due edifici a forma di parallelepipedo. Il primo edificio è disposto in modo frontale, diciamo come il cubo di lato 1 nel primo ottante che contiene (0,0,3) ed ha la faccia più vicina al piano di proiezione parallela ad esso e disposta sul piano  $\{z = 3\}$ , e le altre due facce che contengono (0,0,3) disposte rispettivamente sui piani  $\{x = 0\}$ e  $\{y = 0\}$ . Per dipingere questo edificio il pittore sceglie la distanza d del proprio punto di osservazione dal piano della tela (supponiamo che il piano sia  $\{z = 2\}$  e sia d = 1), ed un punto di fuga per i lati del cubo diretti parallelamente all'asse z (ossia ortogonalmente alla tela). Supponiamo che come punto di fuga scelga  $\mathbf{f_3} = (1, 1, 2)$ .

154

L'altro edificio è anch'esso a forma cubica, con angolo di  $\pi/4$  (45 gradi) rispetto alla direzione di osservazione: ad esempio il cubo di altezza  $\sqrt{2}$  sopra il quadrato sul piano {y = 0} di vertici (x = 10, z = 3), (11, 4), (10, 5), (9, 4). Il pittore come disegna in maniera prospetticamente corretta i lati di questo secondo edificio?

*(ii)* La stessa scena dipinta dal pittore viene fotografata con una macchina fotografica disposta dove era il pittore, con la stessa angolazione (quindi frontalmente rispetto al primo edificio, ossia con le stesse angolazioni orizzontale e verticale). Quella che era la distanza del pittore dalla tela ora diventa la distanza fra il punto di messa a fuoco ed il piano di proiezione, ossia il piano della pellicola (o del sensore, in fotografia digitale). (Nota: nella macchina fotografica i raggi di luce convergono verso il punto di messa a fuoco e colpiscono il sensore (che di solito è situato oltre tale punto, ma questo non è troppo rilevante per il nostro modello matematico: l'unica differenza è che si ottiene una immagine rovesciata). Però questi raggi non provengono in linea retta dalla scena, perché nell'attraversare la lente (o le lenti) dell'obiettivo subiscono una deviazione, che dipende dalla lunghezza focale: questa deviazione è misurata dall'angolo di campo dell'obiettivo (l'angolo di campo è grande nel caso di un grandangolo e piccolo nel caso di un teleobiettivo: per obiettivi n di lunghezza focale di circa 50 mm l'angolo di campo coincide approssimativamente con quello del nostro sistema visivo, a meno di dilatazione di scala per l'adeguamento del fotogramma al formato standard 35 mm). Qui ignoriamo questa deviazione, che comunque si traduce in nient'altro che un ingrandimeno o riduzione della grandezza della scena, o se si preferisce un cambiamento di scala dell'immagine. Si badi però che ignorare questa dilatazione di scala corrisponde a scegliere una lunghezza focale effettiva diversa da quella reale). Poiché il piano del sensore è abbastanza vicino al piano centrale dell'obiettivo, con buona approssimazione possiamo rimpiazzare questa distanza d con la lunghezza focale dell'obiettivo, che, quando guardiamo la fotografia, non ci è nota. Come possiamo ricostruire dalla fotografia l'angolazione nello spazio tridimensionale del secondo edificio?

SVOLGIMENTO. (i). Sappiamo che  $\mathbf{n} = (0, 0, 1)$ , quindi  $n_z = 1$ . Dall'equazione (6.5.1c) del punto di fuga principale  $\mathbf{f_3}$  ricaviamo  $c_z + d = 2$ , quindi  $c_z = 1$ , ed anche le altre coordinate del centro di proiezione valgono 1. In altre parole  $\mathbf{c} = (1, 1, 1)$ . Le due facciate visibili del secondo edificio sono dirette come  $\mathbf{u}^- = (-1/\sqrt{2}, 0, -1/\sqrt{2})$  e  $\mathbf{u}^+ = (1/\sqrt{2}, 0, -1/\sqrt{2})$ . I punti di fuga per questi versori si ottengono dal Teorema 6.5.1:

$$\mathbf{f}_{-} = (1, 1, 1) - \sqrt{2}(-1/\sqrt{2}, 0, -1/\sqrt{2}) = (2, 1, 2)$$
  
$$\mathbf{f}_{+} = (1, 1, 1) - \sqrt{2}(1/\sqrt{2}, 0, -1/\sqrt{2}) = (0, 1, 2).$$

Il pittore disegna quindi i bordi del secondo edificio facendo convergere le linee parallele ai rispettivi punti di fuga appena trovati.

(*ii*). Dalla fotografia misuriamo le coordinate del punto di fuga dei lati del primo edificio ortogonali al piano della pellicola, che abbiamo chiamato  $f_3$ . Grazie all'equazione (6.5.1c) questa misura ci dà il vettore  $\mathbf{c} + d\mathbf{e_3}$ , ossia le prime due coordinate di  $\mathbf{c}$  ed il valore della quantità  $c_z + d$ , che è la terza componente di  $\mathbf{f_3}$  e vale 2. Ora consideriamo i punti di fuga associati agli spigoli non verticali del secondo edificio (quelli verticali rimangono paralleli al piano della pellicola, perché l'edificio e questo piano sono entrambi verticali, quindi paralleli: il terzo punto di fuga è all'infinito). Vogliamo determinare i versori  $\mathbf{u}_{\pm}$  delle due famiglie ortogonali di tali spigoli. Osserviamo che  $\mathbf{n} = \mathbf{e_3}$ , e pertanto  $\mathbf{u}^{\pm} \cdot \mathbf{n} =$  $u_z^{\pm}$ . Ora dalle equazioni (6.5.1a) e (6.5.1c) ricaviamo i vettori  $\mathbf{c} + \frac{d}{u^{\pm}} \mathbf{u}^{\pm}$ . Poiché le prime due componenti di  $\mathbf{c}$  sono già note, questo ci restituisce i valori di  $du_x^{\pm}/u_z^{\pm}$  e  $du_y^{\pm}/u_z^{\pm}$  (ma attenzione: sappiamo già che  $u_y^{\pm} = 0$ , perché l'edificio è verticale, non inclinato, quindi l'ultima informazione non ci serve). Per le terze componenti, troviamo in entrambi i casi il valore di  $c_z + d$ , ma sapevamo già che questo valore è 2, quindi anche questa informazione non ci dà nulla di nuovo. Infine sappiamo che  $\mathbf{u}^+$  e  $\mathbf{u}^-$  sono ortogonali e di norma 1. Ricapitolando tutte queste informazioni nell'ordine dato, abbiamo:

$$c_{z} + d = 2$$
  

$$d \frac{u_{x}^{+}}{u_{z}^{+}} = \text{costante nota}$$
  

$$d \frac{u_{x}^{-}}{u_{z}^{-}} = \text{costante nota}$$
  

$$u_{x}^{+}u_{x}^{-} + u_{z}^{+}u_{z}^{-} = 0$$
  

$$(u_{x}^{+})^{2} + (u_{z}^{+})^{2} = 1$$
  

$$(u_{x}^{-})^{2} + (u_{z}^{-})^{2} = 1.$$

156

Si tratta di sei equazioni quadratiche nelle sei incognite d,  $c_z$ ,  $u_x^{\pm}$ ,  $u_z^{\pm}$ (si rammenti che  $u_y^{\pm} = 0$ ). Se il sistema è risolvibile, in tal modo si determina d ed i due versori  $\mathbf{u}^{\pm}$  richiesti. Lasciamo al lettore considerare le condizioni di risolubilità e sviluppare i calcoli numerici dopo aver fissato a proprio piacere le posizioni dei tre punti di fuga utilizzati (ma ci si rammenti che questi tre punti debbono essere scelti tutti alla stessa altezza y, perchè giacciono sulla retta che rappresenta l'orizzonte sul piano della pellicola, e questa retta è orizzontale perché la macchina fotografica non è inclinata di lato).

Si noti anche che ora conosciamo anche  $\mathbf{c}$  e d, e quindi l'intera matrice della trasformazione prospettica. Il lettore è invitato a scriverla esplicitamente.

ESERCIZIO 6.5.7. Per accentuare la drammaticità della fotografia, un fotografo vuol riprendere un grattacielo in modo che i due spigoli adiacenti del tetto visibili dalla sua posizione si dispongano parallelamente a due lati adiacenti del fotogramma (si veda la Figura 5). È possibile? Come si deve orientare la macchina fotografica?

SVOLGIMENTO. Senza perdita di generalità poniamo nell'origine il punto di messa a fuoco dentro la macchina fotografica: in tal modo la trasformazione prospettica corrisponde alla matrice  $M_0$  particolarmente semplice del Teorema 6.4.4, dove d è la distanza fra il centro di proiezione (l'origine) ed il piano di proiezione (il piano del sensore o della pellicola). Dobbiamo trovare se esiste un versore  $\mathbf{n}$  verso cui orientare la macchina fotografica onde ottenere l'inquadratura voluta. Osserviamo che, se tale versore esiste, allora possiamo ruotare l'immagine nel piano di proiezione semplicemente ruotando assialmente (intorno a  $\mathbf{n}$ ) la macchina fotografica in verso opposto: pertanto non importa se i due lati della sommità del grattacielo si proiettano parallelamente ai lati del fotogramma, quello che conta è che si proiettino in maniera perpendicolare. Chiamiamo **p** il punto di congiunzione dei due lati del tetto che il fotografo vede: i due lati sono disposti lungo due rette uscenti da **p** e dirette come due vettori canonici di base, diciamo  $\mathbf{e_1} \in \mathbf{e_3}$  qui adottiamo la consueta terminologia della Computer Graphics e disponiamo il lato della profondità del grattacielo lungo l'asse z: l'asse x rappresenta la direzione, diciamo, quasi frontale rispetto al fotografo, e l'asse y è l'altezza). Una volta proiettate prospetticamente, queste due rette passano per il punto  $\mathbf{p}'$  immagine di  $\mathbf{p}$  sotto la trasformazione prospettica, e sono dirette verso i punti di fuga principali corrispondenti ai versori  $\mathbf{e_1} \in \mathbf{e_3}$ . Diciamo che, a meno di cambiamento di scala, l'edificio abbia larghezza 1 (ovvero estensione 1 lungo l'asse x), profondità w



FIGURA 5. Prospettiva centrale di un edificio, una inquadratura con taglio drammatico (l'immagine è una elaborazione di una fotografia di Michael Freeman, pubblicata in *L'occhio del fotografo*, Logos, 2008, edizione originale The Ilex Press, 2007).

(lungo l'asse z) ed altezza h (lungo l'asse y). Per comodità, ma senza perdita di generalità, disponiamo la base dell'edificio sul rettangolo  $(1,0,1), (1,0,w), (2,0,w) \in (2,0,1)$ : pertanto il punto  $\mathbf{p} \in (1,h,1)$ , ed i due lati visibili del tetto hanno come rispettivi punti terminali  $\mathbf{p_1} := \mathbf{p} + \mathbf{e_1} = (2,h,1) \in \mathbf{p_3} := \mathbf{p} + w\mathbf{e_3} = (1,h,w)$ . In realtà non abbiamo bisogno di utilizzare la individuazione dei punti di fuga bprincipali dato dal Corollario 6.5.4: basta effettuare la trasformazione prospettica (data nel nostro caso dalla matrice particolarmente semplice  $M_0$ ) ai punti  $\mathbf{p_1} \in \mathbf{p_3}$  ed imporre che i loro trasformati  $\mathbf{p'_1} \in \mathbf{p'_3}$  verifichino la relazione di ortogonalità

$$p_1'-p' \quad \bot \quad p_3'-p'\,.$$

Ma è altrettanto semplice, e a questo punto più elegante, utilizzare le formule per i punti di fuga principali del Corollario 6.5.4: la precedente relazione di ortogonalità allora diventa

$$\mathbf{p}_1' - \mathbf{p}' \perp \mathbf{p}_3' - \mathbf{p}' \tag{6.5.2}$$

(osserviamo che non occorre moltiplicare per w il vettore  $\mathbf{f}_{\mathbf{e}_3}$  del punto di fuga (6.5.1c) per w onde ottenere proprio  $\mathbf{p}'_3$ : l'ortogonalità non dipende dalla lunghezza di questo vettore).

Vista la forma della matrice  $M_0$  del Teorema 6.4.4 ed il fatto che  $\mathbf{c} = 0$ , troviamo  $\mathbf{p}' = (d/\langle \mathbf{n}, \mathbf{p} \rangle)\mathbf{p}$ . Allora, in base alle identità (6.5.1a) e (6.5.1c), la relazione di ortogonalità (6.5.2) diventa

$$(d/n_x)\mathbf{e_1} - (d/\langle \mathbf{n}, \mathbf{p} \rangle)\mathbf{p} \perp (d/n_z)\mathbf{e_3} - (d/\langle \mathbf{n}, \mathbf{p} \rangle)\mathbf{p}$$

Vista la scelta di  $\mathbf{p} = (1, h, 1)$ , un facile calcolo diretto permette di riscrivere questa condizione come

$$n_x^2 + n_z^2 - n_x n_z h^2 + (n_x + n_z) n_y h = 0.$$
 (6.5.3)

Rammentando anche la condizione di normalizzazione  $n_x^2 + n_y^2 + n_z^2 = 1$ , abbiamo due equazioni quadratiche, che corrispondono a superficie quadratiche che si intersecano in una curva. Tutti i versori in questa curva risolvono il problema. Ad esempio, una soluzione banale è  $\mathbf{n} = \pm \mathbf{e_2} = (0, \pm 1, 0)$ : in tal caso l macchina fotografica è orientata verticalmente, e se il grattacielo fosse trasparente in effetti si vedrebbero i lati del tetto rimanere paralleli dopo la proiezione, perché sono paralleli al piano di proiezione (ovviamente il gtrattacielo non è trasparente, ma questa soluzione rimane valida se invece che mettere la macchina fotografica nell'origine la mettiamo alta sopra l'edificio in un punto dell'asse y, e la orientiamo in basso, lungo il versore (0, -1, 0). Oltre a questa soluzione banale c'è una famiglia ad un parametro di soluzioni non banali.

La curva delle soluzioni non è degenere, ossia non consiste dei soli punti  $(0, \pm 0)$ , almeno se h è abbastanza grande. Infatti, poniamo  $\mathbf{n} = (\mathbf{t}, \sqrt{1 - 2\mathbf{t}^2}, \mathbf{t})$ , con  $0 < t < 1/\sqrt{2}$ : allora la condizione (6.5.3) diventa

$$0 = 2t^{2} - 4t^{2}h^{2} + 2t\sqrt{1 - 2t^{2}}h = (1 - 2h^{2})t^{2} + t\sqrt{1 - 2t^{2}}h.$$

C'è una soluzione per t = 0; per piccoli t > 0, il primo termine all'ultimo membro cresce quadraticamente ed è negativo se  $h > 1/\sqrt{2}$ , mentre il secondo cresce linearmente ed è positivo perché h > 0, quindi la somma è positiva. Quando t si avvicina a  $1/\sqrt{2}$  la somma si avvicina a  $(1 - 2h^2)/2$ , che bè negativo: quindi c'è una soluzione non banale nell'intervallo  $0 < t < 1/\sqrt{2}$ .

ESEMPIO 6.5.8. La prospettiva centrale con un solo punto di fuga è quella in cui il piano di proiezione è parallelo a due dei versori canonici, ossia perpendicolare al terzo. Ad esempio, se come d'abitudine in Computer Graphics la direzione perpendicolare è quella dell'asse z, si ottiene

la prospettiva del tipo della proiezione standard della Sezione 6.1. Abbiamo già determinato nell'Esempio 6.1.3 come appare il cubo unitario in questa prospettiva: lo disegniamo in Figura 6.



FIGURA 6. Prospettiva centrale con un solo punto di punto di fuga al finito

Il caso di due punti di fuga è illustrato nella Figura 7. La retta a cui appartengono i due punti di fuga si chiama l'orizzonte. Si osservi che, in pittura, è il pittore a decidere a che altezza collocare la retta dell'orizzonte, esattamente come, nel caso di un solo punto di fuga, l'altezza di tale punto nel disegno può essere scelta arbitrariamente. In effetti, la precedente Figura 6 illustra il caso di orizzonte a media altezza (e centro di prospettiva alto, ovvero punto di osservazione dall'alto: la faccia superiore del cubo è visibile, il centro di prospettiva (ossia l'osservatore) è più alto del cubo). In Figura 7 disegniamo il cubo con orizzonte e centro alti, in Figura 8 lo ridisegniamo con orizzonte e centro più bassi (la proiezione sul piano di visuale dell'orizzonte è la linea tratteggiata). Se abbassassimo ancora di più il centro di prospettiva, vedremmo anche la faccia inferiore del cubo, dal di sotto.

160



FIGURA 7. Prospettiva centrale con due punti di fuga, centro di proiezione alto, e quindi orizzonte alto



FIGURA 8. Prospettiva centrale con due punti di fuga e centro di proiezione ad altezza media, e quindi orizzonte ad altezza media

# 6.6. Prospettive parallele

Come sappiamo, le trasformazioni prospettiche sono applicazioni da  $\mathbb{R}^3$  ad un piano di visuale  $V \sim \mathbb{R}^2$ . V è un piano in  $\mathbb{R}^3$  che non passa necessariamente per l'origine, e quindi non costituisce in generale un sottospazio vettoriale.

Si chiamano *parallele* le trasformazioni prospettiche senza punti di fuga, ossia le proiezioni che conservano il parallelismo delle rette. A differenza della prospettiva centrale, le prospettive parallele non comportano una divisione prospettica per la distanza, e quindi si possono descrivere in coordinate usuali tridimensionali, senza bisogno di coordinate omogeneee quadridimensionali. Poiché V in generale non contiene l'origine, anche le prospettive parallele, come già quella centrale, non sono applicazioni lineari (non conservano l'origine!), però almeno sono trasformazioni affini, a differenza della prospettiva centrale. Quindi in generale esse sono rappresentate da matrici affini, ossia a dimensione quattro, tranne che nel caso in cui il piano V passi per l'origine, nel qual caso basta una matrice a dimensione tre (la corrispondente matrice affine avrebbe l'ultima colonna e l'ultima riga uguali a (0,0,0,1), e quindi componente di traslazione nulla: essa si identifica in modo naturale con il suo minore a dimensione tre relativo alle prime tre componenti.

Le proiezioni parallele si suddividono in ortogonali (o ortografiche) quando la direzione di proiezione coincide con il versore normale al piano di proiezione, ed oblique altrimenti. Il caso delle proiezioni ortogonali è assai semplice ed è stato già trattato nelle Sezioni 6.2 e 6.3.

**6.6.1.** Proiezione parallela. In questa sottosezione, e nelle prossime, studiamo la *prospettiva parallela* in varie sue presentazioni. La presente sottosezione calcola la matrice di questa trasformazione prospettica in termini del vettore della direzione di proiezione: il risultato si chiama la matrice della *proiezione parallela*. Fissato il vettore di direzione **q** della proiezione, questa trasformazione prospettica manda il generico punto  $\mathbf{p} = (x, y, z)$  nel punto  $\mathbf{p}' = (x', y', z') \in V$  che appartiene alla retta che passa per **p** ed ha direzione **q**. In altre parole, **q** non è parallelo al piano V e

$$\mathbf{p}' \in V$$
 e  $\mathbf{p}' - \mathbf{p} = t\mathbf{q}$  (6.6.1)

dove  $t \in \mathbb{R}$  varia al variare di **p** (si noti che il suo segno è diverso nei due semispazi determinati dal piano V, e se **q** è normalizzato, come d'abitudine, allora |t| è la distanza da **p** al punto proiettato **p**'). Si noti anche che |t| non è la distanza da **p** al piano V a meno che il vettore **q** sia normale a V: in tal caso si dice che la proiezione è ortogonale, ovvero ortografica, e questa proiezione l'abbiamo già studiata nella Sezione 6.2. I punti del piano di visuale V ovviamente rimangono fissi: per loro t = 0. Vediamo prima un esempio semplice.

ESEMPIO 6.6.1. (Proiezione parallela sul piano di base.) Scegliamo  $V = \{(x, y, 0)\}$ . Allora  $\mathbf{q} = (q_x, q_y, q_z)$  con  $q_z \neq 0$  (perché  $\mathbf{q}$  non può

162

essere parallelo a V), e (6.6.1) diventa

$$x' - x = t q_x$$
  $y' - y = t q_y$   $z' - z = t q_z$ 

ma poiché z' = 0 (dal momento che **p**' sta nel piano di base V) si ottiene

$$t = -\frac{z}{q_z} \qquad x' = x - \frac{q_x}{q_z}z \qquad y' = y - \frac{q_y}{q_z}z$$

e quindi la matrice prospettica è

$$M_{\mathbf{q}}^{\mathrm{par}} = \begin{pmatrix} 1 & 0 & -\frac{q_x}{q_z} \\ 0 & 1 & -\frac{q_y}{q_z} \\ 0 & 0 & 0 \end{pmatrix} \ .$$

Il fatto che la terza riga sia tutta nulla è ovvio perché l'immagine della proiezione consiste di vettori con terza componente z = 0. In maniera naturale, questa proiezione si potrebbe quindi rappresentare come una matrice di due righe e tre colonne, da  $\mathbb{R}^3$  al sottospazio  $V \sim \mathbb{R}^2$  (in questo esempio V è un sottospazio vettoriale).  $\Box$ 

Ora studiamo il caso generale, in cui il piano V è generico. Sia  $\mathbf{n} = (n_x, n_y, n_z)$  il suo versore normale (ovvero, sia  $n_x x + n_y y + n_z z = d_0$  l'equazione di V: qui, come al solito, il significato geometrico di  $d_0$  è la distanza di V dall'origine). Ora (6.6.1) diventa il sistema lineare

$$x' - x = t q_x$$
$$y' - y = t q_y$$
$$z' - z = t q_z$$
$$x' + n_y y' + n_z z' = d_0.$$

Calcoli elementari portano alla soluzione seguente:

 $n_r$ 

PROPOSIZIONE 6.6.2. (Proiezione parallela.) La proiezione parallela in direzione del vettore  $\mathbf{q}$  sul piano V con versore normale  $\mathbf{n} = (n_x, n_y, n_z)$  e distanza dall'origine  $d_0$  è rappresentata dalla matrice

$$M_{\mathbf{q},V}^{\text{par}} = \frac{1}{\mathbf{q}\cdot\mathbf{n}} \begin{pmatrix} \mathbf{q}\cdot\mathbf{n} - q_x n_x & -q_x n_y & -q_x n_z & q_x d_0 \\ -q_y n_x & \mathbf{q}\cdot\mathbf{n} - q_y n_y & -q_y n_z & q_y d_0 \\ -q_z n_x & -q_z n_y & \mathbf{q}\cdot\mathbf{n} - q_z n_z & q_z d_0 \\ 0 & 0 & 0 & \mathbf{q}\cdot\mathbf{n} \end{pmatrix}.$$

ESERCIZIO 6.6.3. Si ritrovi la matrice della Proposizione 6.6.2 riconducendo il calcolo all'Esempio 6.6.1. SUGGERIMENTO. Si fissi un punto  $\mathbf{r} = (x_0, y_0, z_0) \in V$ , lo si trasli all'origine tramite la matrice affine di traslazione

$$T_{-\mathbf{r}} = \begin{pmatrix} 0 & 0 & 0 & -x_0 \\ 0 & 0 & 0 & -y_0 \\ 0 & 0 & 0 & -z_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} ,$$

L'unica difficoltà consiste nel ricavare la matrice di rotazione R. Come matrice affine, R è in realtà lineare, percè fissa l'origine, e quindi la sua ultima riga e colonna coincidono con (0, 0, 0, 1). L'inverso  $R^{-1}$  di R è uguale al suo trasposto  $R^*$ , perché R è una matrice ortonormale. D'altra parte  $R^* \mathbf{e}_3 = R^{-1} \mathbf{e}_3 = \mathbf{n}$ , e quindi la terza *riga* di R ha come prime tre componenti quelle di  $\mathbf{n}$ . Abbiamo un grado di libertà nella scelta di R, perché se prima di far agire R ruotiamo tutto lo spazio intorno all'asse di rotazione  $\mathbf{n}$  la trasformazione R non ne risente. e la trasformazione risultante continua a mandare n in  $e_3$ . Quindi possiamo scegliere la prima componente della seconda riga di R uguale a 0, ed allora, per l'ortogonalità con la terza riga, si ottiene che le prime tre componenti della seconda riga di R devono essere  $(0, n_z, -n_y)$  (a parte il segno, che è irrilevante, ma che comunque sceglieremo, nel prossimo passaggio, in modo che R abbia determinante +1 invece che -1: ma si troverebbe la soluzione anche con la scelta opposta). Occorre però normalizzare la seconda riga della matrice, perché le righe di una matrice ortonormale sono una base ortonormale. A questo punto, per l'ortonormalità, la prima riga è il prodotto vettoriale fra la seconda e la terza. Si verifichi che in tal modo si trova la seguente forma matriciale di R:

$$R = \begin{pmatrix} 1 & \frac{-n_x n_y}{\sqrt{1+n_x^2}\sqrt{n_y^2+n_z^2}} & \frac{-n_x n_z}{\sqrt{1+n_x^2}\sqrt{n_y^2+n_z^2}} & 0\\ 0 & \frac{n_z}{\sqrt{n_y^2+n_z^2}} & \frac{-n_y}{\sqrt{n_y^2+n_z^2}} & 0\\ n_x & n_y & n_z & 0\\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Si noti che abbiamo scritto R come una matrice a dimensione quattro, ma in realtà la rotazione è in tre dimensioni. Però R deve essere moltiplicata per matrici di trasformazioni affini che mandano il piano di base in un piano che non passa per l'origine, e quindi non è un sottospazio bidimensionale di  $\mathbb{R}^3$ . Pertanto queste trasformazioni affini hanno matrici a dimensione quattro, e per questo dobbiamo immergere la matrice di R in una matrice a dimensione quattro, prendendo la sua somma diretta con la matrice identità a dimensione 1. **6.6.2.** Proiezione obliqua. La matrice della *proiezione obliqua* è un'altra presentazione della trasformazione prospettica parallela, questa volta espressa non in termini del vettore direzionale bensì di un *fattore di accorciamento* e di un *angolo di azimuth*.

Supponiamo che il piano di proiezione sia il piano  $\{z = 0\}$ , e che il punto  $\mathbf{e}_3 = (0,0,1)$  sia proiettato sul punto  $\mathbf{p} = (x',y',0)$ . Se  $f = \sqrt{x'^2 + y'^2}$  e  $\theta$  è l'angolo fra il semiasse x positivo ed il vettore  $\mathbf{p}$ , allora  $x' = f \cos \theta$  e  $y' = f \sin \theta$ . Il fattore f misura l'accorciamento prospettico dell'asse della profondità (l'asse perpendicolare al piano di proiezione).

NOTA 6.6.4. Sebbene di solito il fattore di accorciamento f sia minore di 1, e quindi rappresenti un vero accorciamento dell'asse della profondità, possiamo avere prospettive con f > 1, in cui quindi abbiamo un allungamento. Ne vedremo un esempio in seguito nella Sottosezione 6.7.2. Naturalmente, l'allungamento dell'asse delle profondità provoca una distorsione prospettica, particolarmente evidente se f è grande.



FIGURA 9. Proiezione obliqua sul piano  $\{z = 0\}$ : fattore di accorciamento f e azimuth  $\theta$ .

Pertanto il vettore della direzione di proiezione è

$$\mathbf{q} = \mathbf{p} - \mathbf{e}_3 = (f \cos \theta, f \sin \theta, -1),$$

e dall'Esempio 6.6.1 troviamo la forma seguente della matrice di proiezione:

$$M_{f,\theta}^{\text{par}} = M_{\mathbf{q}}^{\text{par}} = \begin{pmatrix} 1 & 0 & f \cos \theta \\ 0 & 1 & f \sin \theta \\ 0 & 0 & 0 \end{pmatrix} \quad . \tag{6.6.2}$$

Se si preferisce, si può immergere questa matrice in una a quattro dimensioni, come abbiamo fatto prima nell'Esercizio 6.6.3:

$$M_{f,\theta}^{\text{par}} = \begin{pmatrix} 1 & 0 & f \cos \theta & 0 \\ 0 & 1 & f \sin \theta & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} .$$

La trasformazione diventa

$$\begin{aligned} x' &= x + zf\cos\theta\\ y' &= y + zf\sin\theta. \end{aligned} \tag{6.6.3}$$

Da questo caso particolare si passa al caso generale di piano di proiezione arbitrario tramite coniugazione con la matrice della trasformazione affine che mappa tale piano sul piano  $\{z = 0\}$ , come accennato nell'Esercizio 6.6.3. Come visto in quell'Esercizio, quando il piano di proiezione non passa per l'origine, e quindi non è un sottospazio vettoriale, questa matrice affine è una vera matrice a dimensione quattro (non è l'immersione quadridimensionale di una matrice a dimensione tre), e per questo è opportuno immergere R in uno spazio a quattro dimensioni.

In realtà, procedendo in base a semplici illustrazioni geometriche, possiamo talvolta evitare il procedimento di moltiplicazione matriciale, algebricamente accurato ma geometricamente piuttosto oscuro. Osserviamo anzitutto che la trasformazione prospettica data dalla matrice 6.6.2 cambia aspetto a seconda del piano di proiezione. Anche nel caso in cui questo piano passi per l'origine, ovvero sia un sottospazio vettoriale, i coefficienti della trasformazione dipendono dalla scelta di una base in questo sottospazio. In generale, anche se il piano di proiezione è un sottospazio affine (ossia non passa per l'origine di  $\mathbb{R}^3$ ), l'espressione della trasformazione, rispetto ad una sua base affine, dipende dalla scelta di tale base.

Ad esempio, consideriamo il caso del piano di proiezione  $\{z = -1\}$ . Rispetto a prima, dobbiamo traslare da z = 0 a z = -1. Consideriamo adesso la trasformazione prospettica come mappa dalle coordinate canoniche di  $\mathbb{R}^3$  ad una scelta di coordinate intrinseche sul piano di proiezione: allora è inutile considerare la traslazione nella variabile z, la quale individua solo la traslazione dell'origine di  $\mathbb{R}^3$  alla nuova origine (0, 0, -1) del piano affine  $\{z = -1\}$ . Pertanto, in tal modo ci riconduciamo sempre alla situazione in cui il piano di proiezione è  $\{z = 0\}$ : assumiamo dunque d'ora in poi che questo sia il piano di proiezione. Se volessimo ritornare al caso della proiezione su un piano V in  $\mathbb{R}^3$  con le coordinate di V indotte dalle coordinate canoniche di  $\mathbb{R}^3$ , basterebbe comporte con una trasformazione affine da V al piano di base  $B = \{z = 0\}$ , ovvero con l'inverso di una trasformazione affine da B a V: ad esempio, nel caso di cui sopra,  $V = \{z = -1\}, e \text{ la trasformazione affine ovviamente } e^{-1} z \mapsto z - 1, z \mapsto z - 1$ il cui inverso è  $z \mapsto z + 1$ : quindi, in tutte le prossime formule basta sostituire  $z \operatorname{con} z + 1$ . Osserviamo che la scelta di questa trasformazione affine ammette un solo grado di libertà: data una trasformazione affine da V a B, il piano di arrivo B resta lo stesso solo se dopo la trasformazione affine operiamo una rotazione intorno al versore normale di B, ossia l'asse z. Quindi possiamo supporre, senza perdita di generalità, che la proiezione del punto  $\mathbf{e}_3 = (0, 0, 1)$  giaccia sull'asse x, ossia si abbia  $\theta = 0$  nella matrice (6.6.2).

Scriviamo quindi la trasformazione relativamente alla scelta di base, in questo piano, concorde con la restrizione della base canonica di  $\mathbb{R}^3$ , ossia formata dai versori  $\mathbf{e}_1$  dell'asse  $x \in \mathbf{e}_2$  dell'asse y: in generale indichiamo con  $\mathbf{i} \in \mathbf{j}$  i due versori di base nel sottospazio di proiezione (o i vettori della base affine se, come nel caso presente, il piano di proiezione non passa per l'origine), e chiamiamo  $x' \in y'$  i corrispondenti assi coordinati in tale piano. Con la scelta naturale di base nel piano di proiezione data come sopra dalla restrizione, nelle nuove coordinate  $x' \in y'$  la trasformazione ritorna ad essere quella data dalle identità (6.6.3).

Per comodità, scambiamo fra loro gli assi  $y \in z$  in  $\mathbb{R}^3$ : l'asse delle profondità è ora l'asse y, il piano di proiezione è  $\{y = 0\}$  e la trasformazione diventa

$$\begin{aligned} x' &= x + yf\cos\theta\\ y' &= yf\sin\theta + z \,. \end{aligned} \tag{6.6.4}$$

Ora operiamo un'altra scelta di base, in maniera tale che gli assi x e y siano visti, in prospettiva, come nella Figura 10: il semiasse y' positivo coincide nella visualizzazione con il semiasse z positivo, il semiasse y positivo forma un angolo  $\alpha$  (con  $0 < \alpha < \frac{\pi}{2}$ ) con il semiasse x' positivo ed è orientato nel verso positivo di x', ed infine il semiasse x positivo forma un angolo  $\beta$  (con  $0 < \beta < \frac{\pi}{2}$ ) con il semiasse x' negativo (e quindi è orientato nel verso negativo di x'). Questa è la trasformazione obliqua più naturale per rappresentare il grafico di una funzione definita su un quadrato centrato nell'origine in maniera che la vista sia non

frontale bensì ad angolo. Adesso, per tener conto dell'eventualità di voler cambiare le proporzioni e la grandezza dell'immagine proiettata, è opportuno utilizzare non più un solo fattore di accorciamento f, ma tre, per descrivere gli accorciamenti dei versori di tutti e tre gli assi: indichiamo questi fattori con  $f_x$ ,  $f_y$  e  $f_z$  (l'ultimo è quello che prima abbiamo chiamato f): i loro valori sono le lunghezze dei trasformati prospettici dei versori della base canonica  $\mathbf{e}_1$ ,  $\mathbf{e}_2$ ,  $\mathbf{e}_3$ , rispettivamente. Ovviamente, se non si è interessati a controllare le proporzioni (ossia il rapporto fra larghezza ed altezza) dell'immagine, per ragioni di simmetria si porre  $f_x = f_y$  (la scelta di  $f_z$  può essere diversa per via della proiezione obliqua, che privilegia l'asse z). In effetti, in seguito useremo questa prospettiva per la rimozione di linee nascoste in grafici prospettici tridimensionali, e nella versione più realistica porremo  $f_x = f_y$ (Nota 6.7.1 e Sottosezione 6.7.3), ma avremo bisogno di disegnare il grafico con fattori di scala diversi sui due assi  $x \in y$ : allora trarremo vantaggio della possibilità di scegliere  $f_x \neq f_y$  (si veda l'algoritmo che conduce all'identità (6.7.5)).

In seguito alla rotazione e traslazione, e quindi alla scelta dei due angoli di inclinazione prospettica ed ai tre fattori di accorciamento, la trasformazione ora diventa:

$$x' = -xf_x \cos\beta + yf_y \cos\alpha \tag{6.6.5}$$

$$y' = -xf_x \sin\beta - yf_y \sin\alpha + zf_z \tag{6.6.6}$$

(si osservi che i segni meno corrispondono al fatto che, al crescere di  $x, x' \in y'$  decrescono, ed al crescere di y si ha che x' decresce ma y' cresce, per come la scelta dei versi di visualizzazione degli assi (si veda la Figura 10).

NOTA 6.6.5. La scelta di usare  $f_x$  e  $f_y$  diversi può essere molto conveniente in Computer Graphics, perché spesso occorre adattare le dimensioni orizzontali e verticali sulla finestra di visuale per adattare l'output alle proporzioni dei monitor, che non sono di solito quadrate, bensì in rapporto 4:3 o 16:9. In tal caso, si può scrivere il codice grafico in maniera che, alla scelta delle proporzioni del monitor, corrisponda una scelta analoga del rapporto  $f_x : f_y$ : in tal modo l'output si riscala nel passaggio da un monitor ad un altro (ad esempio, può riempire esattamente entrambi i monitor, senza tagli).

Proiezioni cavaliera e cabinet. Il caso della proiezione obliqua senza accorciamento, ossia con fattore di accorciamento f = 1, si chiama proiezione cavaliera. Sappiamo da (6.6.2) che la forma tridimensionale



FIGURA 10. Visualizzazione in proiezione obliqua con vista non frontale bensì angolare.

della sua matrice è

$$M_{1,\theta}^{\text{par}} = \begin{pmatrix} 1 & 0 & \cos\theta \\ 0 & 1 & \sin\theta \\ 0 & 0 & 0 \end{pmatrix} .$$
 (6.6.7)

Questa proiezione è spesso usata nel disegno tecnico per scopi militari, poiché la sua immagine fornisce la mappa prospettica di una zona vista dall'alto senza distorzioni di accorciamento. Se invece proiettiamo sul piano  $\{x = 0\}$  oppure  $\{y = 0\}$  otteniamo le proiezioni laterali o frontali senza accorciamenti (anche esse utili nel campo militare o architettonico e nel disegno tecnico), le cui matrici si ottengono dalla precedente tramite coniugazione per la matrice ortogonale del cambiamento di base che manda  $\mathbf{e}_3$  in  $\mathbf{e}_1$ ,  $\mathbf{e}_1$  in  $\mathbf{e}_2$  e  $\mathbf{e}_2$  in  $\mathbf{e}_3$ , ossia

$$\left(\begin{array}{rrr} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{array}\right)$$

oppure, rispettivamente, la sua trasposta: in tal modo si vede che la proiezione cavaliera laterale (sul piano  $\{x = 0\}$ ) ha per matrice

$$\left(\begin{array}{rrr} 0 & 0 & 0\\ \cos\theta & 1 & 0\\ \sin\theta & 0 & 1 \end{array}\right)$$

e quella frontale (sul piano  $\{y = 0\}$ ) ha per matrice

$$\left(\begin{array}{ccc} 1 & \cos\theta & 0\\ 0 & 0 & 0\\ 0 & \sin\theta & 1 \end{array}\right) . \tag{6.6.8}$$

•

Un'altra scelta tipica di fattore di accorciamento, usata spesso nel disegno tecnico, è  $f = \frac{1}{2}$ . La proiezione corrispondente si chiama *proiezione cabinet*. Anche in tal caso possiamo trascrivere la matrice corrispondente, ad esempio sul piano di base  $\{z = 0\}$ :

$$M_{\frac{1}{2},\theta}^{\text{par}} = \begin{pmatrix} 1 & 0 & \frac{1}{2}\cos\theta\\ 0 & 1 & \frac{1}{2}\sin\theta\\ 0 & 0 & 0 \end{pmatrix}$$

Le scelte tipiche di azimuth per un disegno realistico sono  $\theta = \frac{\pi}{4}$  e  $\theta = \frac{\pi}{6}$ . La prima porta alla matrice di proiezione

$$M_{\frac{1}{2},\frac{\pi}{4}}^{\text{par}} = \begin{pmatrix} 1 & 0 & \frac{1}{2\sqrt{2}} \\ 0 & 1 & \frac{1}{2\sqrt{2}}2 \\ 0 & 0 & 0 \end{pmatrix} , \qquad (6.6.9)$$

la seconda alla matrice

$$M_{\frac{1}{2},\frac{\pi}{4}}^{\text{par}} = \begin{pmatrix} 1 & 0 & \frac{\sqrt{3}}{4} \\ 0 & 1 & \frac{1}{4} \\ 0 & 0 & 0 \end{pmatrix} \quad . \tag{6.6.10}$$

**6.6.3.** Vari tipi di proiezioni ortogonali. Rammentiamo che le proiezioni ortogonali sono le proiezioni parallele la cui direzione di proiezione è perpendicolare al piano di proiezione. Nella Sezione 6.2 abbiamo trattato il caso in cui la direzione di proiezione sia parallela all'asse z, o ad uno degli assi coordinati. In generale, si dice che la proiezione è

- *isometrica* se la direzione di proiezione determina angoli uguali con tutti e tre gli assi coordinati;
- *dimetrica* se la direzione di proiezione forma angoli uguali con due assi coordinati;

• *trimetrica* se i tre angoli formati dalla direzione di proiezione con gli assi coordinati sono tutti diverso.

Nel caso isometrico, il vettore normale al piano di proiezione è proporzionale a  $(\pm 1, \pm 1, \pm 1)$ : ci sono otto direzioni possibili, quelle delle bisettrici degli otto ottanti (se non si ha interesse ad eliminare linee nascoste, conta solo la direzione di proiezione e non il verso, e quindi le direzioni sono a due a due equivalenti: rimangono solo quattro casi).

# 6.7. Applicazione: rimozione di linee nascoste in grafici 3D

Prendiamo in esame un problema tipico della Computer Graphics, quello del disegno del grafico di una funzione F di due variabili, che per il momento, seguendo la notazione usuale in matematica, indichiamo con x e y. Vogliamo proiettare i punti (x, y, z = F(x, y)) sul piano frontale  $\{y = 0\}$ : quindi ora y misura la profondità rispetto al piano di proiezione. Supponiamo per semplicità che la funzione sia stata calcolata sui punti  $(x_j, y_m)$  di una griglia di passo  $\tau$ , con  $j, m = 1, \ldots, N$ . Per semplicità poniamo  $x_i = j\tau$ ,  $y_m = m\tau$ . Osserviamo che in questo modo i valori  $y_m$  di profondità della griglia sono tutti positivi, e quindi dallo stesso lato del piano di proiezione  $\{y = 0\}$ : è opportuno che sia così affinché il disegno sia realistico. Poniamo  $z_{jm} = (x_j, y_m, F(x_j, y_m)).$ Vogliamo proiettare prospetticamente i punti  $\mathbf{p}_{im} = (x_i, y_m, z_{im})$  sul piano  $\{y = 0\}$  e poi unire i punti proiettati con segmenti, in modo che la proiezione prospettica di ciascun punto  $\mathbf{p}_{im}$  sia unita a quella del suo predecessore e del suo successore sulla stessa linea x = costante(ossia  $\mathbf{p}_{j\pm 1,m}$ ) e y = costante (ossia  $\mathbf{p}_{j,m\pm 1}$ ). In questo modo otteniamo una famiglia a due parametri di spezzate nel piano di proiezione, che chiamiamo le poligonali x = costante e y = costante. Esse producono un disegno a maglie del grafico, fatto con una rete poligonale, la quale però, se i punti della griglia sono vicini rispetto al tasso di variazione di F (ossia se la griglia è fitta:  $\tau$  piccolo), approssima bene un disegno a curve lisce. Nella Figura 11 illustriamo il grafico a maglie della funzione  $\sin(x^2 + y^2)$  su un quadrato centrato nell'origine.

Il problema però è che questo disegno a maglie è trasparente: ci si vede in mezzo, le parti posteriori del grafico risultano visibili attraverso le maglie anteriori. Dovremmo quindi rimuovere le zone posteriori nascoste. Se lo facciamo, otteniamo il grafico in Figura 12. Ma come si deve procedere al fine di eliminare le linee nascoste?



FIGURA 11. Grafico a maglie di $\sin(x^2+y^2)$ senza rimozione delle linee nascoste



FIGURA 12. Grafico a maglie di $\sin(x^2+y^2)$  con rimozione delle linee nascoste

6.7.1. Rimozione di linee prospetticamente nascoste. Proviamo a trovare un algoritmo per rimuovere le parti nascoste. Per semplicità, restringiamo per ora l'attenzione alle poligonali x = costante, ossia quelle con punti nodali  $\mathbf{p}_{jm}$ ,  $j = 0, \ldots N$ , dove  $0 \leq m \leq N$  è fissato. Osserviamo che la loro profondità in tre dimensioni varia proporzionalmente a m: infatti la m-sima poligonale x = costante giace in un piano, questi piani sono paralleli ed all'aumentare di m si allontanano dal piano di proiezione (nel caso non si sia scelto di far aumentare la profondità all'aumentare di m, occorre ora cominciare il disegno a partire da m = 0 oppure da m = N a seconda di quale delle due poligonali, ossia di quale dei quattro vertici del rettangolo della griglia, sia più vicino al punto di visuale).

La prima di queste poligonali, per m = 0, viene proiettata sul piano di proiezione ed interamente disegnata: è la più frontale, nulla può nasconderla. Anche la seconda viene interamente disegnata: non può infatti essere nascosta dalla prima, può certo accavallarsi e passarle da sopra a sotto o viceversa, ma in entrambi i casi, o da sopra o da sotto, è visibile. Si ottiene così lo schizzo della Figura 13.



FIGURA 13. Le prime due spezzate x = costante in un grafico 3D prospettico

Ma a partire dalla terza poligonale le cose cambiano. Infatti, le prime due ora racchiudono un'area opaca nel piano di proiezione, che nasconde la zona retrostante, e quindi attraverso la quale non si deve vedere dietro. L'area può avere l'aspetto di una striscia se le prime due poligonali non si accavallano, ed altrimenti ha una o più strozzature. I segmenti proiettati della terza poligonale vengono rimossi per la parte che penetra dentro tale area. La terza poligonale si aggiunge alla prime due ed aumenta l'area di opacità, ossia di invisibilità. In effetti, dopo aver tracciato m poligonali, tale area è delimitata dai *profili massi*mo e minimo che il disegno forma fino a quel momento sul piano di proiezione. In Figura 14 schizziamo il risultato del tracciamento della terza e quarta poligonale.



FIGURA 14. Le prime quattro spezzate x = costante: a partire dalla terza, esse non sono visibili nell'area fra i profili massimo e minimo precedenti.

Nel caso generale nel quale si intenda tracciare il grafico a maglia completo, quindi sia le poligonali x = costante sia quelle y = costante, per tener conto del fatto che la visibilità di segmenti delle poligonali del primo tipo potrebbe essere affetta dall'andamento di quelle del secondo tipo e viceversa, modifichiamo l'ordine di tracciamento in modo che le due famiglie di poligonali siano simultaneamente disegnate all'aumentare della profondità. Ovviamente, al fine di garantire il corretto aggiornamento dei profili all'aumentare della profondità, a prima vista sembra opportuno cominciare dalla poligonale più frontale: prima poligonale x = costante oppure y = costante a seconda di quale fra i lati esterni x e y del rettangolo della griglia sia più vicino al punto di osservazione. In realtà, però, questa cautela è inessenziale nel nostro caso: pur di procedere incrementando la distanza dall'osservatore, qualunque dei due ordini nella scelta fra x = costante e y = costanteproduce lo stesso risultato. Infatti i due lati anteriori della stessa maglia nelle due poligonali x =costante e y = costante si posizionano su aree del piano di visuale nonsovrapposte verticalmente, in quanto questo è ciò che accade per la griglia di base, e lo stesso accade per i due lati posteriori (si veda la Figura 10, e la si adatti ispirandosi alle Figure 15 e 17) e la posizione dei nodi delle poligonali differisce da quello dei nodi della griglia solo per spostamenti verticali, in base alla forma della trasformazione prospettica (6.6.5). Da qui segue che, nel disegnare ciascuna maglia rettangolare (o meglio a quadrilatero) del disegno, non può succedere che, una volta tracciati i segmenti corrispondenti al lato di griglia più frontale diciamo x = costante e quello più frontale di tipo y = costante,i segmenti relativi ai lati posteriori x = costante e y = costante venganoindebitamente nascosti dai profili massimo e minimo stabiliti nel tracciamento dei segmenti precedenti; quindi non si commettono errori di aggiornamento dei profili nel tracciare prima il segmento y = costantee poi i due segmenti limitrofi x = costante in ordine di distanza crescente dall'osservatore, e neppure se ne commettono se si traccia prima il segmento x = costante e poi i due segmenti limitrofi y = costantein ordine di distanza crescente dall'osservatore. Lasciamo al lettore il compito di tracciare opportuni disegni per visualizzare questo fatto. Nondimeno, per semplicità concettuale, continueremo a tracciare prima la poligonale più vicina all'osservatore (ovvero, quella più frontale). A parte la determinazione di quale sia il lato più vicino, è invece essenziale tracciare il grafico a partire dal suo vertice più vicino al punto di visuale (poiché sia i vertici della griglia sia il punto di visuale sono fissati prima dell'inizio della procedura di disegno, queste scelte sono semplici da effettuare; in questa presentazione abbiamo scelto gli indici

Il procedimento è il seguente. Invece di disegnare consecutivamente tutti i segmenti corrispondenti ai tratti relativi ai punti nodali  $\mathbf{p}_{jm}$ ,  $j = 0, \ldots N$ , con  $0 \leq m \leq N$  è fissato (ossia la m-sima poligonale x =costante) e poi tutti quelli del tipo y = costante, tracciamo invece il primo segmento di ciascuna poligonale x = costante, ossia il trasformato prospettico del segmento da  $\mathbf{p}_{0m}$  a  $\mathbf{p}_{1m}$ , poi i trasformati dei due segmenti delle poligonali y = costante che vi si appoggiano, ossia i segmenti da  $\mathbf{p}_{0m}$  a  $\mathbf{p}_{0,m+1}$  e da  $\mathbf{p}_{1m}$  a  $\mathbf{p}_{1,m+1}$ , e così via a maglie adiacenti. Durante il disegno, la cella  $\{j \rightarrow j + 1, m \rightarrow m + 1\}$  viene gestita tracciando prima il segmento da  $\mathbf{p}_{jm}$  a  $\mathbf{p}_{j+1,m}$  seguito dai due segmenti trasversali  $\mathbf{p}_{jm}$  a  $\mathbf{p}_{j,m+1}$  e da  $\mathbf{p}_{j+1,m}$  a  $\mathbf{p}_{j+1,m+1}$ : di ciascuno di questi segmenti tridimensionali disegniamo, in questo ordine, i trasformati prospettici. Ma ovviamente non occorre tracciare due volte lo stesso

della griglia in maniera da cominciare con il vertice di indici massimi).

segmento, quindi per j > 0 basta tracciare i segmenti nell'ordine

$$\mathbf{p}_{jm} \to \mathbf{p}_{j+1,m}$$
$$\mathbf{p}_{j,m} \to \mathbf{p}_{j,m+1} \,. \tag{6.7.1}$$

In questo modo i profili minimo e massimo vengono aggiornati correttamente al crescere della profondità.

Il problema ora consiste nel rimuovere dal disegno, ossia evitare di tracciare, quelle parti dei segmenti delle poligonali che penetrano nella striscia di opacità, ossia le cui altezze sul piano di proiezione sono comprese fra i profili minimo e massimo. In linea di principio, il procedimento è ovvio: basta tracciare i segmenti prospettici sul piano di proiezione un pixel per volta e, prima di tracciare, verificare se il pixel verifica la condizione di invisibilità (altezza compresa fra i profili minimo e massimo): se è così si omette il disegno, altrimenti si disegna. Naturalmente occorre ricavare, per ogni colonna di pixel, le coordinate del pixel di intersezione fra il segmento e quella colonna; la discretizzazione del segmento ai singoli pixel può provocare errori di arrotondamento, ma limitatamente ad uno scarto di un pixel.

Purtroppo in una tipica immagine ci sono almeno un migliaio di pixel in ascissa, ossia un migliaio di colonne. Anche considerando grafici a maglie di bassa risoluzione (una quarantina di linee x = costante), abbiamo bisogno di invocare la routine grafica di tracciamento delle linee circa 40.000 volte (ogni volta per una linea lunga un pixel), anche se stiamo solo cercando di disegnare 1600 segmenti. Questo accade per ciascun singolo disegno, con una aggravio piuttosto oneroso di elaborazione grafica.

Se vogliamo sviluppare un algoritmo rapido di disegno tramite computer, è opportuno procedere diversamente. Osserviamo che il tracciamento di una linea sul viewport discretizzato (composto di singoli pixel) viene effettuato tramite un algoritmo incrementale (algoritmo di Bressenham) che a ciascun passo decide se accendere il pixel successivo orizzontalmente o verticalmente a seconda della pendenza della linea e di quanto fatto al passo precedente: si veda il capitolo sulla Scan Conversion. Quindi è sufficiente modificare l'algoritmo di Bressenham in modo che, oltre alla scelta se accendere il pixel in direzione orizzontale o verticale, faccia anche la scelta se accenderlo o no a seconda del fatto che la condizione di visibilità sia soddisfatta. Questo procedimento funziona quale che sia la prospettiva utilizzata: però occorre riscrivere la routine della libreria grafica responsabile del tracciamento delle linee, e quindi ricompilarla. Questa riscrittura deve essere fatta in linguaggio macchina, oppure nell'Assembler compatibile con il processore del computer, oppure ancora in un linguaggio ad alto livello che abbia accesso

ai puntatori alla memoria grafica (nell'ultimo caso, però, l'elaborazione è più lenta che negli altri due). Se si preferisce evitare questo fastidio, si può ricorrere ad un diverso algoritmo, che presentiamo qui di seguito, basato su un allineamento intelligente dei punti della griglia di base ed una scelta appropriata dei parametri prospettici.

6.7.2. Algoritmo veloce di rimozione di linee nascoste in grafici 3D: allineamento della griglia. Per un algoritmo veloce di rimozione delle linee nascoste impieghiamo la proiezione obliqua introdotta nella Sottosezione 6.6.2.

Rammentiamo che abbiamo denotato con  $z_{jm} = (x_j, y_m)$  i nodi della griglia di base e con  $\mathbf{p}_{jm} = (x_j, y_m, z_{jm})$  i punti nodali del grafico a maglie della funzione F su questa griglia. Chiamiamo  $\mathbf{r}_{jm}$  la proiezione prospettica di  $\mathbf{p}_{jm}$  sul piano  $\{y = 0\}$  e  $\rho_{(ij),(i+1,j)}$  (rispettivamente,  $\rho_{(ij),(i,j+1)}$ ) il segmento su questo piano che congiunge  $\mathbf{r}_{ij}$  con  $\mathbf{r}_{i+1,j}$ (rispettivamente, con  $\mathbf{r}_{i,j+1}$ ).

Il problema, come abbiamo visto, consiste nel fatto che i segmenti  $\rho_{(ij),(i+1,j)} \in \rho_{(ij),(i,j+1)}$  devono essere disegnati solo per le parti che non penetrano nell'area del piano di visuale compresa fra i profili minimo e massimo. Se questo controllo viene eseguito prima di disegnare ciascun pixel il procedimento risulta accurato ma lento. L'algoritmo che proponiamo si basa sull'impiego della prospettiva obliqua e sullo scegliere i suoi parametri in modo che i punti iniziali e finali di questi segmenti siano disposti su linee verticali, ossia allineati verticalmente. Questo è possibile perché, in base alle regole di trasformazione (6.6.4) e (6.6.5), il valore della coordinata z, ossia il valore della funzione di cui tracciamo il grafico, non influenza il valore dell'ascissa x' sul piano di visuale, e quindi i punti estremi dei segmenti  $\rho_{(ij),(i+1,j)} \in \rho_{(ij),(i,j+1)}$  sono allineati verticalmente se e solo se lo sono i trasformati prospettici dei punti nodali della griglia di base.

In tal modo, occorre solo verificare la condizione di non appartenenza all'area di invisibilità all'inizio ed alla fine del segmento che stiamo disegnando: se entrambi i punti estremi sono visibili tracciamo l'intero segmento, altrimenti la linearità ci permette di calcolare immediatamente quale parte tracciare. È quindi necessario memorizzare soltanto le altezze dei profili minimo e massimo in corrispondenza delle colonne di pixel che corrispondono ai punti estremi dei segmenti  $\rho_{(ij),(i+1,j)}$  e  $\rho_{(ij),(i,j+1)}$ , ossia, come appena osservato, i trasformati dei punti nodali della griglia di base.

Qui però dobbiamo affrontare un problema. Nel caso un segmento debba essere tracciato solo in parte, diciamo nella parte iniziale, il punto terminale di questa parte tracciata diventa un nuovo punto per il quale occorre memorizzare i valori dei profili minimo e massimo, in aggiunta a quelli provenienti dalla griglia di base. Ma in questo modo si perderebbe l'allineamento verticale, e la dimensione degli array in cui memorizzare i valori di tali profili potrebbe raddoppiarsi ad ogni passo: questa crescita esponenziale distruggerebbe i vantaggi di velocità di calcolo dell'algoritmo. Conviene pertanto omettere la memorizazione di questi valori intermedi dei profili minimo e massimo. In tal caso, all'ascissa del punto intermedio il profilo massimo viene sottostimato e quello minimo sovrastimato. Come conseguenza, il resto del disegno può omettere qualche piccolo tratto, una imprecisione tollerabile se la risoluzione è elevata, ossia se il passo di incremento di griglia  $\tau$  è piccolo.

Naturalmente, all'inizio del procedimento bisogna determinare se cominciare dalla prima poligonale x = costante oppure y = costante.Rammentiamo che si deve cominciare da quella delle due più vicina al punto di osservazione: pertanto, dall'asse che ha angolo di inclinazione minore. Questa semplice scelta è necessaria solo se il punto di vista è angolato, e quindi ci sono due angoli di inclinazione non nulli: nel caso di punto di vista frontale, l'angolo dell'asse frontale è zero e si comincia con questo.

Aggiorniamo la notazione. Abbiamo chiamato  $\tau$  l'incremento delle coordinate  $x \in y$  fra due consecutive rette  $x = \text{costante} \in y = \text{costante}$ della griglia di base. Denotiamo con  $\Delta x$  l'incremento delle ascisse sul piano di visuale (ossia della variabile x') ottenuto dopo la trasformazione prospettica quando si passa da una retta x = costante della griglia di base alla successiva (ossia da  $(x_j, y_m)$  a  $(x_{j+1}, y_m)$ ) e con  $\Delta y$  l'incremento ancora delle ascisse x' al progredire delle rette y = costante (ossia da  $(x_j, y_m)$  a  $(x_j, y_{m+1})$ ). La condizione di allineamento verticale è

$$\Delta x = \Delta y \,. \tag{6.7.2}$$

Per ottenere l'allineamento voluto, consideriamo dapprima il caso di vista frontale dato dalla trasformazione di coordinate (6.6.4), dalla quale si ottiene

$$\Delta x = \tau \tag{6.7.3}$$

$$\Delta y = \tau f \cos \theta \,. \tag{6.7.4}$$

Pertanto la condizione di allineamento è soddisfatta se e solo se  $f \cos \theta =$ 1. Poiché ovviamente  $\cos \theta \leq 1$ , si deve avere  $f \geq 1$ . L'allineamento dopo la trasformazione prospettica è illustrato in Figura 15. Naturalmente, se  $\theta$  non è piccolo (ovvero se è più vicino a  $\frac{\pi}{2}$  che a 0), allora fè grande e si ha una forte distorsione prospettica, come osservato nella Nota 6.6.4.


FIGURA 15. Allineamento prospettico verticale della griglia di base in proiezione obliqua frontale

Per evitare la distosione prospettica, possiamo adottare una variante più complessa dell'algoritmo, nella quale, invece di allineare verticalmente ogni riga y = costante di punti della griglia, si allinea verticalmente ogni seconda riga successiva, mentre i punti nodali della righe dispari vengono disposti verticalmente a metà strada fra le colonne di allineamento delle righe pari (si veda la Fig. 24). In questo modo la condizione da soddisfare diventa  $f \cos \theta = \frac{1}{2}$ , e quindi possiamo scegliere f più piccolo di prima, ma si raddoppia la lunghezza degli array dei profili minimo e massimo.

Veniamo ora al caso, più verisimile, della prospettiva obliqua con vista angolata, data dalla trasformazione (6.6.5). In base a questa trasformazione, invece delle identità (17), ora abbiamo

$$\Delta x = -\tau f_x \cos \beta$$
$$\Delta y = \tau f_y \cos \alpha \,,$$

e quindi la condizione di allineamento (6.7.2) diventa

$$f_x \cos \beta = f_y \cos \alpha \,. \tag{6.7.5}$$



FIGURA 16. Allineamento prospettico a due passi della griglia, in proiezione obliqua frontale

Una volta scelti i fattori di accorciamento, possiamo adattare gli angoli  $\alpha \in \beta$  di inclinazione prospettica degli assi in modo da soddisfare questa relazione, o viceversa. Questo completa l'algoritmo.

NOTA 6.7.1. Talvolta, per visualizzare un grafico in due variabili in un rettangolo, fissiamo il centro  $(X_0, Y_0)$  di un rettangolo sul piano xy con i lati paralleli agli assi, e le lunghezze  $\Delta X$  e  $\Delta Y$  dei due lati, e scegliamo una griglia equiripartita suddividendo in N-1 parti i due lati. In tal modo ciascuna cella della griglia ha lati non necessariamente quadrati, di lunghezze rispettive  $\Delta X/(N-1)$  e  $\Delta Y/(N-1)$  invece che  $\tau$ . Allora la condizione di allineamento (6.7.5) diventa

$$\Delta X = \Delta Y \frac{f_x \cos \alpha}{f_y \cos \beta}$$

In tal caso, la versatilità aggiuntiva data dai parametri di accorciamento  $f_x e f_y$  non serve più, perché viene rimpiazzata dalla libertà di scegliere a piacimento, e non necessariamente uguali, le lunghezze  $\Delta X e \Delta Y$  dei lati (tranne che per le esigenze di invarianza dell'output al cambiare del monitor spiegate nella Nota 6.6.5). Pertanto di solito si pone  $f_x = f_y$ 



FIGURA 17. Allineamento prospettico verticale della griglia di base in proiezione obliqua angolata

e la regola di allineamento diventa

$$\Delta X = \Delta Y \, \frac{\cos \alpha}{\cos \beta} \, .$$

Per ogni scelta dei rapporti dei lati, si fissa un angolo di inclinazione  $\alpha$ e si determina l'angolo  $\beta$  in modo che soddisfi questa regola (badando a non compiere scelte di angoli troppo estreme per non distorcere la prospettiva). 6.7.3. Appendice: pseudocodice per la rimozione di linee nascoste. Per concludere l'analisi dell'algoritmo, ora scegliamo la versione più tipica e verisimile, quella illustrata nella Nota 6.7.1, e ne presentiamo lo pseudocodice, omettendo le parti di scelta dei parametri grafici (numero di punti della griglia, angoli di inclinazione, scelta dell'asse da cui cominciare il disegno, rinormalizzazioni delle scale x e yper adattare l'output alle proporzioni del monitor, calcolo dell'array dei valori della funzione, inizializzazioni degli array dei profili). Per motivi di chiarezza didattica, lo pseudocodice è scritto in Pascal (le istruzioni grafiche di tracciamento delle linee e spostamento del cursore usano la sintassi di TurboPascal), ma da qui è facile trasportare il codice ad altri linguaggi, ad esempio C o Fortran. Occorre comunque assicurarsi che la routine di tracciamento di un segmento (x1, y1) - (x2, y2) sulla periferica di output abbia la sintassi line(x1, y1, x2, y2), o altrimenti modificare la sintassi appropriatamente.

Questo codice è una nuova redazione, dove sono stati corretti alcuni errori e reso notevolmente più chiaro lo sviluppo logico, di versioni precedenti nei linguaggi Basic, Fortran e Pascal pubblicate nelle seguenti tesi di laurea presso l'Università di Roma "La Sapienza", sotto la direzione di M.Picardello: E. Franti, Algoritmi per grafica tridimensionale prospettica (1982-83), M. Sabbatucci, Integrazione numerica di equazioni alle derivate parziali ed algoritmi di rappresentazione prospettica delle loro soluzioni (1984-85), M. P. Merola, Integrazione dell'equazione delle onde mediante trasformata rapida di Fourier e visualizzazione grafica della loro soluzione (1986-87).

#### PROCEDURE DRAW

```
type punto, puntoprec: array[0..100,0..1] of real;
type fun array[0..100,0..100] of real;
type profmax, profmin: array[0..100] of real;
```

```
var alpha,beta,tau1,tau2,tau3,tau4,tau5: real;
axmax,axmin,aymax,aymin,x1,x2,y1,y2: real;
scalex,scaley,offsetx,offsety,xmax,xmin,ymax,ymin: real;
x1,y1,x2,y2,xprof1,yprof1,xprof2,yprof2: real;
num: integer;
fullyvisible,invisible,drawsegment1,drawsegment2: boolean;
```

```
PROCEDURE INIT
{inizializza i valori da disegnare e gli angoli prospettici}
BEGIN {of procedure init}
  {... codice di inizializzazione, e acquisizione dei
                                                        valori
della funzione (array fun) ...}
tau1=cos(beta);
tau2=cos(alpha);
tau3=sin(beta);
tau4=sin(alpha);
tau5=f;
END; {of procedure init}
PROCEDURE WINDOW (var wxmin,wymin,wxmax,wymax: real);
{Calcola i coefficienti per la trasformazione
dalle coordinate del piano di proiezione a coordinate del viewport
a partire dagli estremi del viewport passati come parametri.
xs, ys: coordinate del piano di proiezione; xmonitor, ymonitor:
coordinate del viewport; wxmax, wymax, wxmin, wymin: estremi
della finestra di visuale in cood=rdinate del piano di proiezione;
xmax, ymax: coordinate del vertice in basso a destra del monitor
(ad esempio 1024 e 768 per risoluzione VGA), assumendo 0,0 lo
coordinate del vertice in alto a sinistra.
La regola di trasformazione e':
xmonitor = xmax*(xs-wxmin)/(wxmax-wxmin) \equiv scalex*xs-offsetx
ymonitor = ymax*(ys-wymin)/(wymax-wymin) = scaley*ys-offsety}
 var wx, wy, scalex, scaley, offsetx, offsety: real;
BEGIN {of procedure window}
  if wxmin>wxmax then {scambiamo wxmin e wxmax}
 begin
   wx:=wxmax;
   wxmax:=wxmin;
   wxmin:=wx
  end;
  if wymin>wymax then {scambiamo wymin e wymax}
  begin
    wy:=wymax;
   wymax:=wymin;
```

```
wymin:=wy
end;
```

```
{determiniamo i parametri della trasformazione dal piano di
proiezione al monitor}
   scalex:=xmax/(wxmax-wxmin);
   offsetx:=xmax*wxmin/(wxmax-wxmin);
   scaley=ymax/(wymax-wymin);
   offsety:=ymax*wymin/(wymax-wymin);
```

```
END; {of procedure window}
```

```
PROCEDURE TRANSFORM (var x,y: real);
{Esegue la trasformazione dalle coordinate del piano di proiezione
ad opportune coordinate del viewport.
xs, ys: coordinate del piano di proiezione; xmonitor, ymonitor:
coordinate del viewport.
La regola di trasformazione e':
xmonitor = xmax*(xs-wxmin)/(wxmax-wxmin) = scalex*xs-offsetx
ymonitor = ymax*(ys-wymin)/(wymax-wymin) = scaley*ys-offsety
Qui scriviamo x, y per le coordinate xs, ys prima della trasformazione
ed anche per le coordinate xmonitor, ymonitor dopo la trasformazione.}
```

BEGIN {of procedure transform}

```
x:=scalex*x-offsetx;
y:=scaley*y-offsety;
{Queste coordinate sono float, ma prima di disegnare dobbiamo
arrotondarle all'intero più vicino se la routine
di tracciamento delle linee richiede la posizione
dei pixel del monitor con indici interi: questa è
l'ipotesi che faremo nella procedure DRAW responsabile
del disegno.
In caso contrario basta omettere le prossime due istruzioni.}
x:=round(x);
y:=round(y);
```

END; {of procedure transform}

PROCEDURE INTERSECT (var x1in,y1in,x1fin,y1fin,x2in,y2in,x2fin,y2fin: real,

```
184
```

#### 6.7. APPLICAZIONE: RIMOZIONE DI LINEE NASCOSTE IN GRAFICI 3D 185

```
x1,y1,x2,y2,yprofmax1, yprofmax2, yprofmin1, yprofmin2: real);
{Calcola i punti di intersezione fra tre segmenti non verticali,
il secondo ed il terzo dei quali si possono pensare come i due
profili, e la porzione del primo segmento da omettere dal disegno
perché nascosta.
Il parametro interno intersmax è true se il profilo massimo
viene intersecato in un punto interno, e analogamente per intersmin
per quanto concerne il profilo minimo.
Il parametro invisible è true se il segmento è completamente
nascosto.
Pertanto questa procedura restituisce 5 possibili casi determinati
da variabili globali boolean
che INTERSECT modifica, come segue:
  fullyvisible=true: si deve disegnare l'intero segmento (x1,y1)-(x2,y2);
  invisible=true: non si deve disegnare niente (questa variabile
  non occorre quindi restituirla al chiamante;
  drawswgment1=true: si deve disegnare la parte iniziale del
  segmento, che viene restituita da questa procedura
  col nome (x1in,y1in)-(x1fin,y1fin)
    (in questo caso x1in=x1 e y1in=y1, ma preferiamo cambiargli
   nome per gestire la routine di disegno in maniera
   più simmetrica);
  drawswgment2=true: si deve disegnare la parte finale del segmento,
  chiamata (x2in,y2in)-(x2fin,y2fin); anche qui, x2fin=x2
  e y2fin=y2.
Se drawsegment1=false o drawsegment2=false non si deve disegnare
la parte corrispondente del segmento.}
{In questa procedura possiamo supporre che i segmenti abbiano
estremi allineati verticalmente, quindi comincino tutti all'ascissa
x1 e finiscano a x2.}
BEGIN {of procedure intersect}
{Inizializzazione:}
  var xintmax, xintmin, yintmax, yintmin: real;
 pendprofmax, pendprofmin, pendenza, pendenzaprof, yprof1,
yprof2, temp: real;
 var intersmax, intersmin: boolean; fullyvisible:=false;
  invisible:=true;
  intersmax:=true;
  intersmin:=true;
  drawswgment1:=false;
```

```
drawswgment2:=false;
{Vogliamo calcolare il punto di intersezione delle rette che
contengono i segmenti (x1,y1)-(x2,y2) e (x1,yprof1)-(x2,yprof2)}
{Riordiniamo le ascisse in ordine crescente se necessario:}
  if x2>x1 then begin
    temp:=x1; x1:=x2; x2:=temp
  end;
  if x2>x1 then begin
   temp:=x1; x1:=x2; x2:=temp
  end;
{Per i nostri scopi possiamo assumere che le due rette non siano
verticali.
Calcoliamo prima la pendenza delle due rette:}
  pendenza:=(y2-y1)/(x2-x1);
  pendprofmax:=(yprofmax2-yprofmax1)/(x2-x1);
 pendprofmin:=(yprofmin2-yprofmin1)/(x2-x1);
  {Trattiamo prima il caso di totale visibilità o di invisibilità.}
  if ((y1>=yprofmax1) and (y2>=yprofmax2)) then
  begin
   fullyvisible:=true; intersmax:=false
  end;
  if ((y1<=yprofmin1) and (y2<=yprofmin2)) then
  begin
    fullyvisible:=true; intersmin:=false
  end:
  {In questi casi il segmento giace sopra il profilo massimo
  o sotto quello minimo, quindi è tutto visibile. Se
  coincide con uno dei profili allora non ci sono
  punti di intersezione interni.}
  if (y1<yprofmax1) and (y1>yprofmin1) and (y2<yprofmax2) and
(y2<yprofmin2) then
 begin
    {In questo caso il segmento è invisibile.}
    invisible:=true;
   return
    end;
{Osserviamo che il codice precedente ha già trattato il caso
di segmento parallelo ad uno dei profili, e più in generale
```

186

```
6.7. APPLICAZIONE: RIMOZIONE DI LINEE NASCOSTE IN GRAFICI 3D 187
di segmenti completamente visibili o invisibili.}
   {Calcoliamo ora l'intersezione col profilo massimo:}
  yprof1:=yprofmax1;
  yprof2:=yprofmax2;
  pendenzaprof:=pendprofmax;
  {Ora possiamo assumere che il segmento non sia parallelo
  al profilo e che sia parzialmente visibile.
  Trattiamo allora il caso di visibilità parziale:
  risolviamo il sistema lineare delle due rette.
  Il punto di intersezione verifica le equazioni
  yinters =x1 + pendenza * (xinters - x1)
  yinters =x1 + pendenzaprof * (xinters - x1)
  quindi la soluzione xinters si ottiene da
  x1+pendenza*(xinters-x1)=x1+pendenzaprof*(xinters-x1)
  ossia
      xinters =(x1*(1-pendenzaprof)
              - x1*(1-pendenza))/(pendenza-pendenzaprof) .
  Scriviamo xintmax o xintmin invece di xinters
  a seconda che il profilo sia quello massimo
  o quello minimo.}
  xintmax:=(x1*(1-pendenzaprof)
-x1*(1-pendenza))/(pendenza-pendenzaprof);
  yintmax:=x1+pendenza*(xintmax-x1);
  if (x1<=xintmax) and (xintmax<=x2) then xintersmax:=true;
{Calcoliamo infine l'intersezione col profilo minimo:}
  yprof1:=yprofmin1;
  yprof2:=yprofmin2;
 pendenzaprof:=pendprofmin;
  xintmin:=(x1*(1-pendenzaprof)
-x1*(1-pendenza))/(pendenza-pendenzaprof);
  yintmin:=x1+pendenza*(xintmin-x1);
  if (x1<=xintmin) and (xintminx<=x2) then xintersmin:=true;
{A questo punto almeno uno dei profili viene intersecato. Calcoliamo
allora l'estremo sinistro (x1in,y1in) e destro (x1fin,y1fin)
della prima parte di segmento da disegnare e l'estremo sinistro
(x2in,y2in) e destro (x2fin,y2fin) della seconda parte.}
  if (intersmax=true) then begin
                                   {Intersezione con prof max.}
    if (y1>=yprofmax1) then begin
```

{bisogna disegnare almeno la prima parte del segmento:}

```
x1in:=x1;
      y1in:=y1;
      x1fin:=xintmax;
      y1fin:=yintmax;
      drawsegment1:=true
      end
    else begin
      {bisogna disegnare almeno la seconda parte del segmento:}
      x2in:=xintmax;
      y2in:=yintmax;
      x2fin:=x2;
      y2fin:=y2;
      drawsegment2:=true
      end
    end;
  end;
  if (intersmin=true) then begin
                                    {Intersezione con prof min.}
    if (y1<=yprofmin1) then begin
      x1in:=x1;
      y1in:=y1;
      x1fin:=xintmin;
      y1fin:=yintmin;
      drawsegment1:=true
      end
    else begin
      x2in:=xintmin;
      y2in:=yintmin;
      x2fin:=x2;
      y2fin:=y2;
      drawsegment2:=true
      end
    end;
           end;
   {Caso di una sola intersezione:}
  if (intersmax=false) then begin
                                   {Niente intersezione con
prof max, solo con prof min}
    if (y1<yprofmin1) then begin
      {bisogna disegnare solo la prima parte del segmento:}
      x1in:=x1;
      y1in:=y1;
      x1fin:=xintmin;
      y1fin:=yintmin;
```

188

```
6.7. APPLICAZIONE: RIMOZIONE DI LINEE NASCOSTE IN GRAFICI 3D 189
      drawsegment1:=true;
      drawsegment2:=false
      end
    else begin
      {bisogna disegnare solo la seconda parte del segmento:}
      x2in:=xintmin;
      y2in:=yintmin;
      x2fin:=x2;
      y2fin:=y2;
      drawsegment1:=false;
      drawsegment2:=true
      end
    end
  end:
  END; {of procedure intersect}
PROCEDURE PERSPECTIVE (var xout, yout: real; x,y,z: real);
{Esegue la trasformazione prospettica in prospettiva obliqua
dalle coordinate 3D x,y,z a coordinate xout, yout del viewplane.}
BEGIN {of procedure perspective}
  xout = -x*tau1 + y*tau2;
  yout = -x*tau3 - y*tau4 + z*tau5;
END; {of procedure perspective}
BEGIN {of procedure draw}
{num è il numero di quadratini in cui è suddiviso ogni lato
della griglia di base.
Gli array x e y contengolo le ascisse e le ordinate dei nodi
della griglia di bae; l'array bidimensionale fun contiene i
valori della funzione da disegnare sui nodi di questa griglia.
L'array bidimensionale punto contiene i punti bidimensionali
(valori x' e y') sul piano di proiezione dei trasformati prospettici
di una riga dei nodi del wireframe 3D. Rammentiamo che, prima
di disegnarli, questi valori devono essere adattati alla scelta
di coordinate nel viewport, tramite la procedura TRANSFORM,
che utilizza gli estremi di viewport calcolati dalla procedura
WINDOW.
```

L'array puntoprec ha le stesse dimensioni di punto, e serve per memorizzare la riga precedente. Questa memorizzazione è necessaria quando si devono tracciare segmenti trasversali alla poligonale corrente nel corso del disegno a maglie spiegato in (6.7.1): i punti iniziali di questi segmenti trasversali sono infatti stati già aggiornati nel corso del ciclo di aggiornamento dell'array unidimensionale punto. Gli array profmax, profmin sono unidimensionali e memorizzano i valori minimo e massimo dei profili della zona invisibile. I corrispondenti array tempmax, tempmin memorizzano la versione di profmax e profmin che si riferisce alle prime due poligonali x=costante: in seguito vengono travasati in profmax e profmin rispettando gli indici in maniera che la capienza di profmax e profmin si raddoppi e tempmax e tempmin occupino la seconda metà di profmax e profmin (che corrisponde alla metà di sinistra se si va all'indietro in profondità, come illustrato nella Figura 10). Da quel momento in poi si ha bisogno solo degli array profmax e profmin per memorizzare i profili, ma bisogna traslare gli indici di un passo ogni volta che si passa alla poligonale un passo più indietro. axmin, aymin, axmax, aymax sono le coordinate sul piano z=0 dei vertici estremi della griglia di base.} {Scelta degli estremi del viewport, in conformità alla Sezione 6.7.2. Come si vede in Figura 10, le ascisse e le ordinate del viewplane diminuiscono all'aumentare di x e non dipendono da z, mentre le ordinate del viewplane diminuiscono all'aumentare di x ma aumentano all'aumentare di y e z. Nel viewplane l'asse delle ordinate è orientato verso il basso; quindi l'ascissa del punto in alto a sinistra del viewplane corrisponde ad ascissa massima della griglia axmax e ordinata minima axmin, mentre la sua ordinata corrisponde a axmax, aymax e al valore minimo della funzione zmin. Per il punto in basso a destra naturalmente succede l'opposto.} x1 = -axmax\*tau1 + aymin\*tau2; y1 = -axmax\*tau3 - aymax\*tau4 + zmin\*tau5; x2 = -axmin\*tau1 + aymax\*tau2;

y2 = -axmin\*tau3 - aymin\*tau4 + zmax\*tau5;

window (x1, y1, x2, y2);

```
190
```

{Tracciamo la prima curva x=costante, a partire dalla curva più vicina al punto di osservazione (ossia x=num). Per le due prime curve x=costante e y=costante non si devono effettuare rimozioni di linee nascoste, e quindi l'ordine di tracciamento dei segmenti e di aggiornamento dei profili massimo e minimo è inessenziale: comunque seguiamo l'ordine di tracciamento dal davanti all'indietro, ossia dall'indice num all'indice 0, come osservato più sopra.}

```
for iy:=num downto 0 do
begin
  xx:=x[num]; yy:=y[iy]; zz:=fun[num,iy];
  perspective(xout, yout, xx, yy, zz);
  punto[iy,0]:=xout ;
  punto[iy,1]:=yout;
  {Inizializzazione dei profili: gli array tempmax e tempmin
  memorizzano i profili massimo e minimo delle prime due
  poligonali x=costante, ossia x=num e x=num-1.}
  tempmax[iy,0]:=punto[iy,0];
  tempmax[iy,1]:=punto[iy,1];
  tempmin[iy,0]:=punto[iy,0];
  tempmin[iy,1]:=punto[iy,1]
end;
for iy:=num downto 1 do
begin
  {Per disegnare i segmenti dobbiamo adattare le coordinate
  del piano di proiezione a coordinate di viewport,
  ossia di monitor, chiamando la procedura TRANSFORM.}
  x1:=punto[iy,0];
  y1:=punto[iy,1];
  x2:=punto[iy-1,0];
  y2:=punto[iy-1,1];
  transform(x1,y1);
  transform(x2,y2);
  line(x1,y1,x2,y2);
end;
{Tracciamo la seconda curva x=costante(x=num-1):}
for iy:=num downto 0 do
begin
  {Memorizziamo in puntoprec i valori correnti dell'array punto
  prima di aggiornarlo: pertanto puntoprec contiene i valori
```

```
di punto al passo precedente di aggiornamento. Questa
  memorizzazione è necessaria per tracciare i segmenti
  trasversali alla poligonale corrente: i punti iniziali di
  questi segmenti trasversali sono stati infatti modificati
  nel corso del ciclo di aggiornamento dell'array punto.}
  puntoprec[iy,0]:=punto[iy,0];
  puntoprec[iy,1]:=punto[iy,1];
  xx:=x[num-1]; yy:=y[iy]; zz:=fun[num-1,iy];
  perspective(xout, yout, xx, yy, zz);
  punto[iy,0]:=xout ;
  punto[iy,1]:=yout
end; for iy:=0 to num-1 do
begin
{Le posizioni dei due array si sfalsano di un passo ogni volta
che l'indice della x aumenta di una unità, e quindi nel passaggio
dalla poligonale x=num a quella x=num-1 occorre far slittare
di un passo gli array perché gli indici siano consistenti.}
  tempmax[iy,0]:=tempmax[iy+1,0];
  tempmax[iy,1]:=tempmax[iy+1,1];
  tempmin[iy,0]:=tempmin[iy+1,0];
  tempmin[iy,1]:=tempmin[iy+1,1]
end;
for iy:=num downto 1 do
        {Disegno a maglie, come spiegato in (6.7.1):}
begin
  {tratto x=costante}
  x1:=punto[iy,0];
  y1:=punto[iy,1];
  y2:=punto[iy-1,1];
  x2:=punto[iy-1,0];
  transform(x1,y1);
  transform(x2,y2);
  line(x1,y1,x2,y2);
  {tratto y=costante: questo è il segmento trasversale}
  x1:=punto[iy-1,0];
  y1:=punto[iy-1,1];
  {è qui che abbiamo bisogno del valore di punto alla riga precedente
  all'ultimo aggiornamento: }
                           y2:=puntoprec[iy-1,1];
  x2:=puntoprec[iy-1,0];
  transform(x1,y1);
  transform(x2,y2);
  line(x1,y1,x2,y2);
end;
```

192

```
{Aggiorniamo i profili massimo e minimo:}
for iy:=num-1 downto 0 do
begin
  if punto[iy,1]>tempmax[iy,1] then
  begin
    tempmax[iy,0]:=punto[iy,0];
    tempmax[iy,1]:=punto[iy,1]
  end;
  if punto[iy,1]<tempmin[iy,1] then
  begin
    tempmin[iy,0]:=punto[iy,0];
    tempmin[iy,1]:=punto[iy,1]
  end
end;
{Tracciamo la prima curva y=costante(y=num):}
for ix:=num downto 0 do
begin
  x:=x[ix]; yy:=y[num]; zz:=fun[ix,num];
  perspective(xout, yout, xx, yy, zz);
 punto[ix,0]:=xout ;
 punto[ix,1]:=yout;
  profmax[ix,0]:=punto[ix,0];
  profmax[ix,1]:=punto[ix,1];
  profmin[ix,0]:=punto[ix,0];
  profmin[ix,1]:=punto[ix,1]
end;
for ix:=num downto 1 do
begin
  x1:=punto[ix,0];
  y1:=punto[ix,1];
  x2:=punto[ix-1,0];
  y2:=punto[ix-1,1];
  transform(x1,y1);
  transform(x2,y2);
  line(x1,y1,x2,y2)
end;
```

{Tracciamo la seconda curva y=costante(y=num-1):}

```
194
               6. TRASFORMAZIONI PROSPETTICHE
for ix:=num downto 0 do
begin
  puntoprec[ix,0]:=punto[ix,0];
  puntoprec[ix,1]:=punto[ix,1];
  x:=x[ix]; yy:=y[num-1]; zz:=fun[ix,num-1];
  perspective(xout, yout, xx, yy, zz);
  punto[ix,0]:=xout;
  punto[ix,1]:=yout
end;
for ix:=0 to num-1 do
begin
{Le posizioni dei due array si sfalsano di un passo ogni volta
che l'indice della y aumenta di una unità, e quindi nel passaggio
dalla poligonale y=num a quella y=num-1 occorre far slittare
di un passo gli array perché gli indici siano consistenti.}
  profmax[ix,0]:=profmax[ix+1,0];
  profmax[ix,1]:=profmax[ix+1,1];
  profmin[ix,0]:=profmin[ix+1,0];
  profmin[ix,1]:=profmin[ix+1,1]
end;
for ix:=num-1 downto 1 do
begin
  {tratto y=costante}
  x1:=punto[ix,0];
  y1:=punto[ix,1];
  x2:=punto[ix-1,0];
  y2:=punto[ix-1,1];
  transform(x1,y1);
  transform(x2,y2);
  line(x1,y1,x2,y2);
  {tratto x=costante}
  x1:=punto[ix-1,0];
  y1:=punto[ix-1,1];
  x2:=puntoprec[ix-1,0];
  y2:=puntoprec[ix-1,1];
  transform(x1,y1);
  transform(x2,y2);
  line(x1,y1,x2,y2);
end;
{Aggiorniamo i profili massimo e minimo:}
```

```
for ix:=num-1 downto 0 do
```

```
6.7. APPLICAZIONE: RIMOZIONE DI LINEE NASCOSTE IN GRAFICI 3D 195
```

```
begin
  if punto[ix,1]>profmax[ix,1] then
  begin
   profmax[ix,0]:=punto[ix,0];
   profmax[ix,1]:=punto[ix,1]
  end;
  if punto[ix,1]<profmin[ix,1] then
  begin
   profmin[ix,0]:=punto[ix,0];
    profmin[ix,1]:=punto[ix,1]
  end
end;
for ix:=1 to num-1 do
begin
{Gli array tempmax, tempmin memorizzavano i profili massimo
e minimo delle prime due poligonali x=costante: ora li travasiamo
in profmax e profmin in maniera da raddoppiare la capienza di
profmax e profmin e disporre i dati di tempmax e tempmin nella
seconda metà di profmax e profmin (che corrisponde alla metà
di sinistra se si va all'indietro in profondità, come illustrato
nella Figura 10).}
 profmax[num+(num-ix)-1,0]:=tempmax[ix-1,0];
 profmax[num+(num-ix)-1,1]:=tempmax[ix-1,1];
 profmin[num+(num-ix)-1,0]:=tempmin[ix-1,0];
 profmin[num+(num-ix)-1,1]:=tempmin[ix-1,1]
end;
{Tracciamo alternativamente un segmento di curva y=cost e x=cost
a partire dalla seconda poligonale in poi, come spiegato in
(6.7.1):
for iy:=num-2 downto 0 do
begin
 for ix:=num downto 0 do
 begin
```

```
puntoprec[ix,0]:=punto[ix,0];
puntoprec[ix,1]:=punto[ix,1];
x:=x[ix]; yy:=y[iy]; zz:=fun[ix,iy];
perspective(xout, yout, xx, yy, zz);
punto[ix,0]:=xout;
punto[ix,1]:=yout
```

```
end;
```

```
{Poichè facciamo avanzare le y, i num+1 indici dei profili massimo
e minimo sono posizioni consecutive in una sequenza di 2*num+1
posizioni che corrispondono alle verticali dei nodi della griglia
vista in prospettiva, metà per i nodi del lato più vicino al
variare di x e l'altra metà al variare di y. Queste num posizioni
si sfalsano di un passo ogni volta che l'indice della y aumenta
di una unità, e quindi ogni volta occorre far slittare di un
passo l'array perché gli indici siano consistenti. Nella scansione
di una poligonale si usano solo num+1 di questi indici.}
  for ix:=0 to (2*num-1) do
  begin
    profmax[ix,0]:=profmax[ix+1,0];
    profmax[ix,1]:=profmax[ix+1,1];
    profmin[ix,0]:=profmin[ix+1,0];
    profmin[ix,1]:=profmin[ix+1,1];
  end:
  for ix:=num-1 downto 1 do
  begin
  {tratto y=costante}
    x1:=punto[ix,0];
    y1:=punto[ix,1];
    x2:=punto[ix-1,0];
    y2:=punto[ix-1,1];
    yprofmax1:=profmax[ix,1];
    yprofmax2:=profmax[ix-1,1];
    yprofmin1:=profmin[ix,1];
    yprofmin2:=profmin[ix-1,1];
    intersect (x1in,y1in,x1fin,y1fin,x2in,y2in,x2fin,y2fin,
    x1,y1,x2,y2,yprofmax1,yprofmax2,yprofmin1,yprofmin2);
    if fullyvisible then
    begin
      transform(x1,y1);
      transform(x2,y2);
      line(x1,y1,x2,y2)
    end
    else
    begin
      if drawsegment1 then
      begin
        transform(x1in,y1in);
        transform(x2in,y2in);
        line(x1in,y1in,x2in,y2in)
```

196

```
end;
if drawsegment2 then
begin
    transform(x1fin,y1fin);
    transform(x2fin,y2fin);
    line(x1fin,y1fin,x2fin,y2fin)
end
end;
```

{Aggiorniamo i profili massimo e minimo in corrispondenza al punto finale a cui siamo arrivati (quello con indice ix-1).}

```
if punto[ix-1,1]>=profmax[ix-1,1] then
begin
    profmax[ix-1,0]:=punto[ix-1,0];
    profmax[ix-1,1]:=punto[ix-1,1]
end;
if punto[ix-1,1]<=profmin[ix-1,1] then
begin
    profmin[ix-1,0]:=punto[ix-1,0];
    profmin[ix-1,1]:=punto[ix-1,1]
end</pre>
```

```
END; {of procedure draw}
```

# **VOLUME 2**

# ALGORITMI CLASSICI IN COMPUTER GRAPHICS

### **QUINTA PARTE**

### GENERALITA' SULLE STRUTTURE ED I PRINCIPI DELLA COMPUTER GRAPHICS

## OCTREES

### **QUADTREES E OCTREES**

Gli octree ed i quadtree sono strutture ad albero i cui vertici (detti anche nodi) sono celle ottenute da partizioni binarie successive (cioè suddivisioni progressive di passo due), rispettivamente di un cubo o di un quadrato, entro il quale si trova una figura, rispettivamente solida o piana, che si vuole approssimare come unione di celle della partizione.

Da questi alberi ricaviamo una parametrizzazione delle celle della partizione in cui si trova l'oggetto. Si ottiene un quadtree suddividendo iterativamente a metà in entrambe le dimensioni un quadrato che contiene la figura piana che vogliamo rappresentare: ciascuna suddivisione dà luogo a quattro celle. Ogni cella può trovarsi in varie condizioni di inclusione rispetto alla figura: può essere piena (completamente interna all oggetto), parzialmente piena oppure vuota (completamente esterna all oggetto). Le successive suddivisioni organizzano le varie celle come vertici di un *albero genealogico*. I vertici dell'albero corrispondono alle celle; i suoi spigoli uniscono ciascuna cella con le quattro celle ottenute dalla sua suddivisione, o viceversa. Possiamo fissare una soglia, cioè una percentuale prefissata, e dichiarare piena una cella che è interna all oggetto per una parte della propria area superiore a tale percentuale, e vuota una cella esterna all'oggetto per un'area inferiore a tale percentuale, per enfatizzare, abbiamo scelto una percentuale molto bassa, dell'ordine del 15%.

Una cella parzialmente piena viene ulteriormente suddivisa in maniera ricorsiva. Questa suddivisione continua fino a che tutte le celle sono omogenee (ognuna piena o vuota entro la soglia prefissata) o fino a che si raggiunge un prefissato numero di iterazioni.

Gli octree costituiscono la analoga rappresentazione ad albero per figure tridimensionali: ogni suddivisione avviene per ciascuna delle tre dimensioni e dà luogo ad otto celle.

Nella figura di sinistra è rappresentata la griglia delle celle relativa alla partizione più fine. Nella figura di destra è visualizzata la stessa immagine con la suddivisione delle celle del quadtree: le celle ora non appartengono più tutte alla partizione più fine, perché per alcune zone, a seconda della figura che si vuole rappresentare, il processo di suddivisione si ferma prima.



Le suddivisioni successive vengono rappresentate in termini di un albero i cui vertici rappresentano le varie celle di ogni generazione. Quelli parzialmente pieni sono qui indicati con una P (*partially*), quelli pieni con una F (*full*) ed infine quelli vuoti con una E (*empty*). Inoltre alle quattro celle create in ogni ripartizione è associato un numero da zero fino a tre, nellàordine presentato nella figura seguente.



La figura qui sopra mostra indica l'albero associato alla ripartizione della figura iniziale. I numeri sui segmenti indicano quale delle quattro celle della suddivisione stiamo raggiungendo.

Nel caso degli octree, ogni ripartizione crea otto celle tridimensionali, che vengono numerate da 0 a 7 nell'ordine visualizzato nella figura seguente, che rappresenta due partizioni successive (nella seconda viene ripartita solo la cella frontale in alto a destra).



#### OPERAZIONI BOOLEANE (CONSTRUCTIVE SOLIDE GEOMETRY, CSG) CON OCTREES

Con gli octree ed i quadtree è possibile eseguire varie operazioni. Ad esempio, con l'uso di quadtree e octree, si possono eseguire operazioni booleane: si può calcolare l'unione o l'intersezione delle rappresentazioni di due oggetti in termini di celle, direttamente sui rispettivi alberi S e T, come mostrato in figura. Per calcolare l'unione, osserviamo che l'unione di una cella parzialmente piena P (ossia che interseca parzialmente la figura rappresentata dal quadtree o l'octree) con una vuota E (ossia una cella disgiunta dalla figura) consiste di una cella parzialmente piena. Invece una cella vuota E unita ad una parte piena F dà come risultato una cella piena, e così via. Viceversa, l'intersezione di qualsiasi cella con una vuota ne dà una vuota. La figura sottostante rappresenta due figure S e T decomposte in celle, sulle quali vengono effettuate le operazioni booleane di unione e di intersezione. I corrispondenti quadtree sono illustrati in fondo alla pagina.













### ALGORITMO DI RICERCA DELLA CELLA ADIACENTE IN UN OCTREE

Un'altra operazione importante nei quadtrees e negli octrees consiste, per ogni vertice dell'albero, nel trovare, fra gli altri vertici, quale sia quello che corrisponde ad una cella adiacente a quella associata al vertice considerato, cioè localizzare un vertice "adiacente" (nel senso della decomposizione in celle del piano o dello spazio) al vertice originale (con cui cioe' condivide una faccia, un lato o un vertice) e di generazione non superiore (se la generazione e' superiore, le celle adiacenti sono piu' d'una!)

Un nodo di un quadtree (ossia una cella nella geometria bidimensionale) ha le celle adiacenti in otto possibili direzioni. Le sue celle vicine a nord, sud, est e ovest sono vicine lungo un lato comune, mentre quelle vicine a nord-ovest, nord-est, sud-ovest e sud-est sono vicine lungo un vertice comune. Invece un nodo di un octree (ossia una cella tridimensionale) ha vicini in 26 possibili direzioni: 6 vicini lungo una faccia, 12 vicini lungo un lato e 8 vicini lungo un vertice. Samet ha descritto un algoritmo per trovare il vicino del vertice lungo una direzione data. Si considerano due celle adiacenti, ed il vertice associato ad una di esse (quella di generazione piu' grande, quindi la piu' piccola in grandezza). Si vuol trovare quale e' il vertice associato all'altra cella. Si parte dal vertice originale e si risale il quadtree (o octree) fino al primo antenato comune fra le due celle. Poi si discende l'albero fino a trovare il vertice corrispondente alla cella adiacente.

Per questo bisogna risolvere due problemi: trovare l'antenato comune e determinare quale dei suoi discendenti è il vicino.

Per prima cosa spieghiamo come si determina l'antenato comune. Il caso più semplice è quando si vuol trovare il vicino del vertice di un octree nella direzione di una delle sue facce: L (left), R (right), U (up), D (down), F (front) oppure B (back). Quando risaliamo l'albero a partire dal vertice originale, l'antenato comune e' il primo vertice che non è raggiunto da un figlio sul lato del vertice. Per esempio se cerchiamo un vicino sul lato sinistro L, allora il primo antenato comune è il primo vertice che non è raggiunto da un figlio in posizione LUF, LUB, LDF oppure LDB. Questo perchè un vertice che è stato raggiunto da uno di questi figli non può avere alcun figlio che è a sinistra del vertice originale.

E' opportuno spiegare qual e' il modo a tal fine piu' opportuno di codificare le direzioni con cui ci si muove da un vertice figlio al padre o viceversa: cioe' codificare gli spigoli. Nel caso di un quadtree, ci sono quattro figli, denotati dai numeri 0, 1, 2, 3, come in figura:



bit meno significativo

Scriviamo questi quattro numeri nel sistema binario. Allora i due figli in basso sono codificati dai numeri 00 e 01, quelli in alto da 10 e 11. In altre parole, il bit piu' alto vale 0 sulla riga bassa, 1 sulla riga alta; il bit basso vale 0 sulla colonna di sinistra, 1 su quella di destra. Allora, raggiungere il progenitore partendo dal suo lato basso significa percorrere un segmento codificato da un numero binario la cui prima cifra e' 0; raggiungerlo dal lato di sinistra significa percorrere un segmento il cui codice binario la cui ultima cifra e' 0, e cosi' via. Analogamente, per un octree, i segmenti hanno codice binario di tre cifre: quelli che portano a figli sul lato posteriore hanno bit piu' significativo uguale a zero (si tratta dei numeri 0=000, 1=001, 2=010, 3=011), quelli del lato sinistro (0=000, 2=010, 4=100, 6=110) hanno bit meno significativo 0, quelli sul lato basso (0=000, 1=001, 4=100, 5= 101) hanno bit centrale uguale a zero, e cosi' via.



Una volta trovato l'antenato comune, per determinare il vertice corrispondente alla cella adiacente nella direzione assegnata e' sufficiente discendere l'albero in maniera speculare a come lo si era risalito, nel senso della riflessione speculare riflesso al bordo comune (nel caso dell'esempio piu' sopra, questo significa nello scegliere la direzione R ogni volta che ci si era mossi in direzione L, e viceversa).

Cosa vuol dire riflesso speculare in termini della precedente codifica binaria? Vuol dire, ad esempio in un octree, che per riflettere specularmente la destra nella sinistra dobbiamo cambiare il valore del bit meno significativo, da 1 a 0 (passando da 1,3,5,7 a 0,2,4,6, cioe' da 001, 011, 011, 111 a 000, 010, 100, 110).

Se la cella vicina è più grande della cella originale, cioe' se il vertice corrispondente appartiene ad una generazione piu' antica di quello originale, allora il percorso riflesso discendente termina prima (cioe' arriva ad un vertice terminale in meno passi di quelli fatti in salita: ha una lunghezza inferiore a quello ascendente).

Un metodo simile può essere usato per trovare un vicino del vertice di un quadtree nella direzione di uno dei suoi lati. Ad esempio, per trovare il vicino in direzione Nord del vertice A della figura sottostante si parte da A, che rappresenta il figlio nord-ovest, e seguiamo il percorso descritto dalla linea spessa nella figura. Saliamo muovendoci da NW verso il suo genitore, ancora da NW verso il nonno, e poi da SW al suo bisnonno, ovvero la radice, fermandoci perchè ci siamo avvicinati ad esso da un vertice S invece che che da un vertice N. Dopodichè seguiamo il percorso speculare rispetto al lato basso (riflesso rispetto bordo N-S), al figlio NW della radice, poi al figlio SW del vertice cosi' raggiunto, che e' un vertice terminale. Pertanto qui il percorso finisce, ed abbiamo trovato il vertice cercato.





### **DEFINIZIONE ED ESEMPI DI COERENZA**

Con il termine coerenza s'intende il grado con cui parti diverse della scena o della sua proiezione prospettica presentano proprietà simili. Le scene di solito si compongono di oggetti all'interno dei quali le proprietà (colore, profondità, tessitura, ecc.) si mantengono quasi costanti.

Il concetto di coerenza si adopera per riutilizzare gli stessi calcoli, effettuati per una parte dell'ambiente o di un'immagine, ad altre parti adiacenti, apportando le eventuali modifiche.

I tipici concetti di coerenza sono i seguenti:

**Coerenza d'Oggetto:** se una scena è composta di oggetti le cui proiezioni sul viewport sono completamente separate da quelle degli altri, i confronti di profondità per determinare la visibilità devono essere effettuati solo fra le parti di ciascun oggetto e non tra un oggetto e gli altri.

**Coerenza di Faccia:** le proprietà lungo una faccia variano poco, e quindi per ogni riga di pixel che attraversa quella faccia è possibile riutilizzare i calcoli per l'illuminazione già fatti per le righe precedenti, nel tratto adiacente.

**Coerenza di Spigolo:** uno spigolo cambia visibilità solo nei punti in cui passa dietro ad un altro spigolo o ad una faccia visibile.

**Coerenza di Spigolo Implicito:** tra due facce che si compenetrano si crea uno spigolo implicito d'intersezione, determinato dai due punti estremi dell'intersezione fra le due facce.

**Coerenza di Riga di Scansione:** l'insieme dei valori di illuminazione dei segmenti visibili in una riga di scansione differisce di poco da quella dei segmenti corrispondenti della precedente.

**Coerenza di Area:** nella maggior parte dei casi un gruppo di pixel adiacenti sul viewport è coperto dalla proiezione di una stessa faccia visibile (caso particolare è la coerenza di tratto visibile, che si riferisce alla copertura di un segmento visibile lungo una riga di scansione).

**Coerenza di Profondità:** superficie o facce diverse della scena che si proiettano sullo stesso pixel del viewport normalmente sono ben separate in profondità, mentre all'interno della stessa superficie la profondità varia di poco.

**Coerenza di Fotogramma (Frame):** immagini della stessa scena in due istanti consecutivi, sono normalmente quasi identiche tranne che per piccoli cambiamenti negli oggetti e nel punto di osservazione.

**Coerenza di orientamento tridimensionale:** la direzione della normale a ciascuna superficie della scena cambia di poco da un pixel a quello adiacente a meno che non si oltrepassi uno spigolo di bordo. In particolare, se la scena è modellata con poligoni piani, le normali restano costanti fino a che non si arriva agli spigoli. Le normali sono essenziali per il calcolo dell'ombreggiatura, ed anche per un altro procedimento preliminare al

# SESTA PARTE

# **RIMOZIONE DELLE AREE NASCOSTE**
## METODI DI RIMOZIONE DI AREE NASCOSTE A PRECISIONE DI IMMAGINE

Questi metodi di rimozione delle aree nascoste operano nel modo seguente. Pper ogni pixel nella parte del piano di proiezione dove si vuole rendere l'immagine prospettica, si determina se e quale degli *n* oggetti della scena è visibile. Per far questo, per ogni pixel nell'immagine si trova quale sia l'oggetto più vicino all'osservatore che viene intersecato dal raggio uscente dalla posizione dell'osservatore e passante per il centro del pixel; il pixel viene poi colorato col colore che quell'oggetto ha nel punto di intersezione col raggio.

Ecco lo pseudocodice:

for (ciascun pixel nell'immagine) {

determinare l'oggetto piu' vicino all'osservatore che viene intersecato dal raggio di proiezione dal punto di visuale che passa per quel pixel;,

disegnare il pixel col colore corrispondente }

Cio' richiede di trovare, per ciascun pixel, le intersezioni del corrispondente raggio di proiezione con tutti gli *n* oggetti per determinare quale è il più vicino all'osservatore. Se la risoluzione di immagine consiste di *p* pixels, la complesita' del calcolo è proporzionale a *np*. Si osservi che per un'immagine tipica di 1024 x 768 pixel, *p* è dell'ordine di un milione. Di solito il numero *n* di oggetti della scena e' molto minore.

## METODI DI RIMOZIONE DI AREE NASCOSTE A PRECISIONE DI OGGETTO

Questi metodi paragonano gli oggetti della scena direttamente fra loro, eliminando interamente gli oggetti o le parti di essi che non sono visibili dal punto di vista dell'osservatore. In altre parole, di ogni oggetto della scena si determinano quelle parti la cui vista non è impedita da altre parti di esso o da altri oggetti, e si disegnano queste parti nel colore appropriato. Ecco lo pseudocodice:

### for (ciascun oggetto nella scena) {

si determinano le sue parti la cui visuale dal punto di ossrervazione non e' ostruita da altre sue parti o da altri oggetti;

si disegnano queste parti visibili nel colore dell'oggetto}

Per eseguire il rendering a precisione di oggetto si deve quindi confrontare ognuno degli n oggetti con sé stesso e con gli altri oggetti, scartando le porzioni non visibili. Il tempo di calcolo quindi è proporzionale a  $n^2$ . Sebbene questo approccio potrebbe richieda meno confronti di quello a precisione di immagine se n < p (il che e' quasi sempre vero nella pratica), i suoi confronti sono di solito più complessi e richiedono più tempo, e pertanto esso è di solito più lento e più complesso da implementare.

Gli algoritmi a precisione d'immagine sono tipicamente elaborati alla risoluzione del monitor, perché determinano la visibilità per ogni pixel. Gli algoritmi a precisione d'oggetto sono invece elaborati alla precisione con cui ogni oggetto è definito nella sua modellazione matematica, e con questa precisione ne determinano la visibilità. Dal momento che i calcoli della precisione d'oggetto sono effettuati senza riferimento ad una particolare risoluzione del monitor, devono essere seguiti da un passaggio in cui gli oggetti di cui si è determinata la visibilità sono poi disegnati sul monitor alla risoluzione di quest'ultimo. Solo quest'ultimo passo dipende dalla risoluzione del monitor, e quindi dal numero di pixel dell'immagine, e solo esso deve essere ricalcolato se si desidera cambiare la risoluzione dell'immagine (per esempio se si cambia il numero di pixel visualizzati dalla scheda grafica, oppure la scala con cui viene disegnata l'immagine, e quindi la sua grandezza sul monitor).

Invece il cambiamento della risoluzione o della scala in un algoritmo di rendering a precisione di immagine richiede di effettuare da capo tutto il calcolo. Ad esempio, immaginiamo di voler ingrandire un'immagine prodotta da un tale algoritmo. I calcoli della superficie visibile sono stati inizialmente sviluppati alla risoluzione originale più bassa, e devono essere ripetuti se Nell'impostazione vogliamo rivelare ulteriori dettagli. di pensiero suggerita dal campionamento, possiamo pensare agli algoritmi a precisione di oggetto come operanti sui dati dell'oggetto originale, modellato in termini di forme continue, e gli algoritmi a precisione d'immagine come operanti su dati campionati. Così gli algoritmi a precisione d'immagine sono soggetti ad aliasing nel calcolo della visibilità, mentre gli algoritmi a precisione d'oggetto no.

## RIMOZIONE DELLE FACCE POSTERIORI (BACK FACE CULLING)

Quando si rende una scena tridimensionale proiettandola sul piano di visuale, le facce posteriori degli oggetti della scena vengono coperte da quelle anteriori, e quindi non possono essere visualizzate (stiamo qui facendo l'ipotesi usuale che tutti gli oggetti siano solidi chiusi, altrimenti una faccia posteriore potrebbe essere visibile da dietro). Pertanto è opportuno rimuoverle dalla modellazione della scena prima di passarle al renderer, in maniera da evitare una mole di calcoli inutili. Questo procedimento di rimozione delle facce posteriori (*back face culling*) è estremamente semplice: è sufficiente identificare le facce posteriori, che sono quelle il cui versore normale esterno **N** forma un angolo di più di 90<sup>0</sup> con la direzione dell'osservatore, che indichiamo con **V**. Il versore **N** è fornito dal modellatore: quindi, per ogni faccia dell'oggetto, basta considerare un so punto, formare il vettore **V** e poi verificare se il prodotto scalare <**N**, **V**> ha segno positivo o negativo: nel secondo caso abbiamo una faccia posteriore, e la rimuoviamo.

# ALGORITMI DI RIMOZIONE DELLE AREE NASCOSTE TRAMITE SCANSIONE RIGA PER RIGA SUL PIANO DELL'IMMAGINE PROIETTATA

**Z-BUFFER** 

### L'ALGORITMO DI Z-BUFFER

L'algoritmo di z-buffer (o depth-buffer) a precisione di immagine, sviluppato in [E. Catmull, A Subdivision Algorithm for Computer Display of Curved Surfaces, Tesi dottorale, Computer Science Department, University of Utah, Salt Lake City, Utah, US, Dicembre 1974, è uno degli algoritmi per la determinazione delle superfici visibili più semplici da implementare sia a livello di software sia a livello di hardware. Richiede che siano disponibili due buffer di memorizzazione: un frame buffer F in cui sono memorizzati i valori dei colori, ed uno z-buffer Z, con lo stesso numero di dati, in cui per ogni pixel viene memorizzata la profondità dell'oggetto della scena tridimensionale che è visibile attraverso quel pixel. Lo z-buffer è inizializzato al valore di zero, che rappresenta la profondità del piano posteriore di clipping, ed il frame buffer è inizializzato al colore dello sfondo. Poiché per convenzione l'asse z che indica la profondità è preso con verso crescente verso l'osservatore, il più grande valore che può essere memorizzato nello z-buffer rappresenta la profondità del piano anteriore di clipping. I poligoni sono scansionati nel frame buffer seguendo un ordine arbitrario; durante questo processo, se un punto (appartenente a un poligono) che viene scansionato alle coordinate (x,y) non è più lontano dall'osservatore del punto il cui colore e la cui profondità sono attualmente memorizzati nei buffer, allora il suo colore e la sua profondità prendono il posto di quelli precedentemente memorizzati. In pratica, per ogni pixel vengono memorizzati il colore e la profondità del punto appartenente alla scena finora più vicino all'osservatore.

Non sono necessari né paragoni *object to object*, né alcun ordinamento preliminare per l'esame dei poligoni della scena. L'intero processo non è nient'altro che ricercare il più alto valore di  $Z_i$  per ogni coppia di  $\{Z_i (x,y), F_i (x,y)\}$  con x e y fissati. Lo z-buffer ed il frame buffer registrano l'informazione associata al più alto valore z (il più vicino all'osservatore) incontrato finora per ogni (x,y). Perciò, i poligoni appaiono sulla scena nell'ordine in cui vengono processati. Quando ad un dato pixel viene processato un poligono piu' vicino all'osservatore di quello che vi appariva precedentemente, il colore del nuovo poligono sostituisce quello del vecchio.

E' cruciale osservare che l'algoritmo non esplora la geometria tridimensionale delle intersezioni dei raggi di proiezioni dall'osservatore alla scena con i poligoni della scena. Si tratta invece di un algoritmo bidimensionale: ogni poligono della scena viene proiettato sul piano di proiezione, intermedio fra osservatore e scena (cioé si esegue la scan-conversion, la proiezione tramite scansione), ed in tal modo per ogni pixel si trova se la proiezione del poligono copre il pixel, cioé se il poligono sarebbe visibile attraverso quel pixel qualora la visuale non fosse eventualmente ostruita da altri poligoni più vicini all'osservatore. Il problema dell'ostruzione, cioè della determinazione di quale poligono sia il più vicino all'osservatore, viene risolto calcolando la profondità nella scena tridimensionale del poligono in corrispondenza del dato pixel. A questo scopo è sufficiente interpolare linearmente i valori di profondità a partire da quelli sui vertici del poligoni, i quali sono noti perché fanno parte dei dati di modellazione della scena.

Esattamente in questo (nella bidimensionalità) consiste la differenza fra il metodo di z-buffer ed il metodo di ray tracing (illustrata in dettaglio più oltre), il quale invece segue i raggi di proiezione nello spazio e ne calcola le intersezioni in tre dimensioni con gli oggetti della scena, determinandone in tal modo il più vicino. Per ogni pixel, l'algoritmo di z-buffer calcola l'informazione solamente per quegli oggetti la cui proiezione cade in quel pixel, sfruttando la coerenza. Il *ray tracing*, invece, interseca ogni raggio che parte dall'occhio con ogni oggetto che si trova sulla scena. Dunque lo *z-buffer* non calcola alcuna dispendiosa operazione di intersezione (come invece fa il *ray tracing*) poiché esso semplicemente opera un confronto fra le coordinate **z** ottenute nella trasformazione prospettica sul view plane: queste coordinate gli vengono passate direttamente dalla routine di trasformazione prospettica, ed esso si limita eventualmente ad interpolarne i valori lungo le scan line, sfruttando il calcolo incrementale.

### PSEUDOCODICE

```
void zBuffer (void)
{
        int x, y;
        for (y = 0; y < YMAX; y++)
                                                  // Cancella il frame buffer e lo z-buffer
                 for ( x = 0; x < XMAX; x++ )
        {
                         WritePixel (x, y, Colore_sfondo);
                 {
                         WriteZ (x, y, 0);
                 }
        }
        for ( ogni poligono )
                                                           // Disegna i poligoni
        {
                 for ( ogni pixel contenuto nella proiezione del poligono )
                         double pz = z del poligono alle coordinate (x, y);
                 {
                                  if (pz \ge ReadZ(x, y))
                                                                   // Il punto è finora il più
                                  vicino
                                                                   // (o vicino quanto il
                                                           precedente)
                                                                   // all'osservatore in quel
                                                  pixel
                         {
                                 WriteZ (x, y, pz);
                                 WritePixel (x, y, colore del poligono alle coord (x, y));
                         }
                }
        }
}
```

#### CALCOLO INCREMENTALE

Grazie alla coerenza di profondità (depth coherence), possiamo semplificare il calcolo della **z** per ogni punto di ogni scan line sfruttando il fatto che un poligono è piano. Normalmente, per calcolare la **z**, dovremmo risolvere per la variabile z l'equazione di un piano Ax+By+Cz+D=0

$$z = \frac{-D - Ax - By}{C}$$

Quindi, una volta calcolata la soluzione **z** come funzione di (x,y), il valore di **z** nel punto  $(x + \Delta x, y)$  è

$$z_1 - \frac{A}{C}(\Delta x)$$

Questo mostra che, a partire dal valore di **z** nel punto (x,y), per calcolare **z**(x+1,y) è necessaria soltanto una sottrazione, poiché il quoziente A/C è costante (lo si calcola una sola volta e poi lo si memorizza) e  $\Delta x = 1$ . Per determinare il primo valore di **z** sulla scan line successiva si effettua un analogo calcolo incrementale: si decrementa di B/C per ogni  $\Delta y$ .

In alternativa, se la superficie non è stata determinata o se il poligono non è piano, z(x,y) può essere calcolata interpolando le coordinate z dei vertici del poligono lungo coppie di spigoli, e dopo lungo ogni scan line. Anche in questo caso possono essere usati calcoli incrementali.



## TEMPO DI CALCOLO

L'algoritmo di *z–buffer* scandisce il piano di visuale ed i poligoni della scena proiettati su di esso con l'ordinamento naturale al crescere di *x* e *y* (*x* da sinistra a destra, *y* dall'alto in basso), senza che siano necessari confronti, e l'ordinamento secondo la *z* richiede soltanto un confronto per pixel per ogni poligono la cui proiezione contiene quel pixel. Il tempo impiegato dai calcoli per la determinazione delle superfici visibili tende ad essere indipendente dal numero di poligoni negli oggetti, poiché mediamente il numero di pixel coperti da ogni poligono diminuisce man mano che aumenta il numero di poligoni sulla scena: infatti, se sulla scena sono presenti tanti poligoni piccoli, ciascuno di essi coprirà meno pixel di quanti ne coprirebbe ciascun poligono più grande, il cui numero di altra parte sarebbe inferiore. Naturalmente, pero, la fase di modellazione subisce un aumento di tempo di calcolo dovuto al maggior numero di poligoni.

Si può notare che se la condizione di visibilità di un oggetto su un dato pixel non è soddisfatta, non occorre calcolare il colore dell'oggetto su quel pixel. Dunque, se i calcoli per lo shading richiedono molto tempo, si può ottenere maggiore efficienza ordinando gli oggetti dal dietro in avanti (*front-to-back*) in modo da mostrare prima gli oggetti più vicini.

L'algoritmo di *z–buffer* richiede molta memoria per il buffer, però è facile da implementare. Se si vuole risparmiare memoria, l'immagine può essere scansionata in strisce, il che richiede soltanto abbastanza memoria di *z–buffer* per la striscia che viene processata, ma comporta di dover effettuare vari e scansioni per ciascun oggetto (una per striscia).

A causa della sua semplicità e della mancanza di strutture dati addizionali, lo *z*–*buffer* viene spesso implementato in hardware e firmware al fine di diminuire l'uso della memoria RAM ed i tempi di calcolo del processore principale.

## IMPLEMENTAZIONE ED ALIASING

Lo *z-buffer* è spesso implementato a livello hardware con valori interi a 16 bit o 32 bit, mentre le implementazioni a livello software usano di solito valori in virgola mobile.

Anche se uno *z-buffer* a 16 bit offre un'adeguata gamma di valori per molte applicazioni CAD/CAM, 16 bit non danno abbastanza precisione per rappresentare ambienti in cui oggetti definiti al millimetro sono posti lontani chilometri, a causa deklla discretizzazione numerica della coordinata z che il computer introduce. A peggiorare le cose, se viene usata una proiezione prospettica, la compressione risultante dalla prospettiva dei valori z più distanti ha un serio effetto sull'ordinamento in profondità e nell'intersezione tra gli oggetti più lontani. Infatti, due punti vicino al piano di visuale che danno luogo a due differenti valori discretizzati di z, se spostati più lontano potrebbero dare luogo ad uno stesso valore discretizzato di z.

La precisione finita dello *z-buffer* è responsabile di un altro problema: l'aliasing. Di solito gli algoritmi di scan-conversion rendono due differenti gruppi di pixel quando disegnano la parte comune di due spigoli che appartengono alla stessa retta ma che hanno estremi diversi. Alcuni di quei pixel condivisi potrebbero vedersi assegnati valori di z leggermente diversi a causa della possibile inaccuratezza nell'effetturare l'interpolazione delle z. Questo effetto è molto visibile negli spigoli condivisi dalle facce di un poliedro. Alcuni dei pixel visibili lungo uno spigolo potrebbero essere colorati come se appartenessero ad un certo poligono, mentre i rimanenti potrebbero essere colorati col colore di un poligono vicino. Questo problema può essere risolto inserendo dei vertici aggiuntivi.

## VANTAGGI DELLO Z-BUFFER E SUE VARIANTI

L'algoritmo di *z-buffer* non richiede che gli oggetti siano necessariamente dei poligoni. Quindi, uno dei suoi punti di forza è che può essere usato per rendere ogni tipo di oggetto: si deve solo detrminare il valore **z** della profondità dell'oggetto e la sua tonalità per ogni punto nella sua proiezione sul piano di visuale.

Lo *z-buffer* può essere usato in maniera vantaggiosa anche dopo aver reso un'immagine. Dato che il buffer è l'unica struttura di dati usata dall'algoritmo per la determinazione delle superfici visibili, basta salvarlo con l'immagine e lo si potrà riutilizzare ed aggiornare in seguito se si vogliono aggiungere nuovi oggetti alla scena. L'algoritmo può essere inoltre codificato in modo da non modificare il contenuto dello *z-buffer* mentre si stanno rendendo degli oggetti selezionati. Se lo *z-buffer* viene così mascherato, un singolo oggetto (funzione a singolo valore in x e y) può essere disegnato su un set separato di piani sovrapposti con le superficie nascoste adeguatamente rimosse, e quindi può essere cancellato senza per questo toccare i contenuti dello *z-buffer*. Perciò, un semplice oggetto, come per esempio una freccia o un righello, può essere mosso lungo l'immagine nelle direzioni x, y e z, e servire da *cursore 3D* che oscura gli oggetti nell'ambiente o ne viene oscurato. Possono essere create delle sezioni di profondità evitando di scrivere sul *frame buffer* e sullo *z-buffer* quando il valore della **z** è dietro ad un dato piano di taglio. Inoltre, se gli oggetti rappresentati hanno un solo valore di **z** per ogni (x,y), allora i contenuti dello *z-buffer* possono essere usati anche per calcolare l'area ed il volume.

In [*J.R. Rossignac, A.A.G. Requicha, Depth-Buffering Display Techniques for Constructive Solid Geometry, Comp. Graphics and Appl. 6(9), 1986, 29-39*] viene adattato l'algoritmo di *z-buffer* per trattare oggetti definiti nella Constructive Solid Geometry (un metodo per descrivere la geometria di una scena complessa applicando un set di operazioni a forme primitive, abbreviato in CGS). Ogni pixel che si trova sulla proiezione di una superficie viene disegnato solo se il punto corrispondente della scena più vicino in profondità (valore **z**) appartiene ad un oggetto CSG costruito a partire dalla superficie.

Invece di memorizzare solo il punto con la z più vicina ad ogni pixel, in [*P.R. Atherton, A Method of Interactive Visualization of CAD Surface Models on a Color Video Display, SIGGRAPH 81, in Computer Graphics 15(3) Agosto 1981, ACM SIGGRAPH, New York*] viene proposto di salvare una lista di tutti i punti, ordinati secondo la z e per ciascuno di essi anche un riferimento alla superficie a cui appartengono: in questo modo creiamo un *objectbuffer.* Questa struttura di dati permette di visualizzare la scena con vari effetti, come per esempio la trasparenza, il clipping e le operazioni booleane, senza bisogno di rieseguire la scansione gli oggetti. Il *ray tracing*, nella sua forma più semplice ed elementare, per ogni pixel traccia dei raggi che partono dal punto di vista dell'osservatore e li interseca con ogni oggetto presente sulla scena al fine di determinare l'oggetto più vicino e dunque le caratteristiche di quel pixel.

Mentre l'algoritmo di *z-buffer* approssima un oggetto tramite un set di valori **z** (indicanti la profondità) lungo i proiettori che intersecano quell'oggetto, il *ray tracing* approssima gli oggetti tramite un set di intersezioni lungo ogni proiettore che interseca la scena.

Per ogni pixel, l'algoritmo di *z-buffer* calcola l'informazione solamente per quegli oggetti la cui proiezione cade in quel pixel, sfruttando la coerenza. Il *ray tracing*, invece, interseca ogni raggio che parte dall'occhio con ogni oggetto che si trova sulla scena. Dunque lo *z-buffer* non calcola alcuna dispendiosa operazione di intersezione (come invece fa il *ray tracing*) poiché esso semplicemente opera un confronto fra le coordinate **z** ottenute nella trasformazione prospettica sul view plane: queste coordinate gli vengono passate direttamente dalla routine di trasformazione prospettica, ed esso si limita eventualmente ad interpolarne i valori lungo le scan line, sfruttando il calcolo incrementale.

## **ESERCIZIO SULLO Z-BUFFER**

*Esercizio.* Una scena consiste dei seguenti poligoni, visti frontalmente nel modo in cui si ricoprono. I due rettangoli orizzontali si indicano con 0, 1, dall'alto in basso, quelli verticali con 2, 3, 4 da sinistra a destra. Le profondità dei vertici sono le seguenti:

0: vertici di sinistra z= 1, vertici di destra z=0

1: vertici di sinistra z= 0, vertici di destra z=1

2: vertici alti z= 0, vertici bassi z=1

3: vertici alti z= 1, vertici bassi z=1

4: vertici alti z= 1, vertici bassi z=0



Applichiamo il metodo di z-buffer scandendo i poligoni nell'ordine ciclico a partire dal poligono il cui indice è 1. Tracciare i grafici dello z-buffer uno per volta, dopo la scansione di ciascun poligono, sui pixel dell'asse orizzontale tratteggiato.

#### Svolgimento.

La riga di scansione interseca i poligoni 0, 2, 3 e 4. Si tratta di tracciare i grafici della sezione su quella retta dello z-buffer (ossia della profondità), che cambiano via via che viene scandito un nuovo poligono: cambiano perché per ogni nuovo poligono inseriamo nello z-buffer, punto per punto della retta di scansione, il valore massimo fra la profondità del nuovo poligono e quella trovata in base ai poligoni precedenti. Il valore iniziale dello z-buffer è 0. Esaminiamo i valori di profondità di ciascuno dei poligoni sulla retta di scansione.

Profondità del poligono 0:



Profondità del poligono 2: la profondità varia da 0 (in alto) a 1, la retta di scansione si trova a circa 4/5 dell'altezza del poligono, quindi lì la profondità vale 0.2:



Profondità del poligono 3: si trova a profondità costante 1:



Profondità del poligono 4: varia da 1 a 0, e sulla retta di scansione, a circa 4/5 dell'altezza, vale 4/5=0.8:



Ora scandiamo i poligoni, nell'ordine 0,2,3,4.

Sezione dello z-buffer dopo la scansione del poligono 0: è uguale al grafico della profondità del poligono 0:



Dopo aver scandito anche il poligono 2, il grafico rimane lo stesso, perché la profondità del poligono 2 è dappertutto inferiore a quella del poligono 0.

Dopo aver scandito anche il poligono 3, il grafico cambia perché il poligono 3, nel segmento della retta di scansione in cui viene intersecato, ha profondità superiore a quella dello z-buffer fino a quel momento, e diventa il seguente:



La stessa cosa capita dopo aver scandito anche il poligono 4. Prendendo il massimo fra la profondità del poligono 4, pixel per pixel sulla retta di scansione, e lo z-buffer precedente, si ottiene:



Questo è il risultato finale. Si arriva allo stesso risultato finale se si effettua la scansione dei poligoni in qualsiasi altro ordine, ma i grafici alle varie fasi sono diversi.

ALGORITMO DI SCANSIONE DI RIGA

## ALGORITMO DI SCANSIONE DI RIGA

L'algoritmo di scansione di riga si propone di determinare la visibilità dei poligoni presenti nella scena proiettandoli sulla finestra di visuale (*viewport*) e poi processandoli riga per riga, su righe di scansione successive. L'algoritmo utilizza tre strutture dati per memorizzare le informazioni relative agli oggetti presenti nella scena e quelle concernenti il processo di scansione.

La prima struttura dati o più semplicemente tabella si chiama Tabella degli Spigoli (ET).

I suoi dati riguardano tutti gli spigoli non orizzontali di tutti i poligoni proiettati sul piano di visuale.

Gli spigoli sono ordinati in gruppi in base alla loro coordinata y minore (y<sub>min</sub>) sul piano di visuale; all'interno dei gruppi gli spigoli sono invece ripartiti in ordine crescente rispetto alla coordinata x dell'estremo più basso.

Ogni dato (*record*) della tabella contiene:

- la coordinata x<sub>min</sub> corrispondente alla coordinata y<sub>min</sub> più piccola dello spigolo (che quindi comincia al punto (x<sub>min</sub>, y<sub>min</sub>));
- la coordinata y del vertice opposto (y<sub>max</sub>);
- l'incremento ∆x nella variabile x che si ha quando si passa da una riga di scansione alla successiva;
- un numero ID di identificazione, cioè un puntatore ad un poligono, per identificare il poligono della modellazione della scena a cui lo spigolo considerato appartiene. Di solito uno spigolo appartiene a due poligoni proiettati, uno a sinistra ed uno a destra, ed in tal caso si potrebbero memorizzare nel record entrambi i puntatori, ma l'algoritmo, poiché scandisce le righe da sinistra a destra, ha solo bisogno dell'identificatore del poligono che sta alla destra dello spigolo). Quando a destra non c'è nessun poligono l'ID è quello dello sfondo, considerato come un poligono dietro a tutti gli altri.



La seconda tabella è la **Tabella dei Poligoni (PT)**. Per ogni poligono vengono registrate, oltre all'ID, le seguenti informazioni:

- i coefficienti A, B, C, D dell'equazione del piano su cui giace il poligono;
- le informazioni riguardanti l'ombreggiatura ed il colore;
- una variabile booleana in\_out, inizializzata a *false* e utilizzata durante il processo di scansione.



## La terza tabella è la Tabella degli Spigoli Attivi (AET).

Questa tabella memorizza, per valori crescenti dell'ascissa di scansione x, gli spigoli intersecati dalla riga di scansione corrente. Per sapere a priori quali saranno tali spigoli basta consultare la Tabella ET degli spigoli, dove tutti gli spigoli sono raggruppati per gruppi con la stessa ordinata minima y<sub>min</sub>, cioe' con il vertice inferiore alla stessa altezza, e selezionare quei gruppi di spigoli la cui ordinata minima è inferiore alla ordinata della riga di scansione; poi, in questi gruppi, si selezionano quegli spigoli la cui ordinata massima y<sub>max</sub> è superiore alla ordinata della riga di scansione; poi, and tabella AET gli spigoli attivi sono memorizzati in maniera ordinata, al crescere

dell'ascissa di intersezione. Ovviamente, per ricavare tale ascissa non occorre seguire pixel per pixel la retta di scansione: si ricava per interpolazione a partire dall'ordinata della retta di scansione e dalle coordinate dei punti estremi dello spigolo.

In Figura 3 illustriamo il contenuto della Tabella AET in corrispondenza alle quattro righe di scansione disegnate nella successiva Figura 4:

AET					
Scan Lines	DATI				
У1	AD	AB			
y2	AD	AB	EF	FG	
Уз	AD	EH	AB	FG	
У4	AD	EH	BC	FG	

## Fig. 3

Dopo la costruzione delle tre tabelle, l'algoritmo procede alla scansione dei poligoni della scena (proiettati sul viewport). Si comincia con il poligono che ha per lato il primo spigolo nel primo gruppo della tabella ET: cioè si prende come ordinata y di scansione il valore della coordinata y più piccola che si trova nella ET (quella corrispondente al nel primo gruppo).



Fig. 4

Esaminiamo separatamente ciascuna riga di scansione in Fig. 4.

*Per*  $y = y_1$ *:* 



Quando la riga di scansione interseca il lato AD, i dati delle tabelle vengono aggiornati:

nella AET viene inserito il lato AD;

nella PT si seleziona il record corrispondente al poligono di appartenenza, cioè al poligono associato al numero ID di identificazione dello spigolo AD (il poligono sulla destra di AD), ed il valore della variabile logica in\_out di tale poligono viene invertito da *false* a *true*.

Una volta arrivati con la scansione al pixel in cui la variabile diventa nuovamente *false*, cioè al punto di intersezione con il lato di uscita, viene determinato il colore di tutti i pixel del poligono (ABCD) sul segmento scandito, shading, in modo consecutivo, sfruttando il concetto di coerenza di tratto visibile oppure, se si sta usando un meccanismo di ombreggiatura interpolata (ad esempio i meccanismi di ombreggiatura interpolata di Gouraud o di Phong), tramite interpolazione fra i colori dei pixel di bordo (quello di ingresso e di quello di uscita) del segmento orizzontale che abbiamo scandito nel poligono ABCD.

*Per*  $y = y_2$  :



La scansione riguarda ora due poligoni, ma la riga di scansione è "in" in un poligono solo alla volta. Si procede a colorare i due poligoni nello stesso modo del caso precedente.



La riga di scansione y<sub>3</sub> illustra un caso critico.

Per  $y = y_3$ :

I poligoni presenti nella scena sono ora contemporaneamente attivi, nel senso che le rispettive variabili booleane hanno entrambe valore true.

Quale dei due sia quello visibile si decide calcolando i valori della z di ciascun poligono.

I valori vengono trovati risolvendo le equazioni del piano di entrambi i poligoni rispetto alla profondità z, dopo aver posto  $y = y_3$  (l'altezza della riga di scansione), e x = ( $y_3 \cap EH$ ), cioè l'ascissa dell' intersezione tra la riga di scansione ( $y=y_3$ ) e il lato che ha reso attivo il nuovo poligono (EH).

Nel nostro caso EFGH ha un valore z maggiore, e quindi assegna il colore al segmento che deve essere tracciato.

A priori, lo stesso calcolo andrebbe ripetuto per ogni punto successivo della riga di scansione, perché l'ordine di profondità dei due poligoni potrebbe cambiare. Ma, per coerenza di tratto, si deve solo ripetere il calcolo al punto dove la riga di scansione esce dal poligono di sinistra (oppure, in casi più complicati di questo esempio, dove ci sono più di due poligoni attivi, si deve ripetere il calcolo al punto in cui la riga di scansione incontra un poligono che sta più avanti, cioè che viene intersecato in un punto di profondità minore). In ogni caso non occorre calcolare le ascisse di questi punti di intersezione, esse sono già tutte disponibili nella Tabella AET, ed anzi, per maggiore comodità, in quella Tabella sono memorizzata per ascisse crescenti, quindi da sinistra a destra. Per ogni tale punto di intersezione la Tabella ET ci dice in quale poligono stiamo entrando, e quindi ci permette di determinare la profondità di tale poligono in quel punto e procedere al confronto questa nuova profondità e quella del poligono da cui la retta di scansione proviene (entrambe si calcolano risolvendo le equazioni dei rispettivi poligoni; la seconda si potrebbe anche calcolare in modo incrementale se avessimo calcolato, dall'equazione del primo poligono, l'incremento di profondità  $\Delta z$  che corrisponde ad uno spostamento di un pixel verso destra quando si scandisce il primo poligono, ma in questo caso il calcolo incrementale non produce guadagno nel tempo di esecuzione, e quindi non lo applichiamo).

Se al punto di uscita l'ordine delle profondità non è cambiato, in tutto questo tratto il colore resta invariato (quello del poligono di sinistra) oppure ombreggiato per interpolazione, se stiamo usando un meccanismo di ombreggiatura interpolata. Altrimenti i due poligoni ssi intersecano in qualche ascissa intermedia. Questa ascissa di intersezione si calcola per linearità. A sinistra di essa si usa il colore di uno dei due poligoni (nel nostro caso EFGH), a destra quello dell'altro.

Osserviamo la seguente figura:



Il poligono IJKLMN (che chiameremo azzurro) è dietro gli altri due.

Quando la riga di scansione interseca lo spigolo BC, essa ha come poligoni attivi (cioè in") il poligono EFGH ed ancora IJKLMN. Per determinare chi dei due sta davanti a quel punto della riga di scansione si dovrebbe procedere con il calcolo del valore della profondità dei rispettivi poligoni; questi calcoli possono però essere evitati se si suppone che il modellatore fornisca poligoni che non si intersecano, cioè che nessun poligono si compenetri con un altro. Il fatto che non ci sia intersezione implica che, quando la riga di scansione non è più attiva nel poligono ADBC, le relazioni di profondità tra il poligono EFGH e IJKLMN non possono cambiare. Quindi il poligono EFGH continua ad essere quello più vicino all'osservatore. L'ipotesi di non intersezione semplifica in maniera analoga l'esame della profondità di due poligoni le cui proiezioni si sovrappongono, analizzato nell'esempio precedente (caso  $y=y_3$ ).

Ovviamente i calcoli dei valori della profondità non sono necessari quando la riga di scansione non è più attiva in un poligono coperto da un altro, ma sono necessari quando la riga esce da un poligono che stava davanti agli altri. Una variante dell'algoritmo di scansione di riga utilizza la coerenza di profondità. Se per due linee di scansione consecutive gli spigoli risultano gli stessi, posti nel medesimo ordine nella AET, allora le relazioni di profondità tra i poligoni in esame, lungo la riga di scansione, non subiscono alcuna variazione.

L'algoritmo di scansione di riga può essere utilizzato non solo per poligoni o superficie piane, ma anche per superficie generiche.

In tal caso le tabelle ET e AET vengono sostituite rispettivamente dalla Tabella delle Superficie (ST) e dalla Tabella delle Superficie Attive (AST), ordinate in base alle bounding boxes delle superficie nelle coordinate x e y (o meglio, dele loro proiezioni sul viewport).

Un approccio alternativo e' quello di utilizzare il concetto di "coerenza di invisibilità" [G.A.Crocker, Invisibility Coherence for Faster Scan-Line Hidden Surface Algorithms, SIGGRAPH 84, in Computer Graphics 18(3), luglio 1984, pagg. 95-102], cioè la tendenza di una superficie a rimanere invisibile se lo era alla riga di scansione precedente. In tal caso i valori della profondità z in ciascuna riga di scansione si memorizzano in uno z-buffer (il buffer ha la occupa capienza di una sola riga, quindi non molta memoria). Contemporaneamente alla AST, viene costruita una Tabella delle Superficie Invisibili (IST), alla quale una superficie viene aggiunta se il suo massimo valore di z nella riga di scansione corrente è minore del valore della z dello z-buffer alla riga di scansione precedente, calcolato sui pixel corrispondenti alle ascisse minima e massima, x<sub>min</sub> e x<sub>max</sub>, della superficie sulla riga di scansione precedente. Naturalmente, alcune volte buna superficie viene messa nella Tabella IST in maniera errata. Per evitarlo, durante la scansione ad ogni pixel si calcola il valore della profondità della superficie visibile attiva e lo si confronta con il massimo valore di z della superficie presunta invisibile: se quest'ultimo è maggiore, allora la superficie "invisibile" viene tolta dalla tabella IST. Se si suppone che le superficie non si intersechino fra loro, lo scavalcamento fra due superficie di cui una alla riga precedente copriva l'altra è possibile solo nel caso che la superficie anteriore sia stata tagliata, dando luogo ad un buco.

Questo metodo usa uno z-buffer di una sola riga. Se volessimo usare uno zbuffer per tutto il viewport la rimozione delle superficie nascoste sarebbe automaticamente ottenuta dall'algoritmo di z-buffer, ma solo al prezzo di confrontare per ogni pixel le profondità di tutte le superficie che vi si proiettano (ciascuna viene confrontata con la massima profondità ottenuta fino a quel momento). Invece l'algoritmo di riga di scansione che abbiamo appena presentato confronta solo le superficie attive su una data riga, ed usa lo z-buffer per diminuire al massimo i confronti (che nel caso presente possono essere più laboriosi che non nel caso dei poligoni, perché ora per ottenere il valore di z è necessario risolvere l'equazione della superficie, che non è più una semplice equazione lineare).

Ci sono vari modi di trattare lo sfondo.

Il modo più semplice è quello di inizializzare il frame-buffer al colore di sfondo. Questo rende inutile processare le righe di scansione che non intersecano spigoli.

Una seconda possibilità è di immaginarsi che della scena faccia parte anche un poligono "sfondo" con le seguenti caratteristiche:

- bounding box proiettata maggiore rispetto a tutti gli altri poligoni;
- profondità minore degli altri poligoni (cioè più indietro);
- parallelo al piano di proiezione;
- del colore dello sfondo.

Un'ultima alternativa è quella di modificare l'algoritmo al fine di scrivere nel frame-buffer il colore di sfondo ad ogni pixel su cui la riga di scansione non appartiene a nessun poligono.

# METODI DI RIMOZIONE DELLE AREE NASCOSTE BASATI SU ORDINAMENTO PER PROFONDITA'

## PRIORITA' PER ORDINAMENTO DI PROFONDITA' (*LIST PRIORITY*)

L'idea base degli algoritmi a priorità per ordinamento di profondità è quella di determinare l'ordine di visibilità degli oggetti presenti in una scena inserendoli in una lista ordinata per profondità. Il fatto che l'ombreggiatura sia effettuata rispettando le priorità stabilite garantisce la correttezza dell'immagine risultante.

I problemi da risolvere per l'implementazione di questi algoritmi riguardano l'ordinamento in profondità degli oggetti e la risoluzione di eventuali casi critici.

Se gli oggetti non si sovrappongono in z (coordinata relativa alla profondità) è sufficiente colorarli seguendo un ordine di profondità crescente.

Nel caso di sovrapposizioni elementari è ancora possibile determinare un ordine di visibilità corretto, mentre sovrapposizioni cicliche e compenetrazioni richiedono la scomposizione (clipping) di uno o più oggetti.

Gli algoritmi a priorità per ordinamento combinano operazioni a precisione d'oggetto con operazioni a precisione d'immagine.

Le prime vengono utilizzate per confrontare le profondità e per la scomposizione degli oggetti.

Le seconde sono utilizzate esclusivamente per la scansione prospettica (<u>scan-conversion</u>) (questa fase è inevitabile in qualsiasi algoritmo di Computer Graphics!).

Fanno parte degli algoritmi a lista di priorità l'algoritmo di ordinamento in profondità (<u>depth</u> <u>sort</u>) e l'algoritmo di partizionamento spaziale con albero binario (<u>BSP tree</u>).

La differenza principale tra i due algoritmi è che il secondo è indipendente dal punto di vista (cioè dalla posizione dell'osservatore).

# ORDINAMENTO PER PROFONDITA' (DEPTH SORT)

L'algoritmo di ordinamento per profondità (*depth sort*), introdotto da Newell, Newell e Sancha [*M.E.Newell, R.G.Newell, T.R.Sancha, <u>A Solution to the Hidden Surface Problem</u>, in <i>Proceedings of the ACM National Conference 1972, 443-450*], è una variante del più semplice "algoritmo del pittore", così

chiamato poiché si ispira alle correzioni su un dipinto: il pittore, dopo aver dipinto una oggetto distante, talvolta corregge il quadro dipingendovi sopra un nuovo oggetto più vicino. In tal modo viene nascosta la parte del primo oggetto non visibile all'osservatore.

L'idea alla base dell'algoritmo *depth sort* è proprio quella di visualizzare gli elementi in ordine di profondità decrescente rispetto al punto di osservazione. In questo modo gli oggetti più lontani sono progressivamente coperti da quelli più vicini.

Il processo si sviluppa in tre passi:

- 1. Stilare una lista ordinando i poligoni a partire dal più distante dal punto di osservazione (cioè quello uno dei cui vertici ha il minimo valore di z).
- 2. Risolvere eventuali ambiguità nell'ordine di sovrapposizione di due o più poligoni componendoli in parti più piccole che evitino l'ambiguità ⇒ Aggiornare la lista.
- 3. Effettuare la scansione prospettica (<u>scan conversion</u>) dei poligoni rispettando l'ordine della lista.

Le seguente figura mostra alcuni esempi di configurazioni spaziali il cui ordine di profondità genera ambiguità al passo 2. La prima scena è vista da sopra, le altre due frontalmente.



Alcuni casi in cui le estensioni in z dei poligoni si sovrappongono.

Tali ambiguità devono essere risolte per ottenere un ordinamento di profondità corretto: a questo scopo, ove necessario, si ritagliano i poligoni opportunamente.

Per questo scopo si confrontano i poligoni coinvolti nella situazione di ambiguità. Chiamiamo P il primo tra essi che compare nella lista e Q i poligoni le cui estensioni z si sovrappongono a quelle di P.

Si prende in esame il primo poligono Q.

Per verificare che P non possa oscurare Q si effettuano cinque test di complessità crescente. Basta che una delle cinque domande seguenti abbia risposta affermativa per essere certi che P non può coprire Q.

- 1. Le estensioni nella variabile x dei poligoni sono disgiunte?
- 2. Le estensioni nella variabile y dei poligoni sono disgiunte?
- 3. P è interamente sul lato opposto al punto di osservazione rispetto al piano individuato da Q?
- 4. Q è interamente sullo stesso lato del punto di osservazione rispetto al piano individuato da P?
- 5. Le proiezioni dei poligoni sul piano (x, y) (cioè il piano di visuale, o un piano ad esso parallelo) non si sovrappongono?

(Lo si può verificare proiettando i vertici dei due poligoni, tracciando i segmenti che uniscono i vertici proiettati – che sono le proiezioni dei lati dei poligoni – e confrontando i segmenti ottenuti da un poligono con quelli dell'altro per vedere se si intersecano).

Se la risposta ad una delle domande è affermativa diciamo che il corrispondente test è vero, altrimenti falso. E' interessante illustrare con un disegno le configurazioni tipiche relative ai test 3 e 4: le due figure seguenti sono viste da sopra.



**Nota**: come si determina se un poligono e' tutto dallo stesso lato di un piano, o tutto dal lato opposto?

- Prima si calcola il versore normale n del piano. E' il versore i cui coseni direttori sono i coefficienti A, B, C nell'equazione del piano (che supporremo normalizzati: A<sup>2</sup>+B<sup>2</sup>+C<sup>2</sup>=1): Ax+by+Cz=D significa appunto dire che il prodotto scalare fra n = (A,B,C) ed il generico punto (x,y,z) è costante, e quindi il punto (x,y,z) varia perpendicolarmente a tale versore; il valore di D rappresenta la proiezione lungo il versore normale, ossia la distanza (con segno) del piano dall'origine. Se D è positivo il piano passa nel semispazio individuato dalla direzione del versore, altrimenti nel semispazio opposto.
- Poi si calcola il prodotto scalare fra n ed i vettori che rappresentano i vertici del poligono: se per tutti i vertici il risultato è maggiore di D allora vuol dire che essi sono oltre il piano, nel semispazio individuato dalla direzione di n che ha per bordo quel piano, se invece il risultato è negativo i vertici sono ma nel semispazio dall'altro lato, al di qua del piano; altrimenti, se per un vertice il risultato è positivo e per un altro negativo, il segmento che li congiunge (e quindi il poligono) interseca il piano.

Non appena uno dei cinque test dà esito positivo si ha la conferma che P non copre Q, e quindi si lascia invariato l'ordinamento relativo fra P e Q e si passa all'eventuale poligono Q successivo, iterando il test per gli altri poligoni Q le cui estensioni si sovrappongono a quelle di P.

Se per tutti i poligoni almeno uno dei test dà esito positivo allora l'ordine di P nella lista va bene come è, e si può effettuare la scansione prospettica (<u>scan conversion</u>) di P.

Invece, nel caso in cui tutti i test diano esito negativo, si assume, temporaneamente, che P copra Q; si invertono provvisoriamente le loro posizioni nella lista e si effettua un nuovo test per verificare se questa scelta è confermata. A questo scopo i test 1, 2 e 5 non devono essere ripetuti, perché sono simmetrici in P e Q, e quindi il risultato non cambia scambiando P e Q. Invece i test 3 e 4 vanno ripetuti scambiando P e Q, e quindi diventano:

3'. Q è interamente sul lato opposto al punto di osservazione rispetto al piano individuato da P?

4'. P è interamente sullo stesso lato del punto di osservazione rispetto al piano individuato da Q?

Se almeno uno dei due nuovi test dà esito positivo è dimostrato che P oscura Q e questo conferma che è giusto invertire le loro posizioni nella lista: Q è il nuovo P e dovrà essere confrontato con tutti i poligoni che gli si sovrappongono in z.

Possono presentarsi dei casi, come quello mostrato in figura 1(b), in cui non si giunge a una conclusione dopo aver effettuato i vari test poiché non esiste un ordine corretto col quale effettuare la scansione prospettica (*scan conversion*).

In questi casi è necessario scomporre uno dei due poligoni tagliandolo con il piano che contiene l'altro.

Il poligono iniziale, non scomposto, viene eliminato dalla lista, ed al suo posto vengono inseriti i due frammenti generati dal ritaglio, nell'ordine appropriato. A questo punto si esegue l'algoritmo di prima sulla nuova lista aggiornata.

La figura 1(c) mostra il caso *ciclico*, in cui i poligoni sono disposti in modo tale che, effettuando i test, ognuno di essi possa essere inserito nella lista nell'ordine corretto rispetto a uno degli altri poligoni, ma non ad entrambi.

In tale situazione l'algoritmo appena spiegato entra in un ciclo infinito.

Per evitare questa eventualità, quando i primi cinque test tra P e Q danno esito negativo, il poligono Q, che viene spostato al posto di P in cima alla lista, viene contrassegnato con un marcatore logico (*flag*). Se dopo lo spostamento di Q in cima alla lista (chiamiamolo quindi P) succede che tutti i test con un altro poligono diano esito negativo, in tal caso i test 3' e 4' non vanno eseguiti sul poligono P già marcato con la flag, ed invece si procede comunque alla scomposizione di uno dei due poligoni e all'inserimento delle sue parti nella lista.

## DECOMPOSIZIONE DELLO SPAZIO TRAMITE ALBERI BINARI (BINARY SPACE PARTITION TREE)

L'algoritmo dell'albero di partizione binaria dello spazio (BSP) è un algoritmo particolarmente efficiente per la determinazione delle relazioni di visibilità tra un gruppo (non in movimento) di poligoni (o più in generale oggetti) in una scena tridimensionale, osservati da un punto di vista arbitrario.

È adatto ad applicazioni in cui può cambiare il punto di osservazione ma non la posizione relativa degli oggetti: per questo motivo viene utilizzato per generare scene animate in videogiochi e per presentazioni di modelli architettonici.

L'algoritmo fu introdotto in [R. Schumaker, B. Brand, M. Gilliland, W. Sharp, *Study for Applying Computer-Generated Images to Visual Simulation*, Technical Report AFHRL-TR-69-14, NTIS AD700375, U.S.Air Force Human Resources Lab., Air Force Systems Command, Brooks Air Force Basis, Texas, settembre 1969]. Esso considera la scena come composta da collezioni di facce, dette clusters.

L'algoritmo comincia col cercare un piano che separi interamente un gruppo di clusters da un altro.

Tale piano individua due semispazi.

I clusters che si trovano nello stesso semispazio del punto di osservazione possono coprire i clusters nel semispazio opposto, ma non esserne coperti.

Lo spazio viene ulteriormente suddiviso finché non è più possibile individuare piani separatori tra i clusters.

Tale decomposizione dell'ambiente può essere rappresentata mediante un albero avente per radice il primo piano separatore scelto, per nodi interni i restanti piani di separazione, e per nodi terminali (*foglie*) le regioni connesse di spazio individuate dall'intersezione dei vari piani.

La regione in cui è posizionato il punto d'osservazione determina un ordine univoco di reciproca copertura dei clusters.



L'idea è di assegnare alle facce dei clusters livelli di priorità indipendenti dal punto di vista (alta priorità corrisponde a bassi numeri di livello).

Si rimuovono le facce posteriori rispetto al punto di osservazione, perché coperte da quelle anteriori (stiamo supponendo che la scena ambiente consista di facce di oggetti poliedrali solidi chiusi).

Tra le facce rimanenti, quelle con un livello di priorità minore coprono quelle con un numero di priorità maggiore nei pixel in cui le proiezioni delle facce si intersecano.

Per ogni pixel, la faccia da visualizzare è quella con maggiore priorità fra tutte quelle la cui proiezione copre il pixel.

L'algoritmo BSP è una generalizzazione di questa idea.

È caratterizzato da un primo passo (costruzione dell'albero) costoso in termini di memoria e tempo, che viene eseguito una sola volta per scene statiche, seguito da un secondo (visualizzazione dell'albero), più rapido, necessario per ogni variazione nella posizione del punto di vista.

Il procedimento si basa sull'osservazione che nella scansione prospettica (<u>scan</u> <u>conversion</u>) dei poligoni della scena la rimozione delle parti nascoste è corretta se si adotta il seguente ordine di profondità crescente:

- 1. poligoni sul lato opposto a quello dell'osservatore
- 2. poligono dato
- 3. poligoni sullo stesso lato dell'osservatore

Ci si deve assicurare che quest'ordine venga rispettato nel considerare ogni singolo poligono per la visualizzazione.

Per questo scopo si realizza un albero di poligoni.

Come radice dell'albero, e di ogni sotto-albero, può essere scelto arbitrariamente uno qualsiasi dei poligoni nella scena, ma tale scelta può influenzare la velocità di esecuzione dell'algoritmo.

Per questo motivo è preferibile scegliere il poligono che tagli il minimo numero di poligoni "figli", in modo da limitare la proliferazione di frammenti di poligoni e quindi la quantità complessiva di dati da trattare.

Il piano su cui giace il poligono radice suddivide l'ambiente in due semispazi che contengono i restanti poligoni: un semispazio frontale (nel verso della normale uscente) e uno posteriore (nel verso opposto).

Se un poligono è attraversato dal piano del poligono radice, viene scomposto, e ciascuna delle due parti viene assegnata all'appropriato semispazio.

Per il poligono radice si scelgono due figli: uno appartenente al semispazio frontale (*front child*) e uno a quello posteriore (*back child*).

Ogni figlio viene a sua volta utilizzato per separare, rispetto al piano su cui giace, i poligoni appartenenti al proprio semispazio.

Il primo passo termina quando ogni nodo contiene un singolo poligono.



(a) inizio costruzione albero, con poligono 3

per radice

- (b) costruzione sotto-albero di sinistra
- (c) albero completo
- (d) albero alternativo, con poligono 5 per radice

Il secondo passo dell'algoritmo consiste nello stabilire l'ordine di visualizzazione dei poligoni della scena a seconda di dove è ubicato il punto di vista.

Si noti che l'albero può essere percorso seguendo cammini differenti in base alla

posizione del punto d'osservazione, per ottenere la giusta lista di priorità dei poligoni. Ad esempio, se l'osservatore si trova nel semispazio frontale individuato dal poligono radice, l'algoritmo visualizza nell'ordine:

- 1. tutti i poligoni appartenenti al semispazio posteriore, che potrebbero essere coperti dalla radice
- 2. la radice
- 3. tutti i poligoni appartenenti al semispazio frontale, che potrebbero coprire la radice

Ogni figlio della radice, e il relativo sotto-albero, vengono ricorsivamente processati dall'algoritmo nello stesso modo.

Ovviamente, se l'osservatore si trova nel semispazio posteriore l'ordine di visualizzazione sarà invertito.

Le seguenti figure riassumono i vari passi dell'algoritmo.

• Figura 1: scena vista dall'alto con cinque poligoni (a, b, c, d, e).



• Figura 2: creazione dell'albero



• Figura 3: fissato un punto di osservazione, viene stabilito l'ordine di visualizzazione dei poligoni (a, f, b, e, d, c).





# METODI DI RIMOZIONE DELLE AREE NASCOSTE BASATI SULLA SUDDIVISIONE DI AREA

# ALGORITMI A SUDDIVIONE D' AREA

L'idea di base è di suddividere il piano di visuale in aree diverse e vedere quali poligoni, proiettati su questo piano, cadono dentro quali aree.

Se la disposizione spaziale dei poligoni nella scena consente di individuare chiaramente quali poligoni sono visibili dentro una data area del viewport, non viene richiesta nessuna operazione ausiliaria; in caso contrario l'area deve essere suddivisa in regioni più piccole.

Questo procedimento è applicato in maniera ricorsiva fino a quando diviene possibile determinare quale poligono sia visibile in un dato pixel, cioè il colore del pixel.

Gli algoritmi a suddivisione d'area si ispirano al concetto di <u>coerenza d'area</u>, poiché aree sufficientemente piccole di un'immagine di solito sono contenute al massimo in un singolo poligono visibile (rammentiamo che l'immagine consiste di una tabella finita di pixel, celle indivisibili di colore). Nei casi in cui questo non avviene, allora si suddividono le aree fino ad arrivare a singoli pixel, e si determina il colore calcolando la media pesata dei colori dei poligoni che intersecano quel pixel, con pesi dati dalla percentuale di area coperta.

## L'ALGORITMO DI WARNOCK

L'algoritmo a suddivisione d'area di Warnock [J. Warnock, <u>A Hidden-Surface Algorithm for</u> <u>Computer Generated Half-Tone Pictures</u>, Technical Report TR 4-15, NTIS AD-753 671, Computer Science Department, University of Utah, Salt Lake City, UT, June 1969] si basa sulla divisione di un'area in quattro quadranti inizialmente uguali. Ogni quadrante, a turno, è considerato l'area corrente.

Ad ogni fase del processo di suddivisione ricorsiva, è possibile stabilire il rapporto fra ciascun poligono della scena (o meglio, della sua proiezione sul viewport) e l'area corrente, mediante una delle quattro relazioni seguenti:

il poligono circonda l'area in esame (Fig 1a);

il poligono interseca l'area (Fig 1b);

il poligono è interno all'area (Fig 1c);

il poligono e l'area sono disgiunti (Fig 1d).





I poligoni disgiunti non hanno chiaramente nessuna influenza sul colore dell'area corrente. Se un poligono interseca l'area lo scindiamo nella parte interna ed in quella esterna all'area. La prima viene considerata come poligono contenuto, mentre la seconda come poligono disgiunto. Questa procedura si chiama "ritaglio" (*clipping*).
Di seguito sono riportate le situazioni in cui si può determinare facilmente il colore dell'area, senza bisogno di ricorrere a suddivisioni ulteriori:

- Se tutti i poligoni sono disgiunti dall'area, viene visualizzato in tutta l'area il colore dello sfondo.
- Nel caso in cui esista un unico poligono che interseca l'area o vi sia contenuto, l'area è riempita con il colore di sfondo e, successivamente il poligono, mappato sui pixel contenuti nell'area tramite scansione prospettica (<u>scan conversion</u>), dà il suo colore ai pixel che ricopre.
- Se esiste un unico poligono che circonda l'area, allora quest'ultima viene riempita con il colore del poligono.
- Se più di un poligono circonda, interseca od è contenuto nell'area, ma ne esiste uno più vicino al punto di osservazione di tutti gli altri, anch'esso che circonda l'area, l'area viene riempita con il colore di quest'ultimo poligono. Per determinare se un poligono è il più vicino calcoliamo le coordinate z, relative alla profondità, di tutti i piani dei poligoni (che circondano, intersecano o sono contenuti) per ognuno dei guattro angoli dell'area di interesse, e vediamo se sono tutte piu' avanti di quelle degli altri poligoni interessati. Se esiste un poligono che circonda l'area le cui coordinate della z sono maggiori, e quindi più vicine rispetto al punto di visuale, allora il colore di sfondo dell'intera area è quello del suddetto poligono (vedere Fig 2). Si osservi che puo' accadere che un poligono sia piu' avanti degli altri, ma le sue coordinate z di intersezione con i quattro vertici dell'area di interesse non siano piu' avanti, perche' i poligoni possono essere disposti su piani obliqui che si intersecano e si accavallano. Il metodo di determinazione del fatto che un poligono sia tutto dal lato opposto rispetto all'osservatore rispetto al piano di un altro che abbiamo presentato nell'algoritmo di ordinamento per profondita' è guindi piu' completo, ma richiede piu' tempo: qui invece procediamo non per determinare in maniera esauriente tutti i casi di ricoprimento, ma per relegare i casi non chiariti ad aree sempre piu' piccole, che al di sotto di una certa soglia verranno colorate per approssimazione.



Figura 2

Se, dopo aver effettuato i calcoli relativi alla profondità, non si giunge ad una situazione univoca, e quindi non si può ancora decidere il colore, l'algoritmo di Warnock continua a suddividere l'area corrente; solo i poligoni intersecanti o contenuti nella area devono essere riesaminati, in quanto i poligoni disgiunti o che circondano continuano ad essere tali anche dopo la suddivisione. Vediamo alcuni esempi.



La Fig (3b) mostra due poligoni, giallo e blu, che circondano la stessa area di interesse. La Fig (3a) mostra l'intersezione dei piani dei due poligoni con l'area nel piano di coordinate (z, y). In essa il punto di visuale è a destra (z grande).

In questo caso, poiché non è possibile stabilire quale dei due poligoni ha profondità minore, in quanto il piano del poligono blu è più vicino all'osservatore di quello giallo per quanto riguarda la parte superiore dell'area di interesse, mentre per la parte inferiore ad essere avanti è il piano del poligono giallo, si deve procedere alla suddivisione dell'area.

Prendiamo ora in considerazione l'area di intersezione tra i due poligoni mostrati in Fig (3b); a tale area d'interesse appartengono i pixel dei due poligoni differenti (Fig.4a).

L'area può essere ulteriormente suddivisa (Fig.4b) fino ad essere visualizzata come composizione di pixel.

L'area di ogni pixel può essere divisa in sottopixel affinché, mediante un algoritmo a precisione di immagine, si determini il colore di ciascun pixel come media pesata dei colori dei poligoni che lo intersecano, con pesi dati dalla percentuale di sottopixel coperti da tali poligoni (Fig 4c).



Figura 4

Tutti i pixel che sono completamente interni o esterni all'area di interesse assumono il colore del pixel corrispondente al poligono più vicino al punto di visuale (valore della coordinata z, calcolata sul centro del pixel, maggiore rispetto a gli altri poligoni).

La figura seguente mostra una semplice scena e le suddivisioni che sono state necessarie per visualizzarla.



Il numero in ogni singola area indica quello dei quattro casi che si applica.

### L'ALGORITMO DI WARNOCK A GRIGLIA VARIABILE

Un'alternativa per la suddivisione in aree uguali è quella di dividere l'area partendo dal vertice di un poligono che vi si proietta (Fig.6). Una scelta del genere consente di evitare suddivisioni inutili e quindi permette di individuare in meno passi le aree d'interesse.



Anche in questo caso le aree contrassegnate numericamente fanno riferimento ai quattro casi.

Le aree, invece, che non hanno alcuna numerazione rappresentano quelle aree in cui nessuno dei quattro casi può essere applicato; esse necessitano di ulteriori suddivisioni prima che si possa determinare il colore.

## **ALGORITMO DI WEILER E ATHERTON**

Questo algoritmo, introdotto in [K.Weiler, P.Atherton, Hidden Surface Removal using Polygon Area Sorting, SIGGRAPH 77, in Computer Graphics 11(2), 1977, 214-222], è un algoritmo di suddivisione di area che utilizza poligoni invece che rettangoli, e ne ritaglia la proiezione sul piano di visuale rispetto a quella di un poligono iniziale prefissato, scomponendo quindi ciascuno degli altri in frammenti visibili all'osservatore e frammenti coperti dal poligono iniziale. Questa procedura viene ripetuta ricorsivamente in caso di situazioni di ambiguità nel ricoprimento, come quelle già incontrate nel presentare l'algoritmo di depth sort. In questa sequela di ritagli possono presentarsi frammenti costituiti da poligoni non convessi, anche nel caso che il modellatore avesse inizialmente fornito una scena composta solo da poligoni convessi. Pertanto l'algoritmo di Atherton-Weiler richiede un potente algoritmo di ritaglio (*clipping*) che consenta di ritagliare un poligono anche non convesso rispetto ad un altro: di solito si usa l'algoritmo di Weiler, introdotto in [K.Weiler, Polygon Comparison using a Graph Representation, SIGGRAPH 80, in Computer Graphics 14(3), Juglio 1980, 10-18. Il primo passo (non obbligatorio ma utile) è di ordinare i poligoni in base ai valori di z, ad esempio per z decrescente (all'aumentare della profondità, cioè all'allontanarsi dal punto di osservazione). Naturalmente la nozione di poligono più lontano o più vicino all'osservatore è mal posta: ha senso per i singoli vertici, non per gli interi poligoni. Infatti due poligoni A e B potrebbero essere disposti in maniera tale che un vertice di A sia più vicino all'osservatore del corrispondente vertice di B, ma per un'altra coppia di vertici l'ordinamento si scambi. Converremo di ordinare i poligono in base alla profondità del loro vertice più lontano dal punto di osservazione, come abbiamo fatto nell'algoritmo di depth sort, ma guesta volta a cominciare dal poligono più vicino all'osservatore. In tal modo l'ordinamento di profondità diventa ben definito, ma potrebbe comunque non essere conforme al vero ordinamento di copertura tridimensionale (l'ordinamento giusto va dal poligono più vicino al punto di visuale a quello più lontano nello spazio tridimensionale). E' proprio per questo che occorre ritagliare ricorsivamente i poligoni che creano una anomalia. In realtà, l'algoritmo di Atherton e Weiler taglia tutti i poligoni, non solo quelli anomali. Se le proiezioni sul piano di visuale di due poligoni non si sovrappongono, allora il ritaglio è banale, ma se si sovrappongono il ritaglio è necessario: in questo caso, infatti, non possiamo determinare la copertura tramite ad argomenti di separazione spaziale come quelli dei test 3 e 4 dell'algoritmo di depth sort, perché questi sono argomenti geometrici nello spazio tridimensionale, mentre il metodo di Atherton e Weiler utilizza le proiezioni bidimensionali. Osserviamo un altro aspetto legato alla stessa situazione bidimensionale: questo metodo si propone di scindere ogni poligono in un pezzo che la cui proiezione sul piano di visuale sta dentro a qualla del poligono di ritaglio ed un altro che invece sta al di fuori e quindi non interferisce nella visualizzazione; il pezzo interno può stare, nello spazio tridimensionale, davanti o dietro al poligono di ritaglio, ma potrebbe anche satre in parte davanti ed in parte dietro, però solo se i due poligoni si intersecano nello spazio. In tal caso la parte interna di uno dei due deve essere ulteriormente ritagliata in una antistante ed uno posteriore. Però questo ulteriore ritaglio dipende dalla geometria tridimensionale e non da guella bidimensionale dopo la proiezione prospettica: quindi non può essere eseguito confrontando queste proiezioni piane, ed invece deve essere svolto in una fase di preprocessing nel modellatore, il quale deve spezzare un poligono in ogni coppia di poligoni che si intersecano onde eliminare le intersezioni tridimensionali.

Una volta per tutte, i poligoni vengono proiettati prospetticamente sul piano di visuale, ma tenendo in memoria il valore di profondità tridimensionale dei loro vertici, come si fa nell'algoritmo di z-buffer. Da questo momento in poi il procedimento diventa puramente bidimensionale: ormai i poligoni sono tutti nello stesso piano (il viewport), ma ricordano i valori della profondità tridimensionale dei loro vertici.

Il primo poligono della lista ordinata, quello più vicino al punto di osservazione, è utilizzato per tagliare tutti i poligoni, incluso sé stesso, inserendoli in due liste contenenti rispettivamente le parti interne (in\_list) ed esterne (out\_list) al poligono di clipping. Tutti i poligoni (o meglio, frammenti di poligoni) nella in\_list che nello spazio tridimensionali risultano essere dietro il poligono di clipping vengono scartati, dal momento che non sono visibili: questo richiede un confronto delle profondità per verificare che essi siano dietro il poligono di ritaglio. Per questo scopo basta confrontare la profondità dei vertici del poligono nella in-list con quella dei punti corrispondenti del poligono di clipping: queste ultime possono essere determinate per interpolazione a partire da quelle dei vertici del poligono di clipping, oppure determinandone l'equazione sempre a partire dagli stessi dati. Il primo metodo è più naturale perché è bidimensionale: si può realizzare in due modi. Il primo modo è tramite interpolazione floating point (occorre determinare come scrivere le coordinate di viewplane dei punti del poligono di clipping come combinazione convessa di quelle dei vertici: se il poligono non è un triangolo questo è disagevole; inoltre in tal caso la combinazione convessa non è unica (ma tutte le varie combinazioni giuste danno lo stesso valore per la profondità). Il secondo modo, facile e rapido, è il metodo di calcolo incrementale riga per riga delle profondità: se si usa questo metodo, il che è quel che si fa di solito, l'algoritmo di Atherton e Weiler diventa basato su calcoli per pixel, e quindi a precisione di immagine.

Ora, quando un poligono della *in\_list* è più vicino all'osservatore del poligono di clipping, l'ordinamento iniziale non corrisponde al vero ordine di priorità della copertura in tre dimensioni. <u>Abbiamo già visto</u> il seguente esempio tipico di situazione ciclica, dove non esiste nessun poligono di clipping che dà un ordinamento corretto.



Ogni poligono non ordinato correttamente viene processato in modo ricorsivo: lo si pone all'inizio di una nuova lista, che coincide con la *in\_list* precedente, e lo si usa per ritagliare i poligoni in tale lista (come ribadiremo fra poco, per effettuare i ritagli si usa il poligono originale, non il suo frammento che si trovava nella *in-list*. il poligono che si usa per effettuare il clipping si effettua è sempre non un frammento bensì il suo poligono progenitore che apparteneva alla lista originaleç a questo scopo per prima cosa, prima di cominciare a ritagliare, si deve fare una copia della lista dei poligoni originali). Osserviamo

che il fatto che il poligono di clipping ritagli anche sé stesso ci assicura che il poligono di ritaglio sia sempre incluso nella propria *in-list* : quindi, quando si passa ad una nuova fase di ricorrenza perché un nuovo poligono che si presumeva essere dietro quello di ritaglio in realtà gli sta davanti, questo nuovo poligono ritaglia la *in-list* residua del precedente *incluso il poligono precedente stesso*, e quindi, correttamente, ne rimuove dalla lista il frammento ritagliato che esso ricopre. Al termine di ogni fase di ricorrenza legato ad un dato poligono di clipping, vengono disegnati i poligoni residui nella sua *in\_list* (generalmente il poligono di clipping stesso, a meno che esso od una sua parte non sia stata rimossa durante una fase di sospensione dell'azione di ritaglio di quel poligono perché esso è risultato coperto da un altro che lo sostituisce temporaneamente come poligono di ritaglio ed in questa fase ne rimuove un frammento).

Al ritorno da tutte le varie fasi di ricorrenza l'algoritmo ha rimosso tutti i frammenti nascosti dietro il primo poligono di clipping, ed anche il frammento di tale poligono di clipping eventualmente coperto da qualche altro poligono: in tal modo, almeno un poligono è stato disegnato e non lo si deve piu considerare nelle fasi successive.

In seguito, l'algoritmo continua allo stesso modo di prima, ma partendo ora dai poligoni della out\_list (che è anch'essa una lista ordinata empiricamente per profondità) ed eseguendo il ritaglio a partire dal poligono presunto frontale in tale lista. La out\_list ha almeno un elemento in meno della lista di poligoni originale (perché almeno un poligono è stato disegnato e scartato), e quindi volta per volta rimangono meno poligono da trattare: quindi l'algoritmo prima o poi termina. Come abbiamo appena osservato, il ritaglio viene sempre eseguito a partire da una copia dei poligoni originari progenitori e non direttamente a partire loro frammenti: questo è più conveniente in termini di tempo, poiché evita la ripetizione ricorsiva a partire da un numero di frammenti che cresce via via ad ogni passo di ricorrenza, il che potrebbe dare luogo ad una crescita esponenziale del tempo di elaborazione, e soprattutto al rischio che l'algoritmo non termini, perché i poligono da usare per ritagliare aumenterebbero, e quindi aumeterebbe il numero delle fasi di ritaglio. Perciò ai frammenti ottenuti dal ritaglio viene associato un puntatore al poligono da cui hanno origine, per poter identificare ed usare quest'ultimo se lo si deve usare in seguito come poligono di clipping. Ogni poligono derivante dal poligono di clipping viene messo nella in\_list senza ulteriori verifiche, perchè per il modo in cui ottenuto esso giace nel poligono originale e quindi è automaticamente davanti ai poligoni che restano (che sono, appunto, la *in\_list* del poligono usato per il ritaglio a quel passo di ricorsione). L'algoritmo utilizza una pila (stack) per tenere conto dei casi di sovrapposizione in cui un poligono è sia davanti sia dietro ad un altro (perché in una parte del viewport il primo ha profondità minore ed in un'altra maggiore - il che può accadere con poligoni non convessi anche se non si intersecano nello spazio, come abbiamo supposto) oppure guando i poligoni di un dato insieme si coprono reciprocamente in ordine ciclico. La pila contiene una lista di poligoni correntemente in uso come poligoni di clipping: quello corrente e quelli momentaneamente sospesi per passare ad un altro poligono di ritaglio nelle varie fasi ricorsive. Se si scopre che un poligono della lista corrente ad una data fase di ricorrenza si trova davanti al poligono di clipping, allora per prima cosa lo si cerca nella pila: se è lì, allora non è necessario iterare la ricorsione per metterlo al primo posto di una nuova lista provvisoria, perché questa lista provvisoria è la sua *in\_list*, ma, dal momento che quel poligono è stato già usato come poligono di ritaglio, tutti i pezzi dentro (e quindi, nella situazione iniziale tridimensionale senza intersezioni, dietro) quel poligono sono già stati rimossi.

#### PSEUDOCODICE:

```
void WA_superficie_visibile
{
polygon list= copia di tutti i poligoni della scena;
ordinamento dei poligoni della scena;
clear stack:
Finché (polygon_list non è vuota) {
WA suddivisione( il primo elemento della polygon list polygon list);
}
}
void WA_suddivisione(poligono di clipping, polygon_list)
/*taglio dei poligoni
{
in list=NULL;
out_list=NULL;
Per ogni poligono nella polygon_list {
Taglia il poligono con il genitore del poligono di clipping e inserisci i
frammenti nella interni al poligono di clipping nella in_list e quelli esterni nella
  out list:
}
rimuovi i poligoni o le parti di poligono sotto il poligono di clipping dalla in list;
/*quando l'ordine dei poligoni risulta incorretto
Per ogni poligono nella in list che non si trova nella pila e non è parte del poligono
di clipping
{
metti il poligono di clipping nella pila;
WA suddivisione(poligono da trattare, in list);
estrai il poligono di clipping dalla pila;
}
/*visualizzazione
Per ogni poligon della in_list {
Mostra il poligono;
}
*polygon_list=out_list; /* questo equivale a scartare dalla polygon_list iniziale
i poligoni della in_list */
} /* fine di WA_subdivide */
```

*Esercizio:* Dato l'insieme di figure sottostante, seguire lo pseudocodice dell'algoritmo di Weiler-Atherton e verificare cosa accade loro (le profondità dei vertici sono indicate nella figura iniziale dello svolgimento).





I	
	e solo Q <i>in</i> T rimane nella in_list.
	Q <i>in</i> T vine visualizzato e successivamente verrà visualizzato anche ToutQ.
IV	
V V	Ora nella polygon_list vengono inseriti i poligoni dell' out_list del poligono di clipping T. Rout T è la parte di poligono più vicino all'osservatore quindi usiamo R, che è il suo genitore, come poligono di clipping. La nuova in_list sarà RoutT e ZoutTinR e la nouva out_list sarà : QoutT, ZoutToutR, CoutT. ZoutTinR viene cancellato dalla in_list perché coperto da R.





Riassumiamo:















 $\mathbf{V}$ 





VII

VIII





Х

T diventa il poligono di clipping e dentro la polygon\_list ritroviamo: T, R, Q, C, Z;

la in\_list si riempie con : CinT, RinT, QinT e T, ZinT;

la out\_list si riempie con: CoutT, RoutT, QoutT e ZoutT;

poiché C*in*T, R*in*T e Z*in*T sono sotto T vengono rimossi e poiché Q*in*T si trova davanti al poligono di clipping entra in gioco la suddivisione ricorsiva e Q*in*T taglia T riempiendo una nouva in\_list con Q*in*T e T*in*Q, e una nuova out\_list con T*out*Q. Siccome T*in*Q è sotto Q*in*T, T*in*Q viene rimosso e solo Q*in*T rimane nella in\_list e vine visualizzato. Successivamente verrà visualizzato anche *Tout*Q. Ora nella polygon\_list vengono inseriti i poligoni dell' out\_list del poligono di clipping T. Rout T è la parte di poligono più vicino all'osservatore quindi usiamo R, che il suo genitore, come poligono di clipping . La nuova in\_list sarà R*out*T e T*out*T*in*R e la nouva out\_list sarà : Q*out*T, Z*out*T*out*R, C*out*T. Z*out*T*in*R viene cancellato dalla in\_list perché coperto da R.

Si continua inserendo la out\_list del poligono di clipping R nella polygon\_list ossia i poligoni QoutT, ZoutToutR, CoutT. Il poligono più vicino sarebbe QoutT ma come è evidente dalla figura IX non interseca nessun altro poligono, quindi può essere mostrato. Si passa a ZoutToutR, il poligono di clipping diviene t e la in\_list è ZoutToutR e CoutTinZ mentre la out\_list è solo CoutToutZ. CoutTinT viene rimosso perché coperto da C e invece ZoutToutR viene mostrato. CoutToutZ diviene l'unico elemento dello della polygon\_list. A questo punto il poligono di clipping diventa C e CoutToutZ viene visualizzato in quanto è l'ultimo poligono della polygon\_list.

# **RAY TRACING**

# **RAY TRACING**

Per questo gruppo di lezioni, consigliamo anche di leggere l'eccellente breve presentazione di Mario Costa Sousa, <u>http://pages.cpsc.ucalgary.ca/~mario/courses/591-691/topics/part-III/1-ray-tracing/index1.htm</u>

Il Ray Tracing è un algoritmo a precisione d' immagine che, per ogni pixel del rettangolo di visuale (viewport), determina la superficie della scena visibile attraverso quel pixel tracciando un *raggio di proiezione* che dal punto di visuale (centro di proiezione) attraverso il centro di quel pixel fino a toccare l'oggetto della scena più vicino. Il colore del pixel è quello di questo primo oggetto intersecato.



Centro di proiezione

PSEUDOCODICE DEL RAY TRACING (NON RICORSIVO):

```
Seleziona il centro di proiezione e il piano della finestra;
Per ogni scan line {
    Per ogni pixels {
        Determina il raggio che va dall' osservatore al centro del pixel;
        Per ogni oggetto della scena {
            Se si ha l' intersezione e l' oggetto è il più vicino all' osservatore
            Memorizza l' intersezione e il nome dell' oggetto;
        }
        inizializza il colore del pixel con il colore dell' oggetto intersecato più vicino
        all' osservatore;
        }
}
```

II Ray Tracing fu sviluppato per la prima volta da Appel [A.Appel, <u>Some Techniques for</u> <u>Shading Machine Renderings of Solids</u>, SJCC 1968, 37-45] e da Goldstein e Nagel [Mathematical Application Group, Inc., 3-D Simulated Graphics Offered by Service Bureau, Datamotion, febbraio 1968, 69; R.A.Goldstein, R. Nagel, <u>3-D Visual Simulation</u>, Simulation 16(1), gennaio 1971, 25-31] per determinare le superficie visibili dal punto di osservazione ed il loro colore in base ad una equazione di illuminazione. In seguito fu esteso ad una versione ricorsiva in grado di calcolare effetti di luce riflessa e rifratta, ed effetti di ombra, da Whitted [T.Whitted, <u>An Improved Illumination Model for Shaded Display</u>, CACM 23(6), giugno 1980, 343-349] e Kay [D.S.Kay, <u>Transparency</u>, <u>Reflection and Ray Tracing for</u> <u>Computer Synthesized Images</u>, M.S. Thesis, Program of Computer Graphics, Cornell University, Ithaca, NY, gennaio 1979].

A titolo di esempio, illustriamo nelle prossime pagine il <u>calcolo della intersezione del raggio</u> <u>di proiezione con una sfera o con un poligono piano</u>, e presentiamo alcune <u>varianti per</u> <u>migliorare l'efficienza</u> del Ray Tracing ricorsivo.

### ESEMPIO DI CALCOLO DELL'INTERSEZIONE DEL RAGGIO PROIETTORE CON UNA SFERA E CON UN POLIGONO PIANO

Il Ray Tracing ha il compito di determinare l'intersezione di un raggio con un oggetto. Per far ciò, usiamo la rappresentazione parametrica di una retta. Ogni punto (x, y, z) lungo un raggio che va da ( $x_0$ ,  $y_0$ ,  $z_0$ ) a ( $x_1$ ,  $y_1$ ,  $z_1$ ) è del tipo

$$x = x_0 + t (x_1 - x_0)$$
  

$$y = y_0 + t (y_1 - y_0)$$
  

$$z = z_0 + t (z_1 - z_0)$$

Per convenienza poniamo

$$\Delta x = x_1 - x_0$$
  

$$\Delta y = y_1 - y_0$$
  

$$\Delta z = z_1 - z_0$$

(si osservi che il vettore ( $\Delta x$ ,  $\Delta y$ ,  $\Delta z$ ) è il vettore che, applicato al punto di osservazione, individua la direzione del centro del pixel). Con questa notazione, l'equazione parametrica del raggio proiettore che parte al punto di visuale ( $x_0$ ,  $y_0$ ,  $z_0$ ) e passa per il centro ( $x_1$ ,  $y_1$ ,  $z_1$ ) di un pixel sulla finestra di visualizzazione (*viewport*) diventa

$$x = x_0 + t \Delta x$$
  

$$y = y_0 + t \Delta y$$
  

$$z = z_0 + t \Delta z$$

I valori di *t* fra 0 e 1 corrispondono al segmento fra il centro di visuale ( $x_0$ ,  $y_0$ ,  $z_0$ ) ed il centro del pixel, ( $x_1$ ,  $y_1$ ,  $z_1$ ). Valori negativi di *t* rappresentano i punti dietro il centro di proiezione, mentre valori di *t* maggiori di 1 corrispondono a punti sull'altro lato del centro di proiezione rispetto al piano del viewport. Se gli oggetti della scena sono dati da primitive grafiche, cioè da forme geometriche standard, per determinare il parametro di intersezione *t* con il raggio proiettore abbiamo bisogno della posizione degli oggetti e dell'equazione che definisce ciascuna di queste forme geometriche. Spesso gli oggetti sono definiti dalla loro superficie, modellata a maglie poligonali (*wireframe*). In tal caso il problema consiste nel calcolare il punto di intersezione dei raggi proiettori con i poligoni: lo tratteremo fra poco. Prima consideriamo un esempio facile di intersezione con un oggetto non poligonale. In questo contesto, l'oggetto più semplice è la sfera. La sfera con centro *a*, *b*, *c* e raggio *r* è rappresentata dall'equazione

$$(x-a)^{2} + (y-b)^{2} + (z-c)^{2} = r^{2}$$

L'intersezione si trova sostituendo in questa equazione i valori di x,y,z dell'equazione parametrica del raggio proiettore:

$$[(\Delta x)^{2} + (\Delta y)^{2} + (\Delta z)^{2}]t^{2} + 2t[(x_{0} - a)\Delta x + (y_{0} - b)\Delta y + (z_{0} - c)\Delta z] + (x_{0} - a)^{2} + (y_{0} - b)^{2} + (z_{0} - c)^{2} - r^{2} = 0$$

Questa equazione è quadratica in *t*. Se essa non ammette radici reali allora il raggio e la sfera non s'incontrano; se una c'è una sola radice reale (con molteplicità 2) il raggio è tangente alla sfera. Altrimenti ci sono due radici reali, che corrispondono ai punti di intersezione con la sfera: la radice (positiva) più piccola corrisponde al punto di intersezione più vicino all'osservatore (e frontale rispetto ad esso: se la sfera circonda l'intera scena, incluso l'osservatore, allora l'altra radice è negativa e corrisponde ad un punto alle spalle dell'osservatore).

Per calcolare l'illuminazione della superficie nel punto di intersezione è necessario determinare anche la direzione normale alla superficie in quel punto. Ciò è particolarmente facile nel caso di una sfera, poiché la direzione normale coincide con il vettore raggio, dal centro al punto d'intersezione. La sfera con centro in (a,b,c) ha quindi una direzione normale ((x - a)/r, (y - b)/r, (z - c)/r) al punto d'intersezione (x, y, z).

Nel caso di un poligono, la direzione normale è ancora più facile da calcolare: se il poligono giace sul piano Ax+By+Cz+D=0, allora il versore normale al piano è proporzionale al vettore (*A*,*B*,*C*), quindi è dato da (*A*,*B*,*C*)///(*A*,*B*,*C*)//, cioè

$$(Ae_1 + Be_2 + Ce_3) / (|A|^2 + |B|^2 + |C|^2)^{1/2}$$
.

Invece, ciò che è più complicato è trovare il punto di intersezione di un raggio con un poligono. Questo perché per determinare dove un raggio incontra un poligono sono necessarie due fasi di calcolo: dobbiamo prima determinare se il raggio si interseca con il piano del poligono e poi se il punto d'intersezione si trova all'interno del poligono.

Partiamo di nuovo dall'equazione del piano, Ax+By+Cz+D=0. Sostituendovi le equazioni parametriche del raggio,  $x=x_0+t\Delta x$ ,  $y=y_0+t\Delta y$ ,  $z=x_0+t\Delta z$ , si ottiene

$$A(x_0+t\Delta x) + B(y_0+t\Delta y) + C(z_0+t\Delta z) + D = 0,$$

cioè

$$(A\Delta x + B\Delta y + C\Delta z) t + Ax_0 + By_0 + Cz_0 + D = 0,$$

da cui

$$t = -(Ax_0 + By_0 + Cz_0 + D) / (A \varDelta x + B \varDelta y + C \varDelta z).$$



Se  $A\Delta x + B\Delta y + C\Delta z = 0$ , allora il raggio e il piano sono paralleli e non s'intersecano. Un modo semplice per determinare se il punto d'intersezione si trova all'interno del poligono è proiettare ortograficamente sia il poligono sia il punto di intersezione su uno dei tre piani che definiscono il sistema di coordinate, come mostrato nella figura qui sopra. Si noti che un punto è interno ad un poligono se e solo se la stessa cosa succede dopo aver proiettato, tranne che nel caso in cui la proiezione del poligono degeneri in un segmento (quindi abbia area nulla). Quindi selezioniamo l'asse lungo cui proiettare in maniera che l'area proiettata sia la più grande: ovvero quello il cui coefficiente nell'equazione del poligono ha il valore assoluto maggiore. Infatti, la coordinata il cui coefficiente ha il valore assoluto maggiore è quella il cui asse ha la direzione più vicina alla perpendicolare del piano del poligono: rammentiamo che, se il poligono giace sul piano Ax + By + Cz = D, allora il versore normale al piano è un multiplo di (A, B, C)., quindi, se ad esempio A è molto più grande di B e C (in valore assoluto), il versore normale è vicino a (1,0,0), cioè al versore dell'asse x (il cui coefficiente è quello più grande di tutti). Osserviamo che questo significa che il piano è disposto in modo tale che piccole variazioni della variabile x producono grandi variazioni delle altre due variabili, cioè, appunto, che il piano ha pendenza minore rispetto alla variabile x che alle altre due. Ad esempio, se A=100, B=30 e C=1, l'asse di proiezione è l'asse x e quindi proiettiamo sul piano  $y_{z}$ : la proiezione sul piano  $y_{z}$  corrisponde a buttare via la variabile x, cioè a porre x=0. Il fatto di proiettare sul piano su cui la proiezione ha area maggiore serve ad escludere che, dopo la proiezione, l'area proiettata, ancorché non nulla, sia piccola, ad esempio dello stesso ordine di grandezza dell'errore di arrotondamento, il che renderebbe affetta da errore la determinazione del fatto che il punto p sia interno od esterno al poligono.

Il Ray Tracing determina il colore dei pixel calcolando l'intersezione del raggio proiettore che passa per il loro centro con tutti gli oggetti della scena. Se si memorizza, per ciascun pixel, il valore del parametro corrispondente all'intersezione, allora, se si cambia la scena aggiungendo un nuovo oggetto, non è necessario rieseguire tutto il calcolo, ma solo calcolare, per ciascun pixel, l'intersezione del relativo raggio con questo nuovo oggetto e poi confrontare i parametri (una caratteristica analoga a quella dell'algoritmo di <u>z-buffer</u>). Naturalmente, per entrambi gli algoritmi, la determinazione del colore richiede anche il calcolo del versore normale alla superficie intersecata.

## CONSIDERAZIONI SULL'EFFICIENZA RAY TRACING PER DETERMINARE LE SUPERFICI VISIBILI

Per ogni pixel, l'algoritmo di <u>z-buffer</u> calcola le informazioni solo per quegli oggetti la cui proiezione prospettica sul piano di visuale copre almeno in parte quel pixel. Invece, la versione elementare del Ray Tracing appena descritta traccia il raggio di proiezione che passa per il centro del pixel e lo interseca con ogni oggetto della scena. Non è sorprendente quindi l'evidenza documentata da Whitted, nel cui elaboratore l'algoritmo impiegava dal 75% al 95% per i calcoli delle intersezioni, in scene tipiche [*T.Whitted, An Improved Illumination model for Shaded Display, CACM, 23 giugno 1980, 343-349*]. Di conseguenza, gli approcci per migliorare l'efficienza di questo algoritmo cercano di accelerare i calcoli delle singole intersezione, o di evitarli completamente.

Ottimizzazione dei calcoli di intersezioni

Ray tracing ed operazioni booleane (CSG)

Antialiasing nel Ray Tracing

## **OTTIMIZZAZIONE DEI CALCOLI D'INTERSEZIONE**

Molti dei termini delle equazioni per le intersezioni fra raggi di proiezione ed oggetti contengono delle espressioni che non cambiano nel corso dell'esecuzione e possono essere calcolate in anticipo e memorizzate una volta per tutte, come per esempio la proiezione ortografica di un poligono su un piano. Grazie a questo si possono sviluppare dei metodi veloci per il calcolo dell'intersezione. Ad esempio si può utilizzare la trasformazione prospettica che sposta il punto di visuale all'infinito. In tal modo i raggi di proiezione diventano tutti paralleli all'asse z. Inoltre possiamo scegliere la trasformazione in maniera che la direzione del punto di visuale corrisponda a x=y=0: quando applichiamo la stessa trasformazione ad ogni oggetto della scena, ogni intersezione avviene a x=y=0. Ciò semplifica il calcolo delle intersezioni e consente di determinare l'oggetto più vicino mettendo in ordine crescente i valori di z assunti dagli oggetti trasformati quando x=y=0. Il punto d'intersezione viene poi ritrasformato attraverso la trasformazione inversa per potere eseguire il calcolo dell'illuminazione.

Un altro modo di diminuire i tempi dei calcoli d'intersezione consiste nel determinare e paragonare poliedri circoscritti (bounding volumes) agli oggetti della scena. Se un oggetto ha una forma complicata che richede troppo tempo per la determinazione di intersezioni, lo si può rinchiudere in un poliedro circoscritto di forma più semplice (quindi i calcoli d'intersezione saranno meno onerosi) come una sfera, un ellissoide o un parallelepipedo. Non è necessario sottoporre a verifica di intersezione quell'oggetto se il raggio di proiezione non interseca il suo poliedro delimitante. Kay e Kajiya [T.L.Kay, J.T.Kajiya, Ray Tracing Complex Scenes, SIGGRAPH 86, in Computer Graphics 20(4), agosto 1986, pagg. 269-2787 hanno suggerito di usare come poliedro delimitante un poliedro convesso formato dall'intersezione di fette (slab), ognuna definita da due piani paralleli che circondano l'oggetto. La figura sottostante mostra, visto dal di sopra, un oggetto circondato da quattro fette (definite, in questa proiezione piana, da coppie di linee parallele) e la loro intersezione. Così ogni fetta è rappresentata dall'equazione Ax+By+Cz+D=0, dove A, B e C sono costanti, e D varia fra due valori estremi D<sub>min</sub> e D<sub>max</sub>. Se lo stesso gruppo di fette parametrizzate è usato per circondare tutti gli oggetti, ogni volume delimitante viene determinato dai valori di D<sub>min</sub> e D<sub>max</sub> di ognuna delle sue fette. L'intersezione di un raggio con la superficie del volume delimitante si esegue considerando una fetta per volta. Consideriamo un raggio di proiezione  $\mathbf{r}(t) = \mathbf{p}_0 + t\mathbf{r} = (x_0 + t r_x, y_0 + t r_y, z_0 + t r_z)$ . Trovare l'intersezione di questo raggio con una fetta significa sostituire le coordinate di  $\mathbf{r}(t)$ nell'equazione Ax+By+Cz+D=0 per ciascuno dei due piani delle fetta, cioè ponendo al posto di D i valori D<sub>min</sub> e D<sub>max</sub>, rispettivamente: in tal modo si trovano i valori vicini e lontani di t. Quindi il valore cercato di t verifica  $Ax_0+By_0+Cz_0+D+t(Ar_x+Br_y+Cr_z)=0$ , e pertanto

 $t = -(Ax_0+By_0+Cz_0+D)/(Ar_x+Br_y+Cr_z).$ 

Nel delimitare diversi oggetti, utilizziamo un numero finito di fette. Raggruppiamole in modo da avere nello stesso gruppo fette parallele, al un insieme di fette parallele (cioè con gli stessi valori di A, B e C, ma valori diversi di D) per delimitare diversi oggetti, alla luce di quanto spiegato nel primo paragrafo di questa pagina si può ottimizzare il calcolo dei parametri di ingresso ed uscita da una fetta scrivendoli come t = (S + D)T, dove S=  $Ax_0 + Bx_0 + Cz_0$  e T=-1/(Ax + By + Cz). Sia S che T possono essere calcolate una sola volta per un dato raggio ed insieme di fette parallele. Poiché ogni volume delimitante è una intersezione di fette, l'intersezione del raggio con un volume delimitante è l'intersezione delle intersezioni del raggio con ciascuna fetta. Per ottenere il parametro giusto basta quindi calcolare il massimo dei valori di t corrispondenti alle intersezioni vicine con i piani delle varie fette ed il minimo dei valori di t corrispondenti alle intersezioni più lontane. Per ottimizzare il procedimento evitando di calcolare casi in cui non c'è intersezione, man mano che vengono considerate le fette vengono aggiornati i valori massimi delle

intersezioni vicine ed i minimi di quelle lontane per ogni coppia di piani delimitanti: fermiamo il procedimento quando i valori vicini smettono di esere inferiori a quelli lontani.



<u>Figura</u>: Poliedro formato da intersezioni di fette. (a) Oggetto limitato da un set fisso di fette parametrizzate. (b) Volume delimitato da due poliedri delimitanti.

**Evitare i calcoli d'intersezione.** Idealmente, per ogni raggio vorremmo verificare l'intersezione solo con oggetti con cui s'interseca veramente, ed inoltre solo con quello che dà luogo al punto di intersezione più vicino al punto di visuale (centro di proiezione). Illustriamo solo uno dei possibili approcci in tal senso: la <u>partizione spaziale</u>.

# SUDDIVISIONE DELLO SPAZIO

Questo procedimento suddivide lo spazio in maniera progressivamente più fine. Dapprima viene calcolato l'inviluppo poliedrico (*bounding box*) dell'intera scena. Esso viene poi suddiviso in una griglia regolare.



Ad ogni cella della partizione, il procedimento associa la lista degli oggetti che essa contiene interamente o in parte. Viceversa, ogni lista contiene puntatori grazie ai quali ogni oggetto in essa è associato a tutte le celle che lo contengono. Man mano che un raggio di proiezione viene tracciato dentro la scena, si deve calcolare l'intersezione del raggio solo con quegli oggetti che sono contenuti nella cella che esso sta attraversando.

Inoltre le celle vengono esaminate in base all'ordine di attraversamento, così quando di determina che in una cella c'è un'intersezione, non se ne deve verificare nessun'altra successiva. Però si calcolano le intersezioni anche con tutti gli oggetti restanti nella cella, perchè se il raggio ne interseca più di uno allora bisogna determinare quello che dà luogo al punto di intersezione più vicino all'osservatore.

Inoltre, se un raggio interseca un oggetto in una cella, si deve necessariamente controllare se il punto d'intersezione si trova nella stessa cella; è possibile infatti che l' intersezione ci sia, ma sia localizzata in una cella successiva, e che invece un altro oggetto nella cella corrente possa avere un punto di intersezione più vicino. Questo fenomeno è illustrato nella figura seguente :



L' oggetto b è intersecato nella cella 3 sebbene l'esistenza della intersezione sia stata determinata quando abbiamo considerato la cella 2.

Il raggio viene seguito mentre attraversa la griglia di celle finchè si determina un'intersezione con un oggetto contenuto in una di esse (ad esempio l'oggetto *a* nella cella 3). Per evitare di ricalcolare l'intersezione di un raggio con un oggetto che si estende a varie celle, il punto di intersezione ed il pixel che corrisponde al raggio vengono memorizzati insieme all'oggetto intersecato la prima volta che questo viene incontrato.

E' dovuto a Dippé e Swenser [M.Dippé, J.Swensen, <u>An Adaptive Subdivision</u> <u>Algorithm and Parallel Architecture for Realistic Image Synthesis</u>, SIGGRAPH 84, in Computer Graphics 18(3), luglio 1984, pagg. 149-158] un algoritmo a suddivisione variabile che produce partizioni diseguali ottenute suddividendo mediante un <u>octree</u> [A.S.Glassner, <u>Space Subdivion for Fast Ray Tracing</u>, IEEE Computer Graphics & Applications 4(10), ottobre 1984, 15-22]. In questo caso, si può usare l'algoritmo di <u>determinazione della cella vicina</u> per determinare la cella successiva che si trova lungo il raggio [H.Samet, <u>Implementing Ray Tracing with Octrees and Neighbor Finding</u>, Computer and Graphics 13(4), 1989, 445-460].

# **OPERAZIONI BOOLEANE CON RAY TRACING (CSG)**

In [R.A. Goldstein, R. Nagel, 3D visual Simulation, Simulation, gennaio 1971, 25-31] si utilizza il Ray Tracing per costruire oggetti mediante operazioni booleane a partire da un insieme di oggetti geometrici primitivi. Determinare l'unione, l'intersezione o la differenza di due solidi è complicato se lo si vuol fare tramite confronto diretto dei solidi. Ma grazie al Ray Racing i problemi tridimensionali vengono ridotti a calcoli in una dimensione (1D). L'intersezione fra un oggetto primitivo ed un raggio (parametrizzato da un parametro t) genera un insieme di valori di t ognuno dei quali specifica un punto nel quale il raggio entra o esce dall'oggetto.

Ogni tale valore di t definisce l'inizio di un tratto nel quale il raggio è dentro o fuori l'oggetto. Consideriamo un raggio alla volta e determiniamo l'unione, la differenza o l'intersezione (unidimensionali) dei due oggetti lungo quel raggio, esaminando i suoi tratti interni o esterni all'uno e all'altro.



In Figura sono evidenziati i tratti interni di un raggio che attraversa due oggetti e i tratti interni all'oggetto risultante dalle varie operazioni booleane.

Un oggetto si compone a partire da oggetti più semplici tramite operazioni booleane (unione, intersezione e differenza). In tal modo l'oggetto viene rappresentato da una struttura gerarchica ad albero: la gerarchia CSG (Costructive Solid Geometry), un albero nel quale i nodi sono gli oggetti. Gli oggetti composti sono i progenitori di quelli più semplici che li compongono. Consideriamo solo oggetti composti a partire da due oggetti/componenti più semplici, e così via iterativamente fino a scindere l'oggetto iniziale in un insieme di componenti primitive.

Per trovare l'intersezione di un raggio con l'oggetto rappresentato da un nodo si procede in maniera ricorsiva: si percorre l'albero dall'alto al basso e si associa a quel raggio e a quel nodo la lista delle intersezioni con i due oggetti che lo compongono corrispondenti ai nodi sinistro e destro del livello sottostante (cioè i componenti booleani dell'oggetto). Questo porta a considerare al più quattro valori di intersezione per il parametro t del raggio. A seconda dell'operazione booleana attuata a quel nodo, si desumono i due valori risultanti per il tratto interno grazie ad una procedura gerarchica di effettuazione della operazione booleana (*CGS\_combine*), che determina se i punti di intersezione con gli oggetti componenti appartengono all'interno dell'oggetto composto tramite quella operazione (questa procedura è presentata in fondo a questa pagina). Si procede così ricorsivamente per tutta la gerarchia. Prima presentiamo lo pseudocodice, poi la procedura gerarchica.

#### **PSEUDOCODICE:**

```
CSG (raggio, nodo) {
       Intersezione_sinistra, intersezione_destra;
       Se (il nodo è composito) {
              Intersezione sinistra=CSG(raggio, nodo -> Figlio sinistra);
              Se (intersezione_sinistra è uguale a NULL e nodo -> op è diverso da
              unione)
              {
                      restituisce NULL;
                      Altrimenti {
                             Intersezione destra=CSG (raggio, nodo -> Figlio destra);
                             restituisce CSG_combinato (nodo -> op, intersezione_sinistra,
                             intersezione_destra);
                      }
              }
              Altrimenti
              restituisce intersezioni tra oggetto e raggio;
       }
}
```

Il codice è stato ottimizzato per tenere conto del fatto che, se il raggio non interseca il componente corrispondente al nodo figlio al lato sinistro dell'albero, non c'è motivo di calcolare l'intersezione con il componente del nodo di destra qualora l' operazione booleana sia una differenza od un'intersezione [S. Roth, "Ray Casting for Modeling Solids", febbraio 1982, 109-144]. Solo se l'operazione è un'unione il risultato può essere non vuoto e bisogna effettuare la verifica.

Il sottoprogramma *CSG\_combina* presentato nello pseudocodice accetta come input le due liste che rappresentano i parametri dei punti di intersezione (cioè di ingresso e di uscita) del raggio dai due oggetti solidi che corrispondono ai nodi figli, e restituisce la lista combinata dei punti di ingresso e d'uscita dell'oggetto composito (cioè al nodo padre). Per far questo bisogna scartare alcuni punti di intersezione. Ad esempio, se l'oggetto composito si ottiene dall'operazione di unione a partire dai solidi componenti, si devono scartare tutti quei punti di intersezione di una delle due liste che sono punti interni dell'altro oggetto solido componente: infatti questi punti sono interni all'oggetto unione, e non punti

di frontiera. Quindi le liste comprendono una flag (cioè una variabile logica) che assume valori "in" o "out" a seconda che un punto di intersezione dia inizio a un tratto interno o esterno al solido. Per ciascuna operazione booleana, la flag nella lista composita si stabilisce a partire dalle flag delle liste componenti sulla base della seguente tabella:

А	В	Unione	Intersezione	Differenza
in	in	in	in	out
in	out	in	out	in
out	in	in	out	out
out	out	out	out	out

Ad esempio, nel caso dell'operazione unione, un punto di intersezione in una delle due liste componenti viene rimosso quando i flag prima e dopo di quel punto assumono il valore "in" in almeno una delle due liste (cioè il punto è interno al solido corrispondente) oppure se la prima lista in quel punto passa da "in" ad "out" e la seconda passa da "out" ad "in" o viceversa (cioè a quel punto si esce da uno dei due solidi componenti ma si entra nell'altro). Questo equivale a dire che il record è rimosso quando nella lista composita la flag vale "in" sia prima sia dopo quel punto. Naturalmente bisogna inizializzare appropriamente la flag se il raggio ha origine dentro uno dei due solidi.

# ANTIALIASING NEL RAY TRACING

Il Ray Tracing campiona i punti di una griglia producendo così immagini con aliasing. In [T. Whitted, "An Improved illumination Model for Shaded Display", CACM 23(6), giugno 1980, 343-349] viene sviluppato un metodo adattivo per emettere più raggi verso quelle parti dell' immagine che altrimenti produrrebbero l'aliasing. Questo sovracampionamento variabile serve ad ottenere una maggiore precisione per ogni pixel. Si associano i raggi ai vertici di ogni pixel piuttosto che al suo centro. Dopo che i raggi sono stati inviati attraverso i quattro angoli del pixel, si fa la media delle luminosità effettive corrispondenti. Se i quattro valori sono vicini illuminano il pixel con questo livello medio di illuminazione. Se no, il pixel viene ulteriormente suddiviso tramite i punti mediani dei suoi lati ed il suo centro, formando quattro sottopixels verso i quali si sparano nuovi raggi che attraversano i loro vertici.

La suddivisione procede in modo ricorsivo fino a che viene raggiunta la suddivisione più fitta prefissata o finchè le luminosità siano sufficientemente vicine. Quindi la luminosità del pixel è data dalla media, pesata in proporzione all'area, delle luminosità dei suoi sottopixels. Chiaramente, il sovracampionamento adattivo produce una approssimazione migliore del campionamento di area non pesato.



Si consideri, per esempio, la figura (a) che mostra i raggi "sparati" attraverso gli angoli di due pixels adiacenti, con una profondità di suddivisione massima di due.

Poiché non è necessaria una suddivisione ulteriore per il pixel determinato dai raggi A, B, C, D ed E (parte b) allora la luminositàsarà (A + B + C + D + E)/4.

Il pixel adiacente richiede un'ulteriore suddivisione, così vengono tracciati nuovi raggi G, H, I, J e K attraverso i vertici dei quattro sottopixel (c). In questo caso solo il sottopixel in basso a destra è diviso ancora tracciando i raggi L, M, N, O e P (d).. A questo punto si raggiunge il livello massimo predefinito di suddivisione. La luminosità del pixel quindi è:



Possono sorgere problemi di aliasing quando i raggi che attraversano il pixel mancano un oggetto piccolo. Ciò produce effetti evidenti se gli oggetti sono sistemati in disposizioni regolari, ad esempio a scacchiera, ed alcuni rimangono visibili, mentre altri scompaiono, oppure se in una serie di immagini consecutive di un oggetto in movimento l' oggetto appare alternativamente (a seconda che venga colpito o mancato dal raggio).

Whitted evita questi fenomeni circondando ogni oggetto con un inviluppo (bounding volume) sferico sufficientemente grande da essere intersecato da almeno un raggio proveniente dall'osservatore. Poichè i raggi si diramano dal centro di proiezione, la grandezza dell'inviluppo deve essere in funzione della distanza dall'occhio. Se un raggio interseca l'inviluppo ma non interseca l'oggetto, allora tutti i pixels che appartengono a quel raggio sono ulteriormente suddivisi finchè l'oggetto non viene intersecato.

# **SETTIMA PARTE**

# RENDERING

# L'EQUAZIONE DELL'ILLUMINAZIONE

# L'EQUAZIONE DELL'ILLUMINAZIONE

Ci proponiamo di stabilire, sulla base per ora di modelli empirici, l'intensità di luce nei punti visibili all'osservatore delle superficie che compongono la scena, sulla base del loro colore e del colore e della posizione delle sorgenti di luce e di quella dell'osservatore. Consideriamo varie componenti di illuminazione:

Luce ambientale Illuminazione diffusa Attenuazione con la distanza Illuminazione speculare Illuminazione da sorgenti non puntiformi

### LUCE AMBIENTALE

Questa componente consiste di una luce di fondo presente nella scena e che illumina ciascun oggetto in maniera non direzionale, in conseguenza alla diffusione della luce causata dalle riflessioni multiple fra gli oggetti della scena. In seguito studieremo una modellazione matematica delle gradazioni di luce di queste componenti diffuse (<u>radiosità</u>), ma per ora ci limitiamo a considerare un modello molto semplificato in cui ad ogni materiale è associata una costante di riflessione ambientale k<sub>a</sub> (nell'intervallo  $0 < k_a < 1$ ) che misura quale percentuale della illuminazione ambientale viene diffusa da quel materiale. Per maggiore precisione potremmo far dipendere k<sub>a</sub> anche dalla frequenza della luce incidente, ovvero dalla sua lunghezza d'onda  $\lambda$ , e quindi scrivere  $k_{a\lambda}$ , ma poiché in seguito studieremo <u>modelli fisici</u> precisi che chiariscono questa dipendenza per ora spesso ignoreremo la dipendenze della costante di riflessione ambientale dalla luce. Sia  $O_{d\lambda}$  la distribuzione spettrale del colore della superficie osservata (quando la si illumina con luce bianca), e  $I_{a\lambda}$  quella della luce ambientale. Allora la componente ambientale dell'illuminazione è

 $I_{\lambda} = I_{a\lambda} \; k_a \, O_{d\lambda}$ 

### ILLUMINAZIONE DIFFUSA: IL MODELLO DI LAMBERT

Questa componente modella la diffusione della luce incidente che proviene da una data direzione L (L è il versore della direzione dal punto osservato sulla superficie alla posizione della sorgente di luce). Assumiamo per semplicità che ci sia una sola sorgente (se ce ne sono di più si devono sommare i corrispondenti contributi), e che essa sia puntiforme ed isotropa (cioè che l'intensità emessa non dipenda dalla direzione di emissione). Vogliamo modellare la diffusione da materiali opachi, come il gesso o il polistirolo. Il modello empirico che presentiamo, che si chiama modello di riflessione di Lambert, assume che l'intensità diffusa da una piccola porzione piana di superficie non dipenda dalla posizione dell'osservatore (purché l'osservatore stia all'esterno dell'oggetto racchiuso dalla superficie, cioè a patto che l'angolo formato dalla direzione dell'osservatore e la normale uscente sia minore di 90°, altrimenti l'illuminazione di Lambert, ed anche tutte le altre forme di illuminazione, si pongono uguali a zero). Naturalmente, però, l'illuminazione deve dipendere dall'angolo  $\theta$  formato dalla posizione della sorgente e la normale N alla superficie, perché un fascio di luce incidente con sezione, diciamo, A illumina la piccola porzione piana di superficie su un'area tanto più grande quanto più radente è la sua direzione L di provenienza, e precisamente pari a A/cos $\theta$  (per il momento stiamo considerando fasci di raggi paralleli, come se la sorgente fosse all'infinito; in seguito tratteremo l'attenuazione dovuta alla distanza). Quindi, se l'intensità emessa dalla sorgente puntiforme al variare della lunghezza d'onda  $\lambda$  è  $I_{p\lambda}$ , la densità di energia luminosa sulla superficie si ottiene dal prodotto scalare  $I_{p\lambda} < L, N > = I_{p\lambda} \cos \theta$ , e l'equazione dell'illuminazione ambientale e diffusa diventa

$$I_{\lambda} = I_{a\lambda} k_a O_{d\lambda} + I_{p\lambda} k_d O_{d\lambda} < L, N >,$$

dove  $k_d$  è un opportuno coefficiente di diffusione, con valore nell'intervallo  $0 < k_d < 1$ , e che può dipendere dalla lunghezza d'onda.

Si noti che il coefficiente di diffusione dipende solo dal materiale, ma l'intensità di luce diffusa varia da punto a punto, perché variano il versore normale N e la direzione L dal punto osservato alla sorgente.
#### ATTENUAZIONE CON LA DISTANZA

L'esposizione precedente non tiene conto che i raggi di luce uscenti da una sorgente puntiforme a distanza d finita di distribuiscono in maniera isotropa, e quindi l'illuminazione si spande su una sfera di raggio d=d<sub>L</sub>. Di conseguenza l'illuminazione che emana da una sorgente puntiforme (non quella ambientale!) decresce proporzionalmente all'area della sfera, cioè proporzionalmente a d<sup>-2</sup>. Nel caso della luce del Sole, la cui grande distanza rende irrilevanti le variazioni di distanza da punto a punto della scena, l'illuminazione è di intensità costante, ed i raggi solari sono paralleli. Ma per le sorgenti a distanza d<sub>L</sub> finita dobbiamo moltiplicare il termine di Lambert per un fattore di attenuazione f<sub>att</sub>:

$$I_{\lambda} = I_{a\lambda} k_a O_{d\lambda} + f_{att} I_{p\lambda} k_d O_{d\lambda} < L, N >,$$

dove  $f_{att} = c/d_L^2$ . In realtà, però, questo termine crea contrasti troppo forti: ombre troppo scure e luci saturate. In parte questo è conseguenza del fatto che quasi nessuna sorgente di luce è puntiforme, e quelle non puntiformi causano, almeno a distanza comparabile con il loro diametro, un decadimento dell'illuminazione di ordine meno elevato di d<sup>-2</sup>. Perciò miglioriamo il modello scegliendo un fattore di attenuazione che sia combinazione lineare di termini di grado 2, 1 e 0 (costante):  $f_{att} = 1/(c_0+c_1d_L+c_2d_L^2)$ . Ma dobbiamo comunque imporre che questo coefficiente sia di attenuazione e non di aumento, cioè non superiore a 1: quindi poniamo

 $f_{att} = \min \{1/(c_0 + c_1 d_L + c_2 d_L^2), 1\}.$ 

Attenuazione atmosferica. Un altro effetto che può essere interessante generare è quello dell'attenuazione atmosferica, dovuta alla distanza non della luce ma dell'osservatore, per modellare fenomeni di foschia.

Questo si realizza attenuando l'illuminazione progressivamente con la profondità z. Vogliamo modificare l'illuminazione interpolandola fra i seguenti due valori: quello precedentemente calcolato,  $l_{\lambda}$ , ed un valore  $l_0$  di illuminazione minima dello sfondo che scegliamo e fissiamo. Per questo scopo si fissano un piano frontale  $z=z_1$  ed un piano di fondo  $z=z_0$ , ovviamente con  $z_0 < z_1$ , e si scelgono due livelli di attenuazione atmosferica  $s_b$  e  $s_f$ , il primo per il piano di fondo ed il secondo per il piano frontale, con  $s_b < s_f$ ; poi si interpola linearmente fra  $s_b$  e  $s_f$  per ottenere un valore  $s_z$  che cresce all'aumentare della profondità, cioè all'avvicinarsi all'osservatore (ricordiamo che l'asse z è orientato verso l'osservatore). Per valori di z inferiori alla profondità dello sfondo, cioè  $z < z_0$ , l'attenuazione rimane costantemente uguale a quella del piano frontale, cioè  $s_f$ . Il calcolo di  $s_z$  è immediato, grazie all'equazione della retta che passa per i punti ( $z_0$ ,  $s_b$ ) e ( $z_1$ ,  $s_f$ ): si ottiene

$$S_z = S_b + (Z-Z_0)(S_f - S_b)/(Z_1-Z_0).$$

Una volta calcolato il fattore di attenuazione  $s_z$ , l'interpolazione attenuata si ottiene dall'interpolazione convessa fra l'illuminazione non attenuata e quella del piano di fondo:

$$I'_{\lambda} = S_z I_{\lambda} + (1 - S_z) I_0.$$

#### RIFLESSIONE SPECULARE: IL MODELLO DI ILLUMINAZIONE DI PHONG

La riflessione speculare è la riflessione della luce su una superficie che avviene nella direzione speculare (rispetto alla normale esterna alla superficie) alla direzione di incidenza. Per materiali non perfettamente speculari, la luce si riflette non solo nella direzione speculare a quella di incidenza, ma anche in direzioni ad essa vicine, generando un alone (*highlight*). Per alcune superficie, come quelle lucidate a cera o la plastica (nella quale i pigmenti di colore sono immersi all'interno di un materiale trasparente) questa luce che si riflette alla superficie ma non all'interno ha il colore della luce incidente. Per altri materiali, come i metalli, le leggi dell'ottica mostrano che il materiale può cambiare il colore della luce riflessa: vedremo in seguito un <u>modello di riflessione basato sulle leggi fisiche</u>.

La direzione della luce riflessa. Per modellare la riflessione speculare in maniera euristica (cioè non basata su modelli fisici, per il momento), anzitutto calcoliamo la direzione della luce riflessa. La proiezione del versore L sul versore normale N vale  $\cos\theta$  N, dove  $\theta$  è l'angolo di incidenza. Perciò il vettore S da L al piede della proiezione vale S =  $\cos\theta$  N – L. Quindi il versore della direzione riflessa, R, è L+2S =  $2\cos\theta$  N – L.



D'altra parte, poiché L e N sono versori, cioè vettori di lunghezza 1, il coseno dell'angolo  $\theta$  che essi formano è il loro prodotto scalare:

$$\cos\theta = \langle L, N \rangle$$

Quindi si ottiene:

$$R = 2 < N, L > N - L$$
 (1)

Ora consideriamo l'angolo  $\alpha$  formato dal versore riflesso e dalla direzione V dell'osservatore.

Questo angolo misura la deviazione angolare dalla direzione di massima intensità di riflessione.



Analogamente a prima, si ha

$$\cos \alpha = \langle R, V \rangle$$

e quindi

$$\cos\alpha = \langle \mathsf{R}, \mathsf{V} \rangle = 2 \langle \mathsf{N}, \mathsf{L} \rangle \langle \mathsf{N}, \mathsf{V} \rangle - \langle \mathsf{L}, \mathsf{V} \rangle$$
(2)

*II modello di illuminazione riflessa di Phong.* Questo modello, introdotto in *[Phong Bui-Tuong, <u>Illumination for Computer Generated Pictures</u>, <i>Commun. ACM, 18* (6), 1975, pp. 311-317], descrive in maniera euristica la riflessione da materiali non perfettamente speculari, per i quali quindi si hanno componenti di luce riflessa non solo nella direzione speculare ma anche in direzioni vicine, con una deviazione  $\alpha$  non nulla, le quali naturalmente diventano progressivamente più deboli. Il modello di Phong utilizza un decadimento rispetto ad  $\alpha$  proporzionale a  $\cos^{n}\alpha$ : poiché  $\cos 0 = 1 \text{ ma } 0 < \cos\alpha < 1 \text{ per } 0 < |\alpha| < \pi/2$ , tutte le curve  $\cos^{n}\alpha$  hanno valore 1 per  $\alpha=0$ , ma tendono a zero quando n tende a infinito per  $\alpha$  diverso da 0. Quindi, per n = 1, il decadimento è blando, e gli aloni sono alquanto diffusi, ma per grandi valori di n il decadimento è rapido e l'alone è concentrato: scompare al superamento di una soglia piccola della deviazione  $\alpha$  dalla direzione di massima riflessione speculare. Il valore di n da scegliere dipende solo dalla natura del materiale, e più precisamente dalla sua specularità.

La frazione di energia luminosa riflessa può dipendere non solo da  $\alpha$  ma anche dall'angolo di incidenza  $\theta$ , quindi dovremmo introdurre un fattore di attenuazione della riflessione speculare k( $\theta$ ) con 0<k( $\theta$ )<1: ma per semplicità assumiamo che il fattore k di attenuazione speculare sia indipendente da  $\theta$ , e lo denotiamo con k<sub>s</sub>. Ovviamente, come sempre, se  $|\theta| > \pi/2$  l'osservatore sta dietro la superficie e il valore dell'illuminazione è posto uguale a zero. A questo proposito, osserviamo che il modello di Phong si trova nell'imbarazzante necessità di decidere cosa fare quando  $|\alpha| > \pi/2$ , caso nel quale cos  $\alpha = < R$ , V> <0. In questo caso l'equazione

dell'illuminazione di Phong non può valere perché termini negativi per l'illuminazione non hanno senso fisico. Il modo più indolore di eliminare il problema è di porre il termine di Phong uguale a zero se <R, V> <0, cioè se  $|\alpha|>\pi/2$ . Si noti che questa modellazione non è fisicamente corretta, perché può succedere che l'angolo  $\alpha$  fra la direzione di visuale e quella del raggio riflesso sia maggiore di  $\pi/2$  (si veda la figura seguente), ma in tal caso l'osservatore vede comunque illuminazione non nulla. Per risolvere questo aspetto in maniera fisicamente corretta bisogna utilizzare modelli più complessi, non euristici ma fisici, come faremo in seguito.



Ora l'equazione dell'illuminazione diventa, grazie a (2):

$$I_{\lambda} = I_{a\lambda} k_a O_{d\lambda} + f_{att} I_{p\lambda} [k_d O_{d\lambda} < L, N > + k_s < R, V >^n]$$
(3)

Si noti che la componente di illuminazione riflessa non ha un colore proprio, bensì quello della luce incidente. Questo infatti è il colore riflesso da uno specchio. Però certi metalli colorano la luce che essi riflettono, diversamente a seconda dell'angolo di incidenza (si veda la trattazione dei modelli fisici di illuminazione nel seguito). Invece, per quanto osservato prima vall'inizio di questa sezione, questa equazione di illuminazione si presta a modellare la riflessione da superficie come la plastica o le superficie smaltate con cera, nelle quali la riflessione avviene alla superficie lucida e trasparente prima che i raggi di luce colpiscano le particelle di pigmento all'interno. Se per correggere questo effetto di resa tipo plastica vogliamo introdurre un *colore speculare*, possiamo modificare (3) inserendo un fattore  $O_{s\lambda}$  di distribuzione spettrale del colore riflesso:

$$I_{\lambda} = I_{a\lambda} k_a O_{d\lambda} + f_{att} I_{p\lambda} [k_d O_{d\lambda} < L, N > + k_s O_{s\lambda} < R, V >^n]$$
(3 bis)

In realtà, però, la resa del colore della riflessione ottenuta in questo modo non è molto accurata, perché è difficile trovare le distribuzioni spettrali adeguate: per renderla verosimile è ancora una volta indispensabile ricorrere a modelli fisici per la riflessione speculare e diffusa.

#### SORGENTI DI LUCI NON PUNTIFORMI: RIFLETTORI DI WARN

Come già accennato, normalmente le sorgenti di luce non sono puntiformi, neppure in via di approssimazione. Nelle riprese fotografice e cinematografiche si usano riflettori, o banchi di luce, in cui la distribuzione angolare dell'emissione di luce non è isotropa. Un modo di estendere il <u>modello di illuminazione di Phong</u> per includere questi riflettori è stato introdotto in *[David R. Warn, Lighting Controls for Synthetic Images, SIGGRAPH 83, in Computer Graphics (1983), pagg. 13-21]*. Al posto della sorgente puntiforme in direzione L si immagina di disporre un riflettore, modellato come una superficie piana illuminata da una nuova sorgente puntiforme disposta frontalmente ad esso, in direzione L'; indichiamo con  $I_{L'\lambda}$  la distribuzione spettrale della luce emessa da questo illuminatore.



Questa nuova sorgente non contribuisce all'illuminazione della scena altro che per il fatto di illuminare il riflettore, il quale a sua volta ne riflette la luce con distribuzione angolare in accordo al modello di illuminazione di Phong: ad un angolo  $\gamma$  di deviazione rispetto alla normale L' del riflettore la luce emessa è  $I_{L'\lambda}$  cos<sup>n</sup>  $\gamma$ . D'altra parte, è immediato dalla figura che

$$\cos\gamma = \langle -L', L \rangle$$

e quindi la luce emanata al riflettore di Warn è determinata dalla seguente equazione dell'illuminazione di Warn:

$$I_{L'\lambda} < -L', L>^n$$
 .

Si possono considerare due varianti del riflettore di Warn che modellano sorgenti di luce di uso comune in fotografia:

- il primo consiste dell'illuminatore con alette di focalizzazione (*flaps*). Le alette oscurano lateralmente il fascio di luce incanalandolo su una banda. Per modellare questo illuminatore basta porre uguale a zero l'illuminazione al di fuori di una banda prestabilita. Ad esempio, per incanalare la luce in una banda trasversale all'asse x, diciamo da x. a x<sub>+</sub>, basta lasciare l'illuminazione come deriva dall'equazione di Warn nei punti (x,y,z) per i quali x<sub>-</sub> < x < x<sub>+</sub>, e porla uguale a zero per gli altri punti dello spazio (quelli esterni a questa banda);
- il secondo consiste nel modellare una luce spot in una direzione prefissata. A questo scopo basta annullare la luce al di fuori di un cono centrato in questa direzione, cioè per i punti dello spazio che eccedono una determinata deviazione rispetto all'asse determinato da questa direzione.

# MODELLI FISICI DI ILLUMINAZIONE

#### 1 Microsfaccettature di una superficie

Il modello di Torrance–Sparrow, introdotto in [K.E. Torrance, E.M. Sparrow, R.C. Birkebak, Polarization, Directional Distribution and Off-Specular Peak Phenomena in Light Reflected from Roughened Surfaces, J.Opt.Soc.Am. 56 (1966), 916–925] and [K.E. Torrance, E.M. Sparrow, Theory for Off-Specular Reflection from Roughened Surfaces, J.Opt.Soc.Am. 57 (1967), 1105–1114], è un modello di superficie riflettenti basato su leggi fisiche. Blinn lo applicò alla Computer Graphics, paragonandolo al modello di Phong, in [J.F. Blinn, Models of Light Reflection for Computer Synthesized Pictures, SIGGRAPH 77, in Computer Graphics 11 (1977), ACM, New York, 192–198]. In seguito, Cook e Torrance ([R. Cook, K.E. Torrance, A Reflectance Model for Computer Graphics, ACM Trans. on Graphics 1 (1982), 7–24) utilizzarono questo modello per ottenere la composizione spettrale della luce riflessa. Nel modello di Torrance–Sparrow la superficie è immaginata come un insieme di microscopiche sfaccettature che costituiscono riflettori perfettamente speculari. La distribuzione, la geometria di tali microsfaccettature e la direzione della luce (immaginata come emessa da una sorgente infinitamente distante, cosicché tutti i raggi si possono approssimare paralleli) determinano l'intensità e la direzione della riflessione speculare come una funzione dell'intensità I della sorgente luminosa puntiforme e del versore normale  $\overrightarrow{N}$ , del versore dal punto illuminato verso la posizione della sorgente di luce  $\overrightarrow{L}$  e di quello verso la posizione dell'osservatore  $\vec{V}$ . Verifiche sperimentali hanno mostrato un'ottima corrispondenza con tale modello.

Il modello fa uso di una adeguata espressione per la *riflettività bidirezionale* speculare, ossia per l'intensità di luce che proviene da una direzione assegnata e viene riflessa specularmente in un'altra direzione assegnata. Rinviamo la definizione formale della riflettività bidirezionale al seguito, nello studio della Radiosità ed ancor meglio della Illuminazione Globale.

Abbiamo già incontrato un modello di illuminazione euristico e non fisico, il modello di illuminazione di Phong, che assume che questo coefficiente di riflettività bidirezionale sia proporzionale ad una opprtuna potenza del coseno fra la direzione di osservazione e la direzione speculare alla luce incidente. Questo termine deve essere moltiplicato per un coefficiente di riflettività che misura il grado di specularità della superficie riflettente, ovvero quanta percentuale della luce incidente viene riflessa: tale coefficiente dovrebbe dipendere dall'angolo di incidenza, ma spesso si assume costante indipendentemente dall'angolo (quindi una proprietà solo del materiale), per semplicità ed ancor più perché, in un modello euristico non basato su leggi fisiche, non si è in grado di ipotizzare una adeguata dipendenza dall'angolo. Invece il modello di Torrance–Sparrow, che è basato su ipotesi fisiche, assume la seguente componente speculare della riflettività bidirezionale legata alla statistica della distribuzione angolare delle microsfaccettature:

$$\rho_s = \frac{F_\lambda}{\pi} \frac{DG}{(\overrightarrow{N} \cdot \overrightarrow{V})(\overrightarrow{N} \cdot \overrightarrow{L})}$$
(1.1)

Qui  $\overrightarrow{N}$  è il versore normale alla superficie al punto di riflessione,  $\overrightarrow{V}$  è il versore che individua la direzione dell'osservatore, ed i numeri  $F_{\lambda}$ , D e G, che calcoliamo nelle prossime Sezioni, rappresentano rispettivamente il termine di Fresnel delle leggi della riflessione dell'ottica fisica (Sezione 4), il coefficiente di attenuazione dovuto alla distribuzione angolare delle microsfaccettature (Sezione 2), ed il fattore di attenuazione geometrica che misura la probabilità che la luce inviata verso una microsfaccettatura ad un'altra sia oscurata da un'altra che si interpone (Sezione 3). Il fattore  $\pi$  al denominatore è introdotto al fine di tenere in considerazione l'aumento dell'area, rispetto ad una superficie piana ideale, dovuto alla ruvidità. Infatti su una superficie perfettamente piana e speculare un fascio cilindrico di raggi paralleli di luce incidente da una data direzione fissa copre un'area circolare, ossia un disco; invece in presenza di microsfaccettature perfettamente speculari ma orientate diversamente fra loro con un gran numero di possibili angoli lo stesso fascio copre un'area maggiore, più vicina a quella dell'emisfero sotteso dal disco, perché le microsfaccettature tendono a disporsi in maniera non parallela ma distribuita al variare dell'angolo. Infine, il fattore  $\overrightarrow{N} \cdot \overrightarrow{V}$  al denominatore serve ad aumentare l'area della superfice illuminata pewr tener conto della sua proiezione nella direzione ell'osservatore , ed analogamente  $\overrightarrow{N} \cdot \overrightarrow{L}$  è il fattore di proiezione nella direzione della luce: questi fattori misurano la attenuazione della riflessione bidirezionale dovuta al fatto che la luce incidente copre un'area maggiore della sezione del fascio (quindi dà luogo ad una densità di illuminazione minore), e l'osservatore vede una sezione di area minore dell'area illuminazta, a causa dell'inclinazione rispetto alla normale sia dell'una sia dell'altro.

## 2 Funzione di distribuzione delle microsfaccettature

Poiché le microsfaccettature sono considerate perfettamente speculari, il modello tiene in considerazione solo quelle le cui normali giacciono lungo la direzione bisettrice fra le direzioni di incidenza e di uscita (ossia di visualizzazione): si tratta della direzione del vettore  $\vec{H} = \frac{1}{2}(\vec{L} + \vec{V})$  (che, una volta normalizzato, si chiama lo  $halfway \ vector$ ). Solo una frazione D del totale delle microsfaccettatura ha tale orientamento. Torrance e Sparrow ipotizzano una funzione di distribuzione gaussiana. Altri lavori usano distribuzioni diverse: ad esempio Cook e Torrance usano una distribuzione più adeguata, la distribuzione di Beckmann, che per le superfici ruvide dà come risultato

$$D = \frac{1}{4m^2 \cos^4 \beta} e^{-[(\tan \beta)m]^2}$$
(2.1)

in cui  $\beta$  è l'angolo tra  $\overrightarrow{N}$  ed  $\overrightarrow{H}$ , m è lo scarto quadratica medio della pendenza delle microsfaccettature (ossia  $m^2$  è la varianza della distribuzione delle inclinazioni). Se m è piccolo allora l'inclinazione delle microsfaccettature varia poco rispetto alla normale della superficie e dunque la riflessione è molto a fuoco sulla direzione speculare (questo è il caso, per esempio, per uno specchio piano). Invece, per m grande l'inclinazione è elevata e la superficie ruvida diffonde in maniera uniforme la luce riflessa su tutto l'emisfero frontale (questo è il caso dei riflettori di Lambert, per esempio il gesso).



Figura 1. Diagramma polare della riflessione a seconda della funzione di distribuzione delle microsfaccettature

Per tenere conto di superficie con varie scale di ruvidità, si usa una combinazione convessa delle funzioni di distribuzione,

$$D = \sum_{j=1}^{n} w_j D(m_j)$$
 (2.2)

dove la somma dei pesi  $w_j$  è pari a 1.

### 3 Il fattore di attenuazione geometrica

Il modello tiene in considerazione il fatto che, rispetto alla sorgente o ad un'altra microsfaccettatura, ci siano microsfaccettature che ne coprono altre, occludendo la luce ad esse inviata dalla prima, o dalla sorgente. Questo fatto dà luogo ad un fattore di attenuazione geometrica G. Torrance e Sparrow, e Blinn, considerano tre differenti situazioni nel calcolo di G:



Figura 2. Paragone fra la distribuzione angolare della riflessione della luce incidente a  $70^0$  per i modelli di Phong e di Torrance–Sparrow

- (a) La luce incidente sulla microsfaccettatura è totalmente riflessa e non la colpisce più la stessa superficie.
- (b) La microsfaccettatura è totalmente esposta alla luce incidente ma quella riflessa viene parzialmente o totalmente intercettata da altre microsfaccettature (questa luce intercettata ed a sua volta riflessa contribuisce alla riflessione diffusa).
- (c) La microsfaccettatura è parzialmente schermata dalla luce incidente

Il fattore geometrico di attenuazione varia da 0 (occlusione totale) ad 1 (nessuna occlusione).

Nel primo caso, nel quale tutta la luce incidente viene riflessa, poniamo il fattore di attenuazione geometrica G uguale ad 1. In entrambi gli altri casi la quantità di luce intercettata da altre sfaccettature è pari a M/A, dove A è l'area totale della microsfaccettatura e M è l'area la cui luce riflessa è bloccata. Quindi  $G_b = G_c = 1 - M/A$ .

La proporzione di luce riflessa nel secondo caso è stata calcolata nel succitato articolo di Blinn ed il risultato è:

$$G_b = \frac{2(\overrightarrow{N} \cdot \overrightarrow{H})(\overrightarrow{N} \cdot \overrightarrow{V})}{\overrightarrow{V} \cdot \overrightarrow{H}}$$
(3.1)

Qui  $\overrightarrow{L}$  è il versore che individua la direzione della sorgente ed  $\overrightarrow{H} = 1/2(\overrightarrow{L} + \overrightarrow{V})$  è il versore normale della microsfaccettatura (prima non l'avevamo normalizzato perché facevamo solo riferimento alla sua direzione). La proporzione di luce riflessa  $G_c$  nel terzo caso si ottiene in modo analogo: basta osservare che in effetti la situazione è identica alla precedente tranne per il fatto che ora la luce bloccata non è quella uscente nella direzione  $\vec{V}$ dell'osservatore, bensì quella incidente, ovvero proveniente dalla direzione di  $\vec{L}$ . Quindi basta rimpiazzare al numeratore il fattore di attenuazione dato dalla proiezione nella direzione dell'osservatore,  $\vec{N} \cdot \vec{V}$ , con quello nella direzione della sorgente di luce,  $\vec{N} \cdot \vec{L}$ . Ovvero:

$$G_c = \frac{2(\overrightarrow{N} \cdot \overrightarrow{H})(\overrightarrow{N} \cdot \overrightarrow{L})}{\overrightarrow{V} \cdot \overrightarrow{H}}$$
(3.2)

Anche al denominatore sembrerebbe necessario sostituire  $\overrightarrow{V}$  con  $\overrightarrow{L}$ , ma è pleonastico perché, per definizione di *halfway vector*, si ha  $\overrightarrow{V} \cdot \overrightarrow{H} = \overrightarrow{L} \cdot \overrightarrow{H}$ . Come valore di G si prende infine il valore minimo tra i tre valori così calcolati.

### 4 Le equazioni di Fresnel per la riflessione della luce

Ora introduciamo nel modello di illuminazione di Torrance–Sparrow la legge fisica che regola la distribuzione in frequenza della luce riflessa: la legge di Fresnel.

L'equazione di Fresnel per la luce non polarizzata determina la frazione di luce riflessa da una superficie dielettrica (ossia non conduttrice):

$$F_{\lambda\theta_i} = \frac{\tan^2(\theta_i - \theta_t)}{\tan^2(\theta_i + \theta_t)} = \frac{1}{2} \frac{\sin^2(\theta_i - \theta_t)}{\sin^2(\theta_i + \theta_t)} \left( 1 + \frac{\cos^2(\theta_i - \theta_t)}{\cos^2(\theta_i + \theta_t)} \right) .$$
(4.1)

Qui  $\theta_i$  l'angolo fra  $\overrightarrow{L}$  e  $\overrightarrow{H}$ , e  $\theta_t$  l'angolo di rifrazione; dalla legge di Snell per la rifrazione sappiamo che sin $\theta_i = \eta_{i\lambda}/\eta_{t\lambda}$ , dove  $\eta_{i\lambda}$  e  $\eta_{t\lambda}$  sono gli indici di rifrazione dei due materiali, superficie riflettente ed ambiente circostante. L'equazione di Fresnel può anche essere espressa come

$$F_{\lambda} = \frac{1}{2} \frac{(g-c)^2}{(g+c)^2} \left( 1 + \frac{[c(g+c)-1]^2}{[c(g+c)+1]^2} \right)$$
(4.2)

dove  $c = \cos \theta_i$ ,  $g^2 = \eta_{\lambda}^2 + c^2 - 1$  e  $\eta_{\lambda} = \eta_{t\lambda}/\eta_{i\lambda}$ .

Invece in un materiale conduttore, che attenua la luce, perché l'equazione (4.2) continui a valere è necessario sostituire l'indice di rifrazione del materiale con un nuovo indice di rifrazione a valori complessi  $\check{\eta}_{\lambda}$ , definito così:

$$\breve{\eta}_{\lambda} = \eta_{\lambda} - ik_{\lambda} \tag{4.3}$$

(qui  $k_{\lambda}$  misura la frazione di luce assorbita per unità di lunghezza nell'attraversare il materiale rifrangente).

La parte immaginaria  $k_{\lambda}$  è il coefficiente di attenuazione della luce nel materiale: esso quindi misura la percentuale di attenuazione dell'intensità della luce a frequenza  $\lambda$  per unità di distanza atrraversata. Per semplificare i calcoli di riflessione e rifrazione, nei materiali conduttori  $k_{\lambda}$  si assume sia zero. Questo elimina la componente immaginaria, e lascia un singolo valore  $\eta_{\lambda}$  reale.

Per apprezzare la differenza del rendering fra i modelli di illuminazione di Torrance–Sparrow e di Phong è opportuno esaminare il caso di luce radente: in questo caso, in (4.2) abbiamo  $\theta_i \approx 0$ , quindi  $c \approx 0$ , e quindi  $F_{\lambda} = 1$  per tutte le frequenze  $\lambda$ . Uno sviluppo di Taylor rispetto a c mostra che la distribuzione dell'illuminazione riflessa ad angoli diversi da quello esattamente speculare decade molto più rapidamente con la deviazione laterale rispetto a tale direzione di quanto non avvenga se si usa il modello di Phong. Infatti, nel modello di Phong dobbiamo calcolare l'angolo fra il versore riflesso  $\vec{R}$  e il versore del punto di osservazione  $\vec{V}$ . Ma se  $\theta_i = \arccos(\vec{L} \cdot \vec{H}) \approx \frac{\pi}{2}$ , allora  $\vec{L}$  è quasi perpendicolare all'*halfway vector*  $\vec{H}$ , e quindi  $\vec{L}$  e  $\vec{V}$  formano un angolo di circa  $\pi$  (poichè  $0 = \vec{H} \cdot \vec{L} = 1/2(\vec{L} \cdot \vec{L} + \vec{V} \cdot \vec{L}) = 1/2(1 + \vec{V} \cdot \vec{L})$ ) da cui  $\vec{V} \cdot \vec{L} = -1$ ). In tal caso  $\vec{V}$  e  $\vec{R}$  sono quasi allineati, quindi il coseno  $\vec{V} \cdot \vec{R}$  vale circa  $\cos(\theta_i - \frac{\pi}{2}) \approx 1$ . Allora lo sviluppo di Taylor rispetto a  $c = \cos \theta_i$  del termine di Phong  $(\vec{V} \cdot \vec{R})^n$  è dell'ordine di

$$\cos^n\left(\theta_i - \frac{\pi}{2}\right) = \sin^n \theta_i^n \approx \left(1 - \frac{(\theta_i - \pi/2)^2}{2}\right) \approx 1 - \frac{n(\theta_i - \pi/2)^2}{2}, \quad (4.4)$$

che vale pur sempre 1 se  $\theta_i = \frac{\pi}{2}$ , ma decade quadraticamente al deviare da questo valore di  $\theta_i$  (e quindi con lo stesso ordine ma comunque più velocemente del coseno, perché il primo termine che abbiamo trascurato nello sviluppo di Taylor in (4.4) ha segno positivo). Questo più elevato decadimento laterale è ben visibile nelle seguenti immagini tratte dal succitato articolo di Blinn, per angoli di incidenza (e quindi di riflessione) di 30<sup>o</sup> (Figura 3) e di 70<sup>o</sup> (Figura 4). In esse l'illuminazione è data da un fascio di raggi paralleli e si assume che il colore della luce speculare dipenda solo da quello della luce incidente (quindi si trascura una delle conseguenze del termine di Fresnel). Quelle che appaiono come superficie solide descrivono in realtà i diagrammi polari tridimensionali dell'intensità di luce riflessa alle varie direzioni (superficie più elongate rappresentano quindi distribuzioni angolari più concentrate). L'effetto che si ottiene è una maggiore concentrazione della luce riflessa quando la luce incidente è radente, come mostra la Figura 5.



Figura 3. Paragone fra la distribuzione angolare della riflessione della luce incidente a $30^0~{\rm per}$ i modelli di Phong e di Torrance–Sparrow



Figura 4. Paragone fra la distribuzione angolare della riflessione della luce incidente a $70^0~{\rm per}$ i modelli di Phong e di Torrance–Sparrow



Figura 5. Paragone fra l'illuminazione di una sfera metallica per i modelli di Phong e di Torrance–Sparrow al variare della direzione della luce incidente

## 5 La variazione del colore nella riflessione speculare

Abbiamo accennato che nell'articolo di Blinn si fa l'ipotesi che la lunghezza d'onda (ovvero il colore) della luce riflessa sia identica a quella della luce incidente. Ma l'equazione di Fresnel prevede che la luce riflessa abbia distribuzione spettrale con una variazione continua della lunghezza d'onda. Pertanto un modello fisico più preciso deve tener conto di tale distribuzione, che dipende dai tre fattori seguenti (si riveda l'uguaglianza (4.1):

- la natura dei materiali (tramite i loro indici di rifrazione complessi);
- la lunghezza d'onda della luce incidente;
- l'angolo d'incidenza.

In funzione di tali parametri, si ha uno shift in frequenza nella luce riflessa rispetto a quella incidente: ovvero, la la luce riflessa viene *colorata* dal materiale su cui si riflette. Questa conseguenza della espressione (4.1) è illustrata nella Figura ??, che mostra l'andamento del termine di Fresnel in funzione della lunghezza d'onda e dell'angolo di incidenza per la riflessione su rame: il termine di Fresnel vale 1 indipendentemente da  $\lambda$  per angolo di incidenza zero (luce ed osservatore disposti in modo radente, l'illuminazione è del colore della sorgente), ma ne dipende per piccoli angoli di incidenza (in tal caso l'illuminazione è sbilanciata verso le lunghezze d'onda maggiori, ossia verso il rosso, il colore del rame).



Figura 5. Valore del termine di Fresnel per uno specchio di rame, in funzione dell'angolo di incidenza  $\theta$  rispetto allo *halfway vector* e della lunghezza d'onda  $\lambda$ 

La colorazione della luce riflessa è stata introdotta in questo modello fisico di illuminazione nel succitato articolo di Cook e Torrance, i quali osservano i seguenti fatti.

Quando la luce incidente è perpendicolare alla microsfaccettaura, e quindi nella direzione di  $\vec{H}$ , allora  $\theta_i = 0$ , quindi c = 1 e  $g = \eta_{\lambda}$ . Sostituendo tali valori nell'equazione di Fresnel (4.2), otteniamo il termine di Fresnel per  $\theta_i = 0$ :

$$F_{\lambda_0} = \left(\frac{\eta_{\lambda} - 1}{\eta_{\lambda} + 1}\right)^2.$$
(5.1)

Invece, come abbiamo appena osservato alla fine della precedente Sezione 4, quando la luce incidente è radente allora  $\theta_i = \pi/2$ , quindi c = 0, ed il termine di Fresnel per  $\theta_i = \pi/2$  è

$$F_{\lambda_{\pi/2}} = 1. \tag{5.2}$$

Quindi, se la luce è normale alla microsfaccettatura, allora  $F_{\lambda_0}$ , e di conseguenza il coefficiente di riflettività speculare  $\rho_s$ , sono funzioni dell'indice di rifrazione della superficie, che a sua volta varia con la lunghezza d'onda della luce incidente.

Rivediamo invece il caso di luce radente. Come sempre, dobbiamo limitare l'attenzione ad una posizione di osservazione situata di fronte alla posizione della luce, cioè ad un angolo  $\pi = 180^{\circ}$  rispetto a  $\overrightarrow{L}$  (si tratta della configurazione nella quale i versori  $\overrightarrow{L}$  della luce e  $\overrightarrow{V}$  della direzione di osservazione sono entrambi tangenziali alla superficie ed opposti di segno, e quindi lo halfway vector  $\overrightarrow{H} = 1/2(\overrightarrow{L} + \overrightarrow{V})$  coincide con il versore normale N, come già notato nella derivazione della identità (4.4)).

Allora  $F_{\lambda_{\frac{\pi}{2}}}$ , e di conseguenza  $\rho_s$ , valgono 1: quindi  $\rho_s$  non dipende da  $\eta_{\lambda}$ . In altre parole il colore della luce riflessa in questo caso è lo stesso della luce

incidente e non dipende dalla natura fisica del materiale su di cui avviene la riflessione.

Però per ogni angolo di incidenza diverso da  $\pi/2$  la riflettività speculare dipende dal coefficiente di rifrazione  $\eta_{\lambda}$  e quindi ha una distribuzione in frequenza, cioè un colore, diverso da quello della luce incidente

Per le superficie metalliche, sostanzialmente ogni riflessione avviene sulla superficie ed è speculare. Solo per angoli molto vicini a  $\frac{\pi}{2}$  la riflettività speculare non viene influenzata dal materiale su di cui l'oggetto si riflette .

Si noti come in questo caso il coefficiente di riflessione speculare differisca da quello di Phong che è sempre indipendente dal colore dell'oggetto. Cook e Torrance osservano che il modello di Phong funziona bene per i materiali plastici, che difatti sono costituiti da pigmenti immersi in un colloide trasparente. Il colore della plastica è determinato dalla luce che penetra nel colloide trasparente ed interagisce con i pigmenti. Ad esempio, se la luce è bianca ed i pigmenti sono rossi, allora la plastica appare rossa. Invece la luce riflessa si riflette sulla superfice del colloide e non vi penetra dentro: perciò non raggiunge i pigmenti. Pertanto Il colore della luce riflessa non dipende dal colore del materiale (plastica rossa, verde od altro). In altre parole la luce riflessa non viene colorata dalla plastica, e quindi in modello di Phong, che è indipendente dal colore del materiale, in questo caso dà una resa fedele. D'altra parte questo vuol dire che gli oggetti resi con il modello di illuminazione di Phong sembrano tutti di plastica. Si veda, ad esempio, la Figura 5, che confronta, a diversi angoli di incidenza della luce, il rendering di sfere metalliche nei modelli di illuminazione di Phong e di Torrance–Sparrow: nel caso di luce radente, il secondo modello rende molto meglio la tipica riflessione caratteristica dei metalli.

Se si conosce l'indice di rifrazione per diversi valori della lunghezza d'onda, allora lo si può direttamente utilizzare nell'equazione di Fresnel. Di solito non lo si conosce, ma il coefficiente di riflettività al variare della lunghezza d'ondaè stato misurato per molti materiali nel caso incidenza perpendicolare ( $\theta_i = 0$ ). In tal caso si può ricavare il coefficiente di rifrazione  $\eta_{\lambda}$  dall'equazione di Fresnel (4.2):

$$\eta_{\lambda} = \frac{1 + \sqrt{F_{\lambda 0}}}{1 - \sqrt{F_{\lambda 0}}} \tag{5.3}$$

Questo valore  $\eta_{\lambda}$  può allora essere utilizzato nell'equazione di Fresnel per ottenere  $F_{\lambda\theta_i}$  per qualunque altro angolo di incidenza  $\theta_i$ . Poiché questo calcolo è oneroso, Cook e Torrance lo semplificano calcolando un valor medio  $\widetilde{F}_{\lambda\theta}$  del termine di Fresnel, corrispondente ad un valor medio  $\widetilde{\eta}_{\lambda}$  dell'indice di rifrazione corrispondente alla media della riflettività perpendicolare. Il valore così calcolato viene utilizzato per interpolare tra il colore del materiale a  $\theta_i = \pi/2$  ed a  $\theta_i = 0$  per le frequenze corrispondenti per ciascuna delle primarie del modello di colore. A  $\pi/2$  come si visto  $F_{\lambda_{\pi/2}} = 1$  e dunque il colore del materiale è quello della sorgente di luce. Pertanto il colore della luce riflessa a quest'angolo di incidenza è uguale a quello della luce incidente: utilizziamo il modello RGB, consideriamo, ad esempio, la componente rossa di tale colore e la chiamiamo  $Rosso_{\pi/2}$ . Chiamiamo poi  $Rosso_0$  il valore della componente rossa del colore del materiale, ossia della luce riflessa frontalmente ( $\theta_i = 0$ ). La componente  $Rosso_0$  si ottiene nel modo seguente. Si calcolano prima le coordinate CIE della luce incidente. Se la luce incidente arriva frontalmente (cioè ad angolo 0), la luce riflessa che vediamo ha distribuzione spettrale (vale a dire in funzione della lunghezza d'onda)  $F_{\lambda 0}P(\lambda)$ , dove  $P(\lambda)$  è la distribuzione spettrale della luce incidente. Allora le coordinate CIE si ottengono come nella *ricostruzione del modello* CIE-LAB

$$X_{0} = c \int_{\infty}^{\infty} F_{\lambda 0} P(\lambda) \,\overline{x}(\lambda) \, d\lambda$$
$$Y_{0} = c \int_{\infty}^{\infty} F_{\lambda 0} P(\lambda) \,\overline{y}(\lambda) \, d\lambda$$
$$Z_{0} = c \int_{\infty}^{\infty} F_{\lambda 0} P(\lambda) \,\overline{z}(\lambda) \, d\lambda$$

Qui  $\overline{x}, \overline{y} \in \overline{z}$  sono le curve di corrispondenza (color matching) del modello CIE, e c è la costante di normalizzazione che determina la scala in modo che la luce bianca più intensa corrisponda al 100÷ di intensità. Dalle coordinate CIE si risale alla componente rossa grazie alla regola di trasformazione CIE-RGB data dalla matrice J nella sezione Uniformizzazione delle coordinate di colore in un monitor : la trasformazione RGB  $\rightarrow CIE$ 

Più precisamente

$$\begin{pmatrix} Rosso_0\\ Verde_0\\ Blu_0 \end{pmatrix} = J^{-1} \begin{pmatrix} \overline{X}_0\\ \overline{Y}_0\\ \overline{Z}_0 \end{pmatrix}$$

Invece, come abbiamo osservato ripetutamente, a  $\theta_i = \pi/2$  la luce riflessa è radente,  $F_{\lambda_{\pi/2}}=1$ , ed il colore della luce riflessa è lo stesso di quella incidente, ovvero la sua distribuzione spettrale rimane invariata:  $F_{\lambda_{\pi/2}}P(\lambda) = P(\lambda)$ 

Perciò le coordinate CIE sono :

$$X_{\pi/2} = c \int_{\infty}^{\infty} P(\lambda) \,\overline{x}(\lambda) \, d\lambda$$
$$Y_{\pi/2} = c \int_{\infty}^{\infty} P(\lambda) \,\overline{y}(\lambda) \, d\lambda$$

$$Z_{\pi/2} = c \int_{\infty}^{\infty} P(\lambda) \,\overline{z}(\lambda) \, d\lambda$$

ed abbiamo

$$\begin{pmatrix} Rosso_{\pi/2} \\ Verde_{\pi/2} \\ Blu_{\pi/2} \end{pmatrix} = J^{-1} \begin{pmatrix} X_{\pi/2} \\ Y_{\pi/2} \\ Z_{\pi/2} \end{pmatrix}$$

Ora possiamo interpolare tra  $\theta_i = 0$  e  $\theta_i = \pi/2$ . In questo modo si ottiene

$$Rosso_{\theta_i} = Rosso_0 + (Rosso_{\pi/2} + Rosso_0) \frac{\max(0, \widetilde{F}_{\theta_i} - \widetilde{F}_0)}{\widetilde{F}_{\theta_i} - \widetilde{F}_0}$$

Ora sostituiamo  $F_{\lambda} = F_{\lambda\theta_i}$  con  $Rosso_{\theta_i}$  nell'espressione di Cook e Torrance (1.1)per la componente speculare del coefficiente di riflettività bidirezionale. Poichè questa approssimazione si basa su una media rispetto alla lunghezza d'onda, nell'equazione dell'illuminazione il termine speculare deve essere ora moltiplicato per un fattore di scala indipendente dalla lunghezza d'onda, invece che dalla distribuzione spettrale dell'intensità luminosa.

Tipicamente, sia per i materiali conduttori sia per i dielettrici, le componenti ambientale, diffusa e speculare sono del colore dell'oggetto. Per i materiali compositi, come la plastica, abbiamo già visto che le componenti diffusa e speculare sono di colore diverso. I metalli hanno una componente diffusa irrilevante, mentre quella speculare è del colore del metallo per angolo di incidenza  $0^0$  (luce frontale) e del colore della luce incidente per illuminazione radente (angolo di incidenza di  $90^0$  rispetto alla normale).

Polarizzazione della luce: l'equazione di Fresnel risulta valida solo per la luce non polarizzata: ma lo stato di polarizzazione della luce varia quando la luce viene riflessa da una superficie, ed il coefficiente di riflettività di una superficie varia in funzione allo stato di polarizzazione della luce incidente su di essa. Wolff e Kustlander hanno esteso il modello di Cook-Torrance per includere la luce polarizzata (la differenza diventa più evidente dopo due o più inter-riflessioni fra oggetti). In primo luogo consideriamo i materiali dielettrici: per ogni dielettrico esiste un angolo, detto angolo di Brewster, al quale la luce incidente viene completamente polarizzata quando riflessa o non viene affatto riflessa se la sua polarizzazione non è appropriata. Se la riflessione interoggetto di luce inizialmente non polarizzata avviene tra due superfici dielettriche e l'angolo di incidenza di ciascuna riflessione è pari all'angolo di Brewster ed il piano che contiene  $\overrightarrow{N}$  ed  $\overrightarrow{L}$  di una superficie è perpendicolare a quello dell'altra, allora la luce non viene riflessa specularmente dal secondo oggetto; variando angoli ed orientamenti otteniamo ancora un effetto di attenuazione, ma meno accentuato. In secondo luogo consideriamo

i conduttori colorati (metalli come rame o oro, ad esempio): essi tendono a polarizzare la luce in modo diverso a differenti lunghezze d'onda, quindi se un conduttore colorato si riflette su una superficie dielettrica, la riflessione ha colore leggermente diverso rispetto a quello che si ottiene se non si considera la polarizzazione.

# OMBREGGIATURA

#### OMBREGGIATURA

Per rendere immagini realistiche non basta illuminare ciascuna superficie o poligono della scena con un colore fisso. Il colore deve cambiare da punto a punto, generando un'ombreggiatura (*shading*). La variazione di colore e luminosità è conseguenza del fatto che, quando cambiamo il punto osservato, cambia la geometria della scena, e più precisamente le direzioni da questo punto alle luci ed al punto di osservazione. Solo nel caso in cui sia le luci sia l'osservatore sono collocati a distanza infinita i corrispondenti versori rimangono invariati, ed in tal caso si ottiene una colorazione fissa di ogni superficie o poligono, quindi un rendering poco realistico. In generale, invece, dovremmo ricalcolare la geometria punto per punto.

Per accelerare il rendering si usano metodi di ombreggiatura interpolata. I metodi di interpolazione richiedono solo informazioni ai vertici dei poligoni con cui si approssimano le superficie della scena: è solo necessario conoscere di partenza il versore normale alla superficie in ogni vertice della maglia poligonale. Questi dati si possono calcolare in maniera analitica se si conosce la descrizione analitica della superficie (ad esempio l'equazione che la determina); altrimenti essi vengono calcolati in maniera approssimata considerando, per ogni vertice, i versori normali dei poligoni che si toccano a quel vertice (i quali sono forniti dal modellatore oppure calcolati grazie al prodotto vettoriale dei vettori che formano due suoi lati consecutivi), poi calcolando la media aritmetica di tali versori (che di solito non ha più norma 1), ed infine normalizzando.

Ci sono due metodi standard che permettono un eccellente compromesso fra velocità ed accuratezza:

Ombreggiatura interpolata di Gouraud

Ombreggiatura interpolata di Phong

Però ogni metodo di interpolazione comporta qualche errore di interpolazione:

Errori di interpolazione

#### OMBREGGIATURA INTERPOLATA: IL METODO DI GOURAUD

Il metodo di ombreggiatura interpolata di Gouraud, introdotto in [H. Gouraud, <u>Continuous Shading of Curved Surfaces</u>, IEEE Transactions on Computers 20 (1971), 623-628], è il più semplice e rapido possibile. Esso interpola solo dati scalari, non vettoriali, e precisamente le intensità di illuminazione ad ogni vertice, calcolate grazie all'<u>equazione dell'illuminazione</u> del <u>modello di Phong</u>, a partire dalla direzione della sorgente (o delle sorgenti) di luce rispetto a quel vertice e dal suo versore normale (calcolato tramite <u>operazioni di media sui poligoni</u> che condividono quel vertice).

In questo modo l'interpolazione all'interno di un poligono può essere calcolata in maniera incrementale al percorrere successivamente le varie righe di scansione della proiezione sul piano di visuale (*scan conversion*) del poligono, esattamente come per il calcolo della profondità nello <u>z-buffer</u>. In tal modo, se si vuole, il calcolo può essere eseguito nel piano (coordinate dell'osservatore), ed allora non richiede l'aggravio di lavoro dei calcoli nello spazio tridimensionale (coordinate dello spazio ambiente): in questo caso, però, esso diventa a <u>precisione di immagine</u> invece che di oggetto.

In ogni sua variante, questo metodo ottimizza la velocità e l'efficienza. Tuttavia, esso non produce sempre un rendering accurato, soprattutto in presenza di superficie molto speculari. Supponiamo ad esempio che la scena contenga un grande poligono perfettamente speculare (o quasi), abbastanza vicino all'osservatore affinché ai suoi vertici le normali siano quasi perpendicolari alla direzione di osservazione. Allora i vertici sono al buio, ma all'interno del grande poligono ci può essere un punto in cui il versore normale è più favorevole, e cioè a metà strada fra la direzione della luce e quello dell'osservatore. In tal caso in quel punto dovremmo vedere la riflessione speculare della sorgente di luce: ma il metodo di Gouraud interpola fra le luminosità dei vertici, e quindi restituisce bassa luminosità. Ad esempio, nella rendering tramite interpolazione di Gouraud a partire dai vertici del triangolo illuminato al centro della seguente figura, il triangolo apparirebbe nero:



Per evitare questo errore dobbiamo interpolare non i valori della luminosità (scalari), bensì quelli dei versori normali (vettoriali), e poi rieseguire il calcolo dell'illuminazione ad ogni pixel della scan-conversion del poligono. Questo è precisamente ciò che si fa in un metodo di interpolazione più accurato, il metodo di <u>ombreggiatura di Phong</u>.

#### **OMBREGGIATURA INTERPOLATA: IL METODO DI PHONG**

A differenza del <u>metodo di Gouraud</u>, il metodo di ombreggiatura interpolata di Phong Bui-Tong, introdotto in *[Phong Bui-Tong, <u>Illumination for Computer</u> <u>Generated Pictures</u>, Comm. ACM 18 (1975), 311-317], interpola non i valori dell'illuminazione (scalari), bensì i versori normali e della direzione della sorgente di luce (vettoriali), e quindi è più accurato. La seguente figura illustra l'interpolazione, a partire dai vertici della modellazione poligonale, dei versori normale, della direzione della luce e della direzione dell'osservatore.* 



In particolare, questo metodo di interpolazione si presta anche al rendering di superficie speculari, senza dar luogo alle <u>patologie</u> illustrate nella spiegazione del metodo di Gouraud.

Per ogni pixel della scan conversion di un poligono della scena, il metodo di Phong applica l'equazione dell'illuminazione del <u>modello di Phong</u>, a partire dai vertici del poligono, dove vengono calcolati il versore di direzione della sorgente di luce, il versore normale alla superficie, ottenuto tramite <u>operazioni di media sui</u> <u>poligoni</u> che condividono quel vertice, ed il versore della direzione dell'osservatore.

In questo modo l'interpolazione all'interno di un poligono può essere calcolata in maniera incrementale al percorrere successivamente le varie righe di scansione della proiezione sul piano di visuale (*scan conversion*) del poligono, esattamente come per il calcolo della profondità nello <u>z-buffer</u>: però ora si devono interpolare tre vettori, e quindi le loro tre componenti, per un totale di nove interpolazioni: il metodo è quindi almeno nove volte più lento di quello di Gouraud (inoltre le operazioni aritmetiche sono più onerose, in quanto richiedono anche il calcolo di prodotti scalari, e non soltanto moltiplicazioni di numeri).

In tal modo, come nel metodo di Gouraud, il calcolo può essere eseguito nel piano (coordinate dell'osservatore però, esso diventa a <u>precisione di immagine</u> invece che di oggetto.

#### Esercizio sul metodo di ombreggiatura di Phong ed i riflettori di Warn

Consideriamo un triangolo ombreggiato col metodo di Phong. I vertici del triangolo sono  $\mathbf{e_1}$ =(1,0,0),  $\mathbf{e_2}$ =(0,1,0),  $\mathbf{e_3}$ =(0,0,1). La sorgente (puntiforme) di luce di intensità 1 (a ciascuna frequenza monocromatica) è nel punto (0,0,10), ma e' diretta verso un riflettore di Warn di equazione z=11: il riflettore di Warn e' la sorgente di luce per la scena. Si suppongano nulli i termini di illuminazione ambientale e di Lambert, e si pongano uguale a 1 l'esponente di Phong del riflettore e del triangolo. L'osservatore è nel punto (5,1,0). I vettori normali alla superficie (non normalizzati) di cui il triangolo fa parte (calcolati tramite medie tra i vettori perpendicolari a questo triangolo ed a quelli delle maglie adiacenti) sono rispettivamente  $\mathbf{n_1}$ =(1,1,0),  $\mathbf{n_2}$ =(1,0,1),  $\mathbf{n_3}$ =(0,1,1). Quale è l'intensità di illuminazione che l'osservatore vede al baricentro del triangolo (cioe' il punto ottenuto dalla combinazione convessa dei tre vertici con pesi uguali)?

#### Svolgimento

Secondo il modello di ombreggiature di Warn l'intensità luminosa è data da :

 $I_{p\lambda}[L * (-L')]^{p} = I_{p\lambda} \cos^{p} \gamma$ 



dove L è il raggio dal punto osservato al riflettore, L' è il raggio normale al riflettore e passante per / la sorgente di luce,  $\gamma$  è l'angolo compreso tra questi due angoli e *p* è l'esponente di Phong.

Calcoliamo adesso *b* il baricentro del triangolo

 $b = 1/3 \mathbf{e_1} + 1/3 \mathbf{e_2} + 1/3 \mathbf{e_3} = (1/3, 1/3, 1/3)$ 

Il raggio L' dal riflettore alla sorgente di luce è :

$$L' = (0,0,1)$$

Calcoliamo il raggio *L* dal punto osservato *b* al riflettore:

$$\mathbf{L} = \frac{(0,0,11) - (1/3,1/3,1/3)}{\sqrt{(-1/3)^2 + (-1/3)^2 + (32/3)^2}} = \left(\frac{-1}{3\sqrt{114}}, \frac{-1}{3\sqrt{114}}, \frac{32}{3\sqrt{114}}\right)$$

Il raggio V dal baricentro b all'osservatore o è:

$$\mathbf{V} = \frac{(5,0,1) - (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})}{\frac{1}{3}\sqrt{201}} = \left(\frac{14}{9\sqrt{201}}, \frac{1}{9\sqrt{201}}, \frac{2}{9\sqrt{201}}\right)$$

La normale al triangolo sul punto *b* col metodo di illuminazione di Phong viene calcolata per interpolazione con i vettori normali alla superficie di cui il triangolo fa parte:  $n_1$ ,  $n_2$  e  $n_3$ . Otteniamo quindi:

$$\mathbf{N} = \frac{\frac{1}{3}(\mathbf{n}_1 + \mathbf{n}_2 + \mathbf{n}_3)}{\sqrt{x_n^2 + y_n^2 + z_n^2}} = \left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right)$$

Adesso possiamo calcolare il versore R del raggio riflesso:

$$\mathbf{R} = 2 < \mathbf{N}, \mathbf{L} > \mathbf{N} - \mathbf{L} = \frac{20}{\sqrt{3}\sqrt{114}} \left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right) - \left(\frac{-1}{3\sqrt{114}}, \frac{-1}{3\sqrt{114}}, \frac{32}{3\sqrt{114}}\right) = \left(\frac{7}{\sqrt{114}}, \frac{7}{\sqrt{114}}, \frac{-4}{\sqrt{114}}\right)$$

Dal momento che i termini di illuminazione ambientale e di Lambert sono nulli ed il coefficiente di Phong p vale 1, l'intensità luminosa finale risulta ora essere

$$I_{p\lambda} < L, -L' > < R, V > = \frac{32}{\sqrt{3}\sqrt{114}} \frac{10}{\sqrt{3}\sqrt{114}} = \frac{320}{342} = 0.93567.$$

#### ARTEFATTI CAUSATI DAGLI ALGORITMI DI ILLUMINAZIONE INTERPOLATA

Gli algoritmi di ombreggiatura interpolata non danno luogo sempre ad immagini realistiche: essi infatti introducono errori visibili. Ecco un elenco.

- Profili poligonali. Uno di questi errori è il fatto che, interpolando a partire da poligoni, si ottengono si gradazioni ben sfumate di ombreggiatura atte a simulare la rotondità delle superficie, ma i bordi esterni di ciascun oggetto rimangono poligonali, cioè con un profilo fatto di segmenti rettilinei. Questo problema è ineliminabile, ma il suo impatto visivo si può ridurre se si aumenta la risoluzione del modellatore, cioè se si prende una maglia poligonale più fitta. In questo modo però aumenta la mole di calcoli.
- Distorsione prospettica della profondità. Un altro errore di interpolazione è la • distorsione prospettica della profondità. Immaginiamo di voler rendere un pavimento fatto di mattonelle quadrate, il cui colore sfuma dal bianco al nero al crescere della distanza dall'osservatore. Supponiamo che il pavimento. La linee che separano una riga di mattonelle dalla successiva si dispongono, nelle coordinate del monitor, in maniera più fitta man mano che ci si avvicina al punto di fuga prospettico, a causa della divisione prospettica responsabile dello schiacciamento con la profondità. Supponiamo che il pavimento costituisca un unico lungo poligono della modellazione, che comincia a distanza di 100 metri e finisce a distanza di 1000 metri, in modo che l'ultima riga di mattonelle si proietti sul piano di visuale in punti molto vicini al punto di fuga (cioè che la sua proiezione sia un segmento orizzontale molto corto. La riga di metà profondità, a 450 metri, appare nella proiezione non a metà altezza, ma assai più vicina alla riga più alta, di nuovo a causa dell'infittirsi delle righe con la profondità. Ma l'algoritmo di ombreggiatura interpola il colore linearmente rispetto alle coordinate del piano di visuale, dalla riga più alta a quella più bassa, e quindi colloca il colore intermedio, il grigio medio di intensità 50%, esattamente sulla riga che sull'immagine proiettata appare a mezza altezza. L'errore è dovuto al fatto che nelle coordinate del piano di visuale la proiezione prospettica, a causa della proiezione prospettica, non è lineare rispetto alla distanza, mentre invece l'interpolazione lo è. La seguente figura mostra, a sinistra, il pavimento reso per interpolazione a partire dai quattro vertici, ed a destra lo stesso pavimento con il colore che cambia proporzionalmente alla profondità nello spazio tridimensionale. Il segmento orizzontale interno rappresenta la linea del grigio intermedio (50% di luminosità).



Per ridurre questo tipo di distorsione occorre frazionare i poligoni (in questo caso il rettangolo del pavimento) in tanti poligoni più piccoli (od almeno più corti nella direzione della profondità).

• *Dipendenza dall'orientamento*. Un altro errore tipico è la dipendenza dell'interpolazione dall'orientamento dei poligoni. Quando si ruota un poligono, un punto al suo interno, dopo la rotazione, può essere interpolato a partire da due lati diversi rispetto a prima, come nella figura seguente:



In tal caso, il colore risultante può cambiare. Ad esempio, se nella disposizione a sinistra i tre vertici più alti del quadrilatero sono neri ma il vertice in basso è bianco, il colore del punto evidenziato è una combinazione convessa del colore dei tre vertici neri, e quindi è nero. Invece, nella disposizione di destra, esso è una combinazione convessa di due vertici neri ed uno bianco, e quindi è grigio. Il problema nasce dal fatto che in questo caso determiniamo il colore come combinazione convessa dei tre vertici più alti, ma i punti interni del quadrilatero non si esprimono in modo unico come combinazione convessa di tre dei suoi vertici.

Però, se invece di un quadrilatero fossimo partiti con un triangolo, il problema non si sarebbe posto, perché i punti interni di un triangolo si esprimono in modo unico come combinazione convessa dei tre vertici. Quindi questo problema si risolve scindendo tutti i poligoni della modellazione in triangoli. Osserviamo che, se non risolto in questo modo, il problema ha un impatto visivo molto fastidioso nelle animazioni, dove accadrebbe che i punti interni di un oggetto ruotante possano cambiare colore per effetto della rotazione rispetto alle coordinate dell'osservatore. Si noti che questo problema capita non solo nell'interpolazione per l'ombreggiatura, ma anche in tutti i tipi di interpolazione, ad esempio quella per il calcolo della profondit profondità nell'algoritmo di <u>z-buffer</u> o per la determinazione di aree nascoste nell'algoritmo di <u>scan-line</u>.

- Vertici interni non condivisi. Un altro artefatto si ha quando un poligono possiede un vertice che separa due lati allineati, mentre il poligono adiacente vede quei due lati allineati come un unico lato, quindi senza il vertice intermedio. In tal caso, per il primo poligono quel vertice ha un colore assegnato dal modellatore, mentre per il secondo ha il colore interpolato a partire dai due vertici estremi dell'unico lato in cui giace. Quindi in generale il colore è diverso nei due casi, e questo fatto, a causa dell'interpolazione, porta ad una discontinuità del colore su tutto il segmento comune ai due poligoni, un artefatto di terribile impatto visivo perché distrugge la percezione della rotondità che l'ombreggiatura vuole simulare e la trasforma nella percezione di uno strappo. Per risolvere il problema occorre scindere il lato unico del secondo poligono in due nuovi lati allineati, che si congiungono al vertice non condiviso: in altre parole, occorre aggiungere tale vertice anche al database del secondo poligono.
- Aliasing. Infine, un artefatto tipico nasce da problemi di risoluzione insufficiente, cioè di aliasing. Infatti può succedere che i versori normali che il procedimento di interpolazione calcola ai vertici siano poco adatti a rappresentare la superficie. Supponiamo ad esempio che la superfcie nabbai la forma di una lamiera ondulata, diciamo di equazione  $f(x,y)=\sin(\pi x)$ , e che la poligonale che la approssima consista di quadrilateri i cui vertici abbiano ascisse x intere, x=0,1,2,.... Allora i quadrilateri nello spazio hanno estensioni in x pari a  $\pi$ , ma la loro escursione in altezza y vale 2, perche' in un semiperiodo la funzione seno ha escursione 2: più precisamente, all'aumentare di x l'altezza dei quadrilateri alternativamente aumenta o diminuisce di 2 unità. Quindi i versori normali dei quadrilateri giacciono nel piano x-y e puntano verso l'alto con inclinazioni di pendenza pari alternativamente a  $2/\pi$  e  $-2/\pi$ . Quando l'algoritmo di interpolazione provvede a calcolare il valore delle normali ai vertici, esso prende la media di due consecutivi di tali versori, ed ottiene nuovi versori tutti diretti verticalmente verso l'alto, come se la superficie fosse orizzontale piatta. Questo problema, come sempre con l'aliasing, si riduce aumentando la risoluzione, cioè infittendo la partizione.



all'uscita dall'oggetto anche il raggio L3 dovrebbe subire una deviazione dovuta alla rifrazione. Il metodo ignora questa deviazione perchè al momento in cui L3 viene generato si conosce la direzione della sorgente di luce, ma non la direzione verso cui inviare il raggio per farlo arrivare alla sorgente di luce tenendo conto della successiva rifrazione all'uscita del solido. Quest'ultima avviene con un angolo che dipende dall'angolo che il raggio forma con la frontiera a quel punto. Per calcolarla esattamente si potrebbero usare alcuni accorgimenti sul campionamento come inviare non un raggio d'ombra, ma un pennello di raggi, tracciarli tutti e interpolare (questo è un procedimento possibile ma oneroso) o <u>sull'algoritmo stesso</u>. Invece si utilizza un <u>Ray Tracing a ritroso</u>, cioè a partire dalle sorgenti luminose.

## SCORCIATOIE PER IL RENDERING: TESSITURE, RILIEVO E RIFLESSIONI

### MAPPA DI TESSITURA

La mappa di tessitura è un metodo per semplificare il rendering di aree della scena complesse ma di significatività non così rilevante da giustificare un rendering in completo dettaglio, mappando su quelle aree immagini prefissate e poi trasformandole in prospettiva. Il metodo fu introdotto in *[E. Catmull, A Subdivision Algorithm for Computer Display of Curved Surfaces, Ph.D. Thesis, Computer Science Department, University of Utah, Salt Lake City, Utah, US, December 1974]*, e *[J.F.Blinn, M.E.Newell, Texture and Reflection in Computer Generated Images, CACM 19, 1976, 542-547]*. Talvolta l'immagine mappata si chiama (impropriamente) *mappa di tessitura* ed i suoi pixel vengono chiamati *texel.* L'immagine è un rettangolo colorato dotato di un suo sistema di coordinate (u,v). Noi riserviamo il nome *mappa di tessitura* alla trasformazione che fa corrispondere le coordinate (u,v) dell'immagine ad appropriati punti di oggetti della scena.

La mappa viene eseguita nel modo seguente. Consideriamo un pixel del piano di visuale. Le coordinate (x,y) dei suoi vertici vengono mappate su quelle dell'oggetto della scena visibile attraverso quel pixel, ed i punti risultanti vengono trasformati nelle coordinate (u,v) dei punti della immagine di tessitura che gli corrispondono (attraverso la inversa della mappa di tessitura).

In tal modo il pixel viene trasformato in un quadrilatero contenuto nel rettangolo dell'immagine della tessitura, il quale generalmente interseca parecchi texel. Il valore di colore del pixel si ottiene come media pesata di quelli di tali texel, ciascuno pesato in proporzione all'area della parte del texel che giace dentro il quadrilatero.

Se nell'eseguire l'inversa della mappa di tessitura accade che una parte di un pixel finisca fuori del rettangolo dell'immagine, questa si puo' supporre prolungata in maniera periodica.

Quando, come di solito, le superficie della scena sono poligoni, è d'uso assegnare direttamente ai loro vertici le coordinate (u.v) dell'immagine che vi deve essere mappata, per poi trovare le coordinate di tessitura dei punti interni per interpolazione lineare. Ci sono però due problemi. Il primo è il fatto che per poligoni di più di tre lati il risultato dell'interpolazione può dipendere dal modo in cui il poligono è orientato: questo fatto costringe a suddividere tutti i poligoni in triangoli proma di procedere all'interpolazione. Il secondo problema è dovuto alla distanza nella trasformazione l'accorciamento prospettica: l'interpolazione, che avviene riga per riga di scansione, non rimpicciolisce corrispondentemente l'immagine mappata perché due righe consecutive differiscono sempre della stessa altezza, ma si proiettano su curve nel poligono della scena a profondità variabili diverse e non necessariamente equispaziate. L'effetto di distorsione prospettica che ne segue si può ridurre sudividendo i poligoni in poligoni più piccoli; per una soluzione non approssimata occorre effettuare la trasformazione prospettica, punto per punto, nel corso della scansione di interpolazione.

#### MAPPA DI RILIEVO

La mappa di rilievo è un metodo per simulare il rilievo di una superficie, ad esempio la sua rugosità. È analoga alla mappa di tessitura, ma le informazioni mappate non rappresentano colore bensì profondità. Il metodo fu introdotto in *[J.F. Blinn, Simulation of Wrinkled Surfaces, SIGGRAPH 78, in Computer Graphics 12, August 1978, pagg. 286-292].* Una mappa di tessitura che proietta sulla superficie una immagine di rugosità non è adeguata per il rendering, perché la rugosità modifica la direzione del vettore normale punto per punto, e quindi produce un effetto di ombreggiatura che varia da punto a punto con la posizione della sorgente di luce e dell'osservatore: invece una tessitura proiettata sulla superficie mostrerebbe la colorazione indotta dall'immagine proiettata, che varia tutta insieme quale che sia il punto osservato quando si spostano luce o osservatore. Inoltre, la direzione da cui proviene la luce nell'immagine proiettata non è generalmente la stessa che nella scena tridimensionale, e questo fatto genererebbe una aberrazione visibile.

Si deve quindi risolvere il problema di calcolare la direzione del versore normale, o almeno approssimarla, dopo aver applicato la mappa di rilievo. Enunciamo la soluzione senza dimostrazione.

Una superficie tridimensionale è parametrizzata da due parametri reali s, t. La superficie di cui stiamo facendo il rendering sta nella scena tridimensionale: lo spazio tridimensionale ha coordinate x, y, z. Individuiamo un punto **p** sulla superficie con tre mappe, dalle coordinate s,t alle coordinate x, y, z. Quindi scriviamo **p** = (x(s,t), y(s,t), z(s,t)). Assumiamo che questa parametrizzazione sia di classe C<sup>2</sup>, ossia derivabile due volte con continuità rispetto ad entrambe le variabili, ed in particolare differenziabile (pertanto deve esistere il piano tangente a ciascun punto).

Supponiamo di tenere fisso  $t=t_0$  e far variare s. Allora P si muove lungo una curva sulla superficie che si chiama curva di livello  $t=t_0$ . Fissato un qualsiasi valore di s, diciamo  $s=s_0$ , il vettore tangente a questa curva di livello al punto ( $s_0$ ,  $t_0$ ) è la derivata parziale

$$\frac{\partial \mathbf{p}}{\partial \mathbf{s}}(\mathbf{s}_0, \mathbf{t}_0)$$

che per semplicità denotiamo con  $D_1\mathbf{p}(s_0, t_0)$ . Analogamente, per lo stesso punto di parametri  $(s_0, t_0)$  della superficie passa una curva di livello s=costante, in questo caso s=s<sub>0</sub>: si tratta della curva parametrizzata da t variabile e s fisso, s= s<sub>0</sub>, il cui vettore tangente a  $(s_0, t_0)$  è  $D_2\mathbf{p}(s_0, t_0)$ .

Allora un vettore normale **n** alla superficie al punto di parametri  $(s_0, t_0)$  è il prodotto vettore dei due vettori tangenti, poiché deve essere ortogonale ad entrambi:

$$\mathbf{n} = D_1 \mathbf{p}(s_0, t_0) \mathbf{x} D_2 \mathbf{p}(s_0, t_0)$$
(1)
Assumiamo che il prodotto vettore a secondo membro di (1) non si annulli (questo equivale a richiedere che lo Jacobiano della mappa  $\mathbf{p} : \mathbf{R}^2 \to \mathbf{R}^3$  sia non nullo a ciascun punto (s<sub>0</sub>, t<sub>0</sub>), ossia che la rappresentazione parametrica sia localmente invertibile; osserviamo che nell'esempio che stiamo per dare, quello della parametrizzazione di una sfera con gli angoli di Eulero (latitudine e longitudine) questa ipotesi vale ovunque tranne che ai poli Nord e Sud, e quindi questi due punti devono essere omessi dalla parametrizzazione). Il versore normale ovviamente è n/||n||.

Ora introduciamo una mappa di rilievo: ad ogni punto (s, t) la superficie viene elevata, cioè spostata perpendicolarmente alla propria normale, di uno spostamento b(s,t). La funzione b si chiama appunto mappa di rilievo. Perciò la nuova posizione della superficie ai parametri (s<sub>0</sub>, t<sub>0</sub>) è

$$\mathbf{p}'(\mathbf{s}_0, \mathbf{t}_0) = \mathbf{p}(\mathbf{s}_0, \mathbf{t}_0) + \mathbf{b}(\mathbf{s}_0, \mathbf{t}_0) \mathbf{n}(\mathbf{s}_0, \mathbf{t}_0) / ||\mathbf{n}(\mathbf{s}_0, \mathbf{t}_0)||$$
(2)

Ora è necessario calcolare il nuovo versore normale per utilizzarlo nella equazione dell'illuminazione. Quale è il suo valore? Una buona approssimazione si ottiene grazie allo sviluppo di Taylor del primo ordine (ossia al teorema del valor medio di Lagrange) dopo aver sostituito l'identità (2) in (1), ed è

$$\mathbf{n}' = \mathbf{n} + [D_1 b (\mathbf{n} \mathbf{x} D_2 \mathbf{p}) - D_2 b (\mathbf{n} \mathbf{x} D_1 \mathbf{p})] / ||\mathbf{n}||$$
(3)

dove **x** denota il prodotto vettoriale.

Dimostriamo l'identità (3). Derivando (2) rispetto a s e t otteniamo:

$$D_{1}\mathbf{p}'(s_{0}, t_{0}) = D_{1}\mathbf{p}(s_{0}, t_{0}) + D_{1}b(s_{0}, t_{0}) \mathbf{n}/||\mathbf{n}|| + b(s_{0}, t_{0}) D_{1}(\mathbf{n}/||\mathbf{n}||)$$
(4)  
$$D_{2}\mathbf{p}'(s_{0}, t_{0}) = D_{2}\mathbf{p}(s_{0}, t_{0}) + D_{2}b(s_{0}, t_{0}) \mathbf{n}/||\mathbf{n}|| + b(s_{0}, t_{0}) D_{2}(\mathbf{n}/||\mathbf{n}||)$$
(4)

Poiché la parametrizzazione è di classe  $C^2$ , il vettore  $\mathbf{n} = D_1 \mathbf{p}(s_0, t_0) \mathbf{x} D_2 \mathbf{p}(s_0, t_0)$ è derivabile con continuità rispetto alle variabili s e t, e quindi lo stesso succede per  $\mathbf{n}/||\mathbf{n}||$ , che giace nella sfera unitaria, la quale è compatta. Pertanto  $D_1(\mathbf{n}/||\mathbf{n}||)$ e  $D_2(\mathbf{n}/||\mathbf{n}||)$  sono continue su un compatto, quindi limitate. Inoltre, naturalmente, lo spostamento b deve essere piccolo. Combinando queste due osservazioni vediamo che l'ultimo termine al lato di destra di (4) e (4') può essere trascurato. Pertanto

$$\mathbf{n'}(s_0, t_0) = D_1 \mathbf{p'}(s_0, t_0) \times D_2 \mathbf{p'}(s_0, t_0) = = D_1 \mathbf{p}(s_0, t_0) \times D_2 \mathbf{p}(s_0, t_0) + D_1 \mathbf{b}(s_0, t_0) (\mathbf{n/||n||} \times D_2 \mathbf{p}(s_0, t_0)) + D_2 \mathbf{b}(s_0, t_0) (D_1 \mathbf{p}(s_0, t_0) \times \mathbf{n/||n||}) + D_1 \mathbf{b}(s_0, t_0) D_2 \mathbf{b}(s_0, t_0) \mathbf{n} \times \mathbf{n/||n||}^2 .$$

Ovviamente l'ultimo termine sul lato destro si annulla, ed il primo vale **n**, per (2). Quindi

$$\mathbf{n}' - \mathbf{n} = [D_1 b (\mathbf{n} \mathbf{x} D_2 \mathbf{p}) + D_2 b (D_1 \mathbf{p} \mathbf{x} \mathbf{n})]/||\mathbf{n}|| \\ = [D_1 b (\mathbf{n} \mathbf{x} D_2 \mathbf{p}) - D_2 b (\mathbf{n} \mathbf{x} D_1 \mathbf{p})]/||\mathbf{n}|| ,$$

ossia (3).

Nelle identità (2) e (3) tutti i vettori sono calcolati allo stesso punto della superficie, parametrizzato da ( $s_0$ ,  $t_0$ ). Abbiamo implicitamente supposto che la mappa di rilievo b sia parametrizzata dalle stesse coordinate s, t con le quali abbiamo parametrizzato la superficie. In realtà la mappa di rilievo di solito usa sue coordinate specifiche u, v, come accade per la mappa di tessitura: b viene memorizzata e richiamata come una immagine rettangolare con proprie coordinate (ed in tal modo può essere usata come tessitura per diverse superficie della scena, o per scene diverse). Quando b viene mappata sulla superficie, viene stabilita una corrispondenza biunivoca fra le coordinate u, v di b e s, t della superficie: cioè si definisce una mappa biunivoca

$$u = u(s,t), v = v(s,t)$$
 (5)

Implicitamente, quindi, nel supporre che b in (3) sia parametrizzato da s, t come la superficie equivale ad identificare le sue coordinate u, v con s, t tramite (5). Pertanto il modo esplicito di scrivere (3) è:

$$\begin{split} &\mathbf{n}'(\mathbf{s}_{0}, t_{0}) = \mathbf{n}(\mathbf{s}_{0}, t_{0}) + \\ &+ \frac{\mathbf{D}_{1} \mathbf{b} \left( \mathbf{u}(\mathbf{s}_{0}, t_{0}), \mathbf{v}(\mathbf{s}_{0}, t_{0}) \right) \times \left( \mathbf{n}(\mathbf{s}_{0}, t_{0}) \times \mathbf{D}_{2} \mathbf{p}(\mathbf{s}_{0}, t_{0}) \right) - \mathbf{D}_{2} \mathbf{b} \left( \mathbf{u}(\mathbf{s}_{0}, t_{0}), \mathbf{v}(\mathbf{s}_{0}, t_{0}) \right) \times \left( \mathbf{n}(\mathbf{s}_{0}, t_{0}) \times \mathbf{D}_{1} \mathbf{p}(\mathbf{s}_{0}, t_{0}) \right)}{\|\mathbf{n}(\mathbf{s}_{0}, t_{0})\|} \\ &= \mathbf{n} \left( \mathbf{s}_{0}, t_{0} \right) \\ &+ \frac{\left( \frac{\partial b}{\partial u} \left( \mathbf{u}(\mathbf{s}_{0}, t_{0}), \mathbf{v}(\mathbf{s}_{0}, t_{0}) \right) \frac{\partial u}{\partial s}(\mathbf{s}_{0}, t_{0}) + \frac{\partial b}{\partial v} \left( \mathbf{u}(\mathbf{s}_{0}, t_{0}), \mathbf{v}(\mathbf{s}_{0}, t_{0}) \right) \frac{\partial v}{\partial s}(\mathbf{s}_{0}, t_{0}) \right) \times \left( \mathbf{n}(\mathbf{s}_{0}, t_{0}) \times \mathbf{D}_{2} \mathbf{p}(\mathbf{s}_{0}, t_{0}) \right)}{\|\mathbf{n}(\mathbf{s}_{0}, t_{0})\|} \\ &- \frac{\left( \frac{\partial b}{\partial u} \left( \mathbf{u}(\mathbf{s}_{0}, t_{0}), \mathbf{v}(\mathbf{s}_{0}, t_{0}) \right) \frac{\partial u}{\partial t}(\mathbf{s}_{0}, t_{0}) + \frac{\partial b}{\partial v} \left( \mathbf{u}(\mathbf{s}_{0}, t_{0}), \mathbf{v}(\mathbf{s}_{0}, t_{0}) \right) \frac{\partial v}{\partial t}(\mathbf{s}_{0}, t_{0}) \right) \times \left( \mathbf{n}(\mathbf{s}_{0}, t_{0}) \times \mathbf{D}_{1} \mathbf{p}(\mathbf{s}_{0}, t_{0}) \right)}{\|\mathbf{n}(\mathbf{s}_{0}, t_{0})\|} \end{split}$$

# Nota: Fattore locale di ingrandimento delle aree nella parametrizzazione di una superficie, e calcolo dell'area di una superficie parametrica

La norma del vettore  $\mathbf{n} = D_1 \mathbf{p}(s_0, t_0) \mathbf{x} D_2 \mathbf{p}(s_0, t_0)$  è data da  $||D_1\mathbf{p}|| ||D_2\mathbf{p}|| |\sin \theta|$ , dove  $\theta$  è l'angolo formato dai due vettori tangenti  $D_1\mathbf{p}$  e  $D_2\mathbf{p}$ . Questi due vettori sono tangenti alle curve s=costante e t=costante sulla superficie parametrica  $\mathbf{p}(s,t) = (p_1 = p_1(s,t), p_2 = p_2(s,t), p_3 = p_3(s,t))$ . La lunghezza di questi due vettori tangenti misura la velocita' di percorrenza delle curve s=costante e t=costante data dalla parametrizzazione scelta. Grazie al fattore |sin  $\theta$ |, quindi, la norma dello Jacobiano || $D_1\mathbf{p}(s_0, t_0) \mathbf{x} D_2\mathbf{p}(s_0, t_0)$ || è esattamente l'area del parallelogramma generato dai due vettori tangenti  $D_1\mathbf{p}$  e  $D_2\mathbf{p}$ , e pertanto è il fattore locale (al punto ( $s_0, t_0$ ) ) di ingrandimento delle aree dato dalla parametrizzazione. In altre parole, se  $\mathbf{p}$  è definita su un rettangolo K, l'area della superficie parametrica  $\mathbf{p}(K)$  è data da

$$\iint D_1 p(s,t) \times D_2 p(s,t) \parallel ds dt$$

Nel caso particolare della parametrizzazione semplice data dal grafico di una funzione f delle variabili s e t, ossia  $\mathbf{p}(s,t)=(s,t, f(s,t))$ , il fattore di ingrandimento delle aree si calcola facilmente. Infatti, come abbiamo visto esso è dato dal prodotto delle norme dei vettori velocità di percorrenza delle curve s=costante e t=costante per il seno dell'angolo che questi due vettori formano. Ma per la scelta della parametrizzazione, ossia  $\mathbf{p}(s,t)=(s,t,f(s,t))$ , i due vettori velocità sono rispettivamente

$$D_1 p(s,t) = (1, 0, \frac{\partial f}{\partial s}(s,t))$$

е

$$\mathbf{D}_{2}\mathbf{p}(\mathbf{s},\mathbf{t}) = \left(0, 1, \frac{\partial \mathbf{f}}{\partial t}(\mathbf{s},\mathbf{t})\right).$$

Pertanto

$$\left\| D_{1}p\left( s,t\right) \times D_{2}p\left( s,t\right) \right\| = \sqrt{1 + \left(\frac{\partial f}{\partial s}\right)^{2} + \left(\frac{\partial f}{\partial t}\right)^{2}}$$

*Esempio 1: area della calotta sferica*. Consideriamo la calotta sferica con disco di base di raggio r<sub>0</sub> nella superficie sferica di raggio R, ossia il grafico della funzione

$$f(s,t) = \sqrt{R^2 - s^2 - t^2}$$
 nel disco  $D = \{s^2 + t^2 \le r_0^2\}$ 

La formula precedente ora porta a

$$||D_1 \mathbf{p} \times D_2 \mathbf{p(s,t)}|| = \sqrt{1 + \frac{s^2 + t^2}{R^2 - s^2 - t^2}} = \sqrt{\frac{1}{1 - \left(\frac{s}{R}\right)^2 - \left(\frac{t}{R}\right)^2}}$$

Passando in coordinate polari s = r cos  $\theta$ , t = r sin  $\theta$ , l'ultima espressione diventa

$$\sqrt{\frac{1}{1 - \left(\frac{s}{R}\right)^2 - \left(\frac{t}{R}\right)^2}} = \sqrt{\frac{1}{1 - \left(\frac{r}{R}\right)^2 \cos^2 \theta - \left(\frac{r}{R}\right)^2 \sin^2 \theta}} = \sqrt{\frac{1}{1 - \left(\frac{r}{R}\right)^2}} = \sqrt{\frac{R^2}{R^2 - r^2}}$$

Pertanto l'area della calotta sferica è

$$\int_{0}^{2\pi} \int_{0}^{r_{0}} \sqrt{\frac{R^{2}}{R^{2} - r^{2}}} r \, dr \, d\theta = \pi \int_{0}^{r_{0}^{2}} \sqrt{\frac{R^{2}}{R^{2} - u}} \, du = \pi \int_{0}^{r_{0}^{2}} \sqrt{\frac{1}{1 - u/R^{2}}} \, du$$
$$= \pi R^{2} \int_{0}^{r_{0}^{2}/R^{2}} \sqrt{\frac{1}{1 - v}} \, dv = 2\pi R^{2} \sqrt{1 - v} \Big|_{r_{0}^{2}/R^{2}}^{0} = 2\pi R^{2} \left( 1 - \sqrt{1 - \frac{r_{0}^{2}}{R^{2}}} \right)$$

*Esempio 2: area del cono retto.* Consideriamo il cono retto con altezza h e disco di base di raggio R, ossia il grafico della funzione

$$f(s,t) = h\left(1 - \sqrt{(s/R)^2 + (t/R)^2}\right)$$
 nel disco  $D = \{s^2 + t^2 \le R^2\}$ 

Ripetendo i calcoli precedenti ora si ottiene il fattore di ingrandimento seguente:

$$||D_1\mathbf{p} \times D_2\mathbf{p(s,t)}|| = \sqrt{1 + \frac{h^2}{R^2} \left(\frac{s^2}{s^2 + t^2} + \frac{t^2}{s^2 + t^2}\right)} = \sqrt{1 + \frac{h^2}{R^2}}$$

Quindi l'area del cono è

$$\sqrt{1 + \frac{h^2}{R^2}} \int_0^{2\pi} \int_0^R r \, dr \, d\theta = \sqrt{1 + \frac{h^2}{R^2}} \text{ area } (D) = \pi R^2 \sqrt{1 + \frac{h^2}{R^2}} = \pi R \sqrt{R^2 + h^2} = \pi R A,$$

dove abbiamo denotato con A la distanza dal vertice del cono ai punti della circonferenza di base (data da  $\sqrt{R^2 + h^2}$  in base al teorema di Pitagora).

# ESEMPI ED ESERCIZI SULLE MAPPE DI TESSITURA E DI RILIEVO

# Esempio: la superficie sferica e la parametrizzazione con gli angoli di Eulero

La sfera con centro l'origine e raggio r si può parametrizzare tramite gli angoli di Eulero. Nella versione che scegliamo adesso, la parametrizzazione di Eulero ricalca la descrizione angolare della sfera in termini degli angoli di latitudine e di longitudine determinati dalla scelta di un asse polare: in particolare, la mappa  $\mathbf{p} = (x(\theta, \phi), y(\theta, \phi), z(\theta, \phi))$  è data da

 $x = r \cos\theta \cos\phi$ ,  $y = r \sin\theta \cos\phi$ ,  $z = r \sin\phi$  (6)

dove r è il raggio della sfera,  $\phi$  varia in [- $\pi/2$ ,  $\pi/2$ ] ed è la latitudine, e  $\theta$  varia in [0,  $2\pi$ ] ed è la longitudine.

Si osservi che questa parametrizzazione non è biunivoca al polo nord ed al polo sud, cioè per  $\phi = -\pi/2$  e  $\phi = \pi/2$ . In effetti, ciascuno dei paralleli dati da  $\phi = -\pi/2$ e  $\phi = \pi/2$  si condensa in un unico punto, il polo nord o il polo sud, e quindi le curve di livello  $\phi = -\pi/2$  e  $\phi = \pi/2$  sono costanti. Perciò D<sub>1</sub>**p**( $\theta$ ,  $-\pi/2$ ) = 0 = D<sub>1</sub>**p**( $\theta$ ,  $\pi/2$ ) per ogni  $\theta$ , e quindi, da (1), **n**( $\theta$ ,  $-\pi/2$ ) e **n**( $\theta$ ,  $\pi/2$ ) sono nulli. Ovviamente i versori normali ai poli nord e sud della superficie sferica non sono affatto nulli (sono i due versori verticali), ma in questa parametrizzazione il calcolo dà risultato nullo perché la parametrizzazione non è biunivoca in corrispondenza dei poli, e quindi non è una parametrizzazione ammissibile.

In tutti gli altri punti, invece, la parametrizzazione è biunivoca.

#### Esercizio 1.

Determinare il versore normale e mostrare che coincide con il versore radiale.



#### Svolgimento.

Calcoliamo  $D_1 \mathbf{p}$  e  $D_2 \mathbf{p}$ . Indicando con **i**, **j**, **k** i versori canonici dei tre assi, otteniamo

 $D_{1}\mathbf{p}(\theta, \phi) = (-r \sin\theta \cos\phi, r \cos\theta \cos\phi, 0) = r (-\sin\theta \cos\phi \mathbf{i} + \cos\theta \cos\phi \mathbf{j})$  $D_{2}\mathbf{p}(\theta, \phi) = (-r \cos\theta \sin\phi, -r \sin\theta \sin\phi, r \cos\phi)$  $= r (-\cos\theta \sin\phi \mathbf{i} - \sin\theta \sin\phi \mathbf{j}) + r \cos\phi \mathbf{k}$ 

Da qui si ottiene

$$\mathbf{n} = D_1 \mathbf{p}(\theta, \phi) \mathbf{x} D_2 \mathbf{p}(\theta, \phi) = r^2 (\sin^2 \theta \sin \phi \cos \phi \mathbf{i} \mathbf{x} \mathbf{j} - \sin \theta \cos^2 \phi \mathbf{i} \mathbf{x} \mathbf{k} - \cos^2 \theta \sin \phi \cos \phi \mathbf{j} \mathbf{x} \mathbf{i} + \cos \theta \cos^2 \phi \mathbf{j} \mathbf{x} \mathbf{k}) = r^2 (\sin \phi \cos \phi \mathbf{k} + \sin \theta \cos^2 \phi \mathbf{j} + \cos \theta \cos^2 \phi \mathbf{i})$$

perché i x j = k = - j x i, j = k x i = - i x k, e i = j x k. Ora è elementare calcolare la norma di n =  $D_1p(\theta, \phi) \times D_2p(\theta, \phi)$ . Si trova

 $||\mathbf{n}(\theta, \phi)|| = r^2 \cos\phi$ 

(si osservi che il coseno è non negativo in  $[-\pi/2, \pi/2]$ ). Pertanto

 $\mathbf{n}(\theta, \phi)/||\mathbf{n}(\theta, \phi)|| = \cos\theta\cos\phi \mathbf{i} + \sin\theta\cos\phi \mathbf{j} + \sin\phi \mathbf{k} = \mathbf{p}(\theta, \phi)/r$ 

è il versore diretto radialmente, esattamente come ci si aspettava.

*Nota:* spesso si usa una parametrizzazione equivalente ma diversa, nella quale l'angolo  $\phi$  è il *complementare* dell'angolo di latitudine. La scelta degli angoli diventa quindi come in questa figura:



Si noti che l'angolo  $\phi$  ora varia da 0 (polo nord) a  $\pi$  (polo sud). Ovviamente, la parametrizzazione corrispondente alla nuova figura, ottenuta scambiando  $\phi$  con il suo complementare, si ricava scambiando fra loro  $sin\phi$  e  $cos\phi$ , e quindi è

 $x = r \cos\theta \sin\phi$ ,  $y = r \sin\theta \sin\phi$ ,  $z = r \cos\phi$  (6')

#### Esercizio 2.

Il piano di coordinate u,v viene munito di una tessitura data da un gradiente che varia linearmente dal nero al punto (0,0) al bianco al punto (1,1): per i punti di tale piano con proiezione negativa sull'asse di questo gradiente (ossia la bisettrice del primo e terzo quadrante) il colore è nero, per quelli per cui la proiezione è >1 il colore è bianco.

Consideriamo la sfera di raggio 1 e centro l'origine, parametrizzata con gli angoli di Eulero  $\phi$  di latitudine (con  $\phi$  = 0 al polo Nord e  $\pi$  al polo Sud) e  $\theta$  di longitudine (con  $\theta$  = 0 o  $\pi$  sul piano y=0).

1. Applichiamo su questa sfera la tessitura, mappando il quadrato  $0 \le u \le 1$ ,  $0 \le v \le 1$  sulla sfera tramite la mappa di tessitura data da  $\phi = \pi u$ ,  $\theta = 2 \pi v$ . Quali punti della sfera risultano di colore grigio con intensità 50%?

2. Ruotiamo la tessitura di 45 gradi in senso orario, ossia ruotiamo l'asse del gradiente in modo che coincida con l'asse u, e dopo riapplichiamo la mappa di tessitura dal quadrato [0,1]x[0,1] alla sfera, definita come prima. Ora quali sono i punti della sfera grigi al 50%?

**Esercizio 3.** Per la sfera del problema precedente, scrivere le coordinate del versore normale esterno al punto generico  $\phi$ ,  $\theta$ , ed applicare la mappa di rilievo  $b(\phi, \theta) = \sin^2(\phi)/10$ . Come cambia il versore normale esterno? Scrivere le sue nuove coordinate, con la consueta approssimazione di Taylor data dalla mappa di rilievo.

Svolgimento. Basta usare la formula di Blinn. La mappa  $\mathbf{p} = (\mathbf{x}(\theta, \phi), \mathbf{y}(\theta, \phi), \mathbf{z}(\theta, \phi))$  è data da  $\mathbf{x} = r \cos\theta \cos\phi$ ,  $\mathbf{y} = r \sin\theta \cos\phi$ ,  $\mathbf{z} = r \sin\phi$ . Poiché la superficie è una sfera, la sua normale  $\mathbf{N}$  è il versore radiale  $\mathbf{N}=\mathbf{p}/||\mathbf{p}||$ . Il prodotto vettore  $\mathbf{n} = D_{\theta}\mathbf{p} \times D_{\phi}\mathbf{p}$  è diretto radialmente, e quindi  $\mathbf{N}=\mathbf{n}/||\mathbf{n}||$ . Sappiamo che, per la mappa  $\mathbf{p}$  (una versione della parametrizzazione di Eulero per la sfera) si ha  $||\mathbf{n}|| = \cos \phi$ .

In base alla formula di Blinn, dopo aver spostato p in p+bN, la nuova normale diventa, al primo ordine di sviluppo di Taylor,

$$N' = N + \frac{D_{\theta}b(n \times D_{\phi}p) - D_{\phi}b(n \times D_{\theta}p)}{\parallel n \parallel}$$

Osservando che  $D_{\theta}b = 0,$   $D_{\phi}b = \sin \phi \cos \phi / 5,$   $D_{\theta}p = (-\sin \theta \cos \phi, \cos \theta \cos \phi, 0),$  $D_{\phi}b = (-\cos \theta \sin \phi, -\sin \theta \sin \phi, \cos \phi)$ 

si arriva facilmente alla soluzione, che è lasciata al lettore.

## MAPPA DI RIFLESSIONE

Quando un oggetto della scena ne riflette altri si parla di riflessione fra oggetti, o interriflessione. L'effetto può variare fra riflessioni speculari il cui risultato cambia sensibilmente con la posizione dell'osservatore a riflessioni diffuse che non ne dipendono. Il primo caso viene reso in maniera realistica dal ray tracing ricorsivo a numerose generazioni di raggi, il secondo dalla radiosità. Entrambi guesti procedimenti sono pero' estremamente dispendiosi in termini di mole di calcoli. Per quanto concerne la interriflessione speculare, una scorciatoia che rappresenta un buon compromesso fra velocità di esecuzione e precisione del rendering è rappresentata dalla mappa di riflessione, introdotta in [James F. Blinn, Martin E. Newell, Texture and Reflection in Computer Generated Images. Comm.ACM 19, Ottobre 1976, 542-547]. Il procedimento è il seguente. Si immagina di inscrivere l'intero ambiente in una grande superficie sferica, con centro prefissato nell'ambiente in vicinanza al centro della scena, se identificabile. Questa sfera contiene tutti gli oggetti al suo interno. Dal centro si esegue la proiezione radiale di ciascun oggetto sulla superficie della sfera, che quindi diventa una tessitura bidimensionale: ciascun texel viene colorato con il colore (e l'ombreggiatura) del punto dell'oggetto più vicino al centro della sfera che la proiezione mappa su quel texel (più vicino al centro della sfera, ossia più lontano dalla sfera stessa). Il colore di tale punto si potrebbe ottenere, ad esempio, mediante il metodo di illuminazione di Phong e l'ombreggiatura di Phong oppure di Gouraud: questa procedura, però, dà origine ad una aberrazione dell'illuminazione (ossia dei valori di luminosità in ogni canale, e quindi del colore), perché il metodo di illuminazione di Phong utilizza l'illuminazione come vista dall'osservatore, mentre in questo caso dobbiamo calcolare invece l'illuminazione come vista dal centro della sfera. Pertanto il meccanismo corretto consiste nel proiettare radialmente sulla sfera il valore dell'illuminazione del punto più vicino al centro della sfera lungo quel raggio, calcolato considerando la posizione delle sorgenti di luce ma immaginando che l'osservatore sia spostato al centro della sfera. Evitare questo spostamento è una semplificazione che talvolta fa risparmiare tempo, perché l'illuminazione vista dalla posizione dell'osservatore a molti punti della scena viene comunque calcolata durante il ray tracing ricorsivo, e potrebbe essere memorizzata per riutilizzo successivo: ma come si è visto introduce una aberrazione cromatica e di luminosità.

Questa tessitura (mappa di riflessione) viene usata come la mappa di tessitura: quando si esegue il rendering di un punto su un oggetto, si calcola il <u>versore</u> <u>riflesso</u>, cioè la riflessione del versore di visuale (direzione dell'osservatore, considerato nella sua posizione vera, non quella spostata al punto osservato) rispetto al versore normale del punto osservato, e lo si usa come puntatore, ossia si recupera il colore da quello del punto della sfera puntato da quel versore (pensato come vettore uscente dal centro della sfera). Pensato come vettore applicato al punto osservato (quindi in un contesto di spazi affini), questo versore indica in che direzione dovremmo guardare, dal punto osservato, per vedere cosa colpisce la direzione speculare della direzione incidente data dalla posizione dell'osservatore. La mappa di riflessione contiene proprio l'illuminazione vista guardando dal punto osservato verso tale direzione speculare. Pertanto, i coseni direttori del versore riflesso vengono usati come indici per la mappa di riflessione (a questo scopo la cosa più pratica è di scrivere le coordinate del versore riflesso in termini degli angoli di Eulero, cioè degli angoli di latitudine e longitudine sulla sfera: questi angoli indicizzano in maniera biunivoca i punti della sfera eccetto che ai poli nord e sud). Questo procedimento equivale a visualizzare la scena a partire da un suo punto di proiezione centrale (diverso da quello dell'osservatore, che non è centrale ma frontale) come se fosse proiettata su una carta geografica.

Una variante di questa parametrizzazione con gli angoli di Eulero, introdotta in [Roy A. Hall, <u>Hybrid Techniques for Rapid Image Synthesis</u>, in Image Rendering Tricks, Course Notes 16 for SIGGRAPH 86, T. Whitted e R. Cook (editors), Dallas, TX, Agosto 1986], consiste nel sostituire l'angolo di latitudine  $\phi$  con sin $\phi$ : in tal modo, poiché la misura equidistribuita sulla sfera in termini degli angoli di Eulero si esprime come r sin $\phi$  d $\theta$  d $\phi$ , la proiezione preserva le aree.

Una variante ancora più utile e comunemente usata consiste nel proiettare non su una sfera ma su un cubo. In tal modo, la determinazione del colore dei texel delle sei facce del cubo si ottiene tramite altrettanti <u>z-buffer</u>, facilmente implementabili via hardware e veloci. Si deve solo fare attenzione ad orientare correttamente gli z-buffer, in maniera che i lati combacianti delle sei facce corrispondano.

La mappa di riflessione introduce distorsioni di parallasse. Infatti, nel determinare il colore di un punto, non si segue il raggio riflesso a partire da quel punto (questo coinciderebbe col metodo di ray tracing), ma si utilizzano i coseni direttori di tale versore come puntatore alla mappa di riflessione, in altre parole si segue la direzione del versore riflesso applicato al centro di proiezione (centro della sfera o del cubo). Quanto maggiore è la distanza fra il punto osservato ed il centro di proiezione, tanto maggiore è l'errore di parallasse. Solo se la sfera ha raggio infinito (cioè se l'intera mappatura avviene nel senso della geometria proiettiva) l'errore di parallasse si annulla. Da questa osservazione segue la necessità di scegliere il raggio della sfera (o cubo) molto grande rispetto alle dimensioni della scena, ed il suo centro molto prossimo al baricentro della scena. In ogni caso, le distorsioni aumentano quando si rende la riflessione di altri oggetti su una superficie della scena lontana dal centro. Per questo motivo spesso si usa più di una mappa di riflessione: si scelgono varie sfere, con centri diversi in prossimità delle superficie riflettenti, e per ciascuna superficie si utilizza la mappa di riflessione sulla sfera con il centro più vicino.

Si deve comunque osservare che la riflessione su una superficie curva è comunque soggetta a distorsione geometrica: le forme vengono distorte dalla

curvatura. Quindi l'effetto visivo della ulteriore distorsione causata dall'errore di parallasse è assai ridotto, tranne nel caso in cui la superficie sia uno specchio piano. D'altra parte, per specchi piani di piccole dimensioni, l'area da rendere è piccola e quindi una distorsione ha poca influenza sul realismo visivo. Invece, per specchi piani grandi, la mappa di riflessione è poco utile e di solito non impiegata, perché la direzione del versore riflesso varia molto lentamente e quindi esso punta più o meno ovunque quasi allo stesso punto della mappa: in tal caso si avrebbe una colorazione quasi costante, mentre nella realtà (e nel ray tracing ricorsivo, che non compie errori di parallasse) la riflessione su uno specchio crea una scena virtuale con colorazione simmetrica a quella della scena originale.

#### Esercizi sulla mappa di riflessione

Nei prossimi esercizi ignoriamo il fatto che i valori di illuminazione nella mappa di riflessione si debbano ottenere proiettando i valori di illuminazione della scena reale calcolati immaginando l'osservatore spostato al centro della sfera di proiezione. In effetti, questi esercizi mirano solo all'esame della parallasse geometrica, e quindi, per semplicità, dipingiamo artificialmente la mappa di riflessione, ossia dipingiamo la sfera (con meridiani e paralleli, numerati).

**Esercizio 1**. Una scena consiste di una sfera S centro l'origine e raggio 1, dalle pareti perfettamente speculari, ed una sfera B con centro l'origine e raggio 10, dipinta con i meridiani ed i paralleli, come un mappamondo. L'osservatore si trova al punto (2,0,0).

- Quale parte di B (in termini di meridiani e paralleli) l'osservatore vede riflessa in S? (Suggerimento: usare la parametrizzazione della sfera tramite gli angoli di Eulero: x=r cos θ cos φ, y=r sin θ cos φ, z = r sin φ, dove r e' il raggio della sfera, φ varia in [-π/2, π/2] ed e' la latitudine, e θ varia in [0, 2π] ed e' la longitudine).
- 2. Se l'osservatore guarda la sfera S con una deviazione angolare di angolo  $\psi$  (piccolo), quali punti della sfera B vi vede riflessi? Ossia, determinare l'andamento dell'angolo  $\varphi$  di apertura angolare del cono che descrive i punti della sfera B visti quando l'osservatore guarda S con una deviazione angolare  $\psi$ . In particolare, determinare l'andamento di  $\theta$  quando  $\psi$  è piccolo (ossia sviluppare secondo Taylor  $\varphi(\psi)$  intorno a  $\psi = 0$ ).
- 3. Valutare l'andamento di  $\varphi$  rispetto a  $\psi$  in prossimità dell'angolo di tangenza: l'ordine di divergenza di tale andamento misura la distorsione geometrica massima della mappa di riflessione, ossia il fattore di ampliamento fra area guardata e corrispondente area riflessa.

#### Svolgimento.

#### Parte 1.

Per simmetria circolare (la parte di *B* osservata attraverso *S* ha gli stessi estremi sia sull'asse delle *y* che sull'asse delle *z*) si riduce il problema a due dimensioni sul piano xy:



L'equazione della sfera centrata nell'origine, intersecata con il piano xy, è  $x^2 + y^2 = 1$ 

Consideriamo ora l'equazione di un fascio di rette passanti per un punto (x<sub>0</sub>, y<sub>0</sub>):

$$(y-y_0) = m(x-x_0)$$

Troviamo l'equazione delle rette passanti per il punto o (osservatore) e troviamo i due punti di intersezione di ciascuna di queste rette con la sfera mettendo a sistema le due equazioni. Imponendo che  $\Delta = 0$  si trovano le due rette con un unico punto di intersezione, quindi tangenti alla sfera *S*. A questo punto troviamo la loro intersezione con *B*. Svolgiamo i passaggi uno ad uno:

$$\begin{cases} x^{2} + y^{2} = 1\\ y = m(x-2) \end{cases}$$
$$\begin{cases} x^{2} + [m(x-2)]^{2} = 1\\ y = m(x-2) \end{cases}$$
$$\begin{cases} (1+m^{2})x^{2} - 4m^{2}x + 4m^{2} = 0\\ y = m(x-2) \end{cases}$$

Ora nella prima equazione poniamo  $\Delta = 0$ :

$$\Delta = 16m^4 - 4(1+m^2)(4m^2 - 1) = 0$$

da cui

$$m = \pm \frac{1}{\sqrt{3}}$$

Le due rette soluzione del sistema sono:

$$y_1 = +\frac{1}{\sqrt{3}}(x-2)$$
;  $y_2 = -\frac{1}{\sqrt{3}}(x-2)$ 

L'intersezione delle rette con la sfera esterna B è la soluzione del sistema

$$\begin{cases} x^2 + y^2 = 100\\ y = \frac{1}{\sqrt{3}}(x - 2) \end{cases}$$

Ci sono due soluzioni:

$$x_{+} = 9,13$$
 ;  $x_{-} = -8,13$ 

L'osservatore guarda nella direzione delle x negative: quindi delle due soluzioni quella positiva è alle sue spalle, e quella da scegliere è quella più negativa, x. Pertanto le due rette intersecano B nei punti

$$b_{+} = (x_{-}, y_{+}) = (-8, 13; 5, 85)$$
 e  $b_{-} = (x_{-}, y_{-}) = (-8, 13; -5, 85)$ .

Adesso, per determinare l'angoli solido richiesto, si ricorre alla parametrizzazione con gli angoli di Eulero. Quella usata fino ad adesso,

$$x = r \cdot \cos \theta \cos \varphi$$
$$y = r \cdot \sin \theta \cos \varphi$$
$$z = r \cdot \sin \varphi$$

è adatta a problemi in cui l'asse polare più naturale è l'asse z. Ma qui, vista la ubicazione dell'osservatore, l'asse più naturale è l'asse x. Quindi riferiamo gli angoli di Eulero a tale asse, prendendo come polo Nord il punto (1,0,0) e come polo sud il punto (-1,0,0). Allora si ha

$$x = r \cdot \cos \varphi$$
$$y = r \cdot \cos \theta \sin \varphi$$
$$z = r \cdot \sin \theta \sin \varphi$$

Sappiamo che z = 0, ossia, dalla terza identità qui sopra,  $\theta = 0$  o  $\pi$ . Pertanto cos  $\theta = \pm 1$ , e, dalla seconda identita', sin  $\varphi = \pm y/r$ . poiche' y/r varia fra –1 e 1, ed in questo dominio la funzione inversa del seno e' il ramo principale dell'arcoseno, da qui si ottiene  $\varphi = \arcsin(y/r)$ . Se sostituiamo ora al posto di y i valori y<sub>±</sub> ai punti di intersezione, troviamo

$$\varphi = \arcsin(y_{+}/r) = \arcsin(\pm 5.85/r) = \pm 0.62rad$$

Ecco allora la parte di *B* che l'osservatore vede riflessa in *S*, espressa in latitudine  $[\phi]$  e longitudine  $[\theta]$ :

$$-0.62 rad < \varphi < \frac{\pi}{2}$$

Parte 2.



Spostiamo il problema sul piano {z=0} e teniamo in considerazione la parametrizzazione usata nella parte 1. Il segmento dall'origine a P' è il raggio del cerchio, mentre quello dall'origine a P ha lunghezza 2 perché P=(2,0,0). Per il teorema dei seni, i seni degli angoli opposti ai lati di un triangolo sono in proporzione alle lunghezze dei rispettivi lati. L'angolo opposto al lato dall'origine a P vale  $\pi$ - $\gamma$ - $\psi$ , e poiché sin ( $\pi$ - $\gamma$ - $\psi$ ) = sin ( $\gamma$ + $\psi$ ) ora abbiamo sin( $\gamma$  + $\psi$ ) = 2 sin  $\psi$ , ossia

$$\gamma = \arcsin(2\sin\psi) - \psi \tag{1}$$

Inoltre, l'angolo della direzione normale al cerchio al punto P' rispetto alla direzione orizzontale (asse x) vale  $\gamma$ , mentre quello fra il segmento da P a P' e la direzione orizzontale vale  $\psi$ . Quindi il raggio riflesso in P', ossia il segmento da P' a P'', forma con il versore normale un angolo pari a  $\psi$ +  $\gamma$ , e quindi rispetto all'asse orizzontale un angolo pari a  $\psi$ +  $2\gamma$ . Il punto P'' giace sul cerchio di raggio 10: calcoliamo le sue coordinate in funzione di  $\psi$  e  $\gamma$ , e quindi di  $\psi$ , visto che  $\gamma$  è funzione di  $\psi$  grazie a (1). Osserviamo che P' = (cos  $\gamma$ , sin  $\gamma$ ). La retta da P' a P'' ha equazione parametrica

$$r(t) = P' + t \frac{P'' - P'}{\|P'' - P'\|} = (\cos \gamma + t \cos(\psi + 2\gamma), \sin \gamma + t \sin(\psi + 2\gamma))$$
(2)

Ora troviamo il valore di t per il quale r(t) ha norma 10:

$$100 = ||r(t_0)||^2$$
  
=  $\cos^2 \gamma + t_0^2 \cos^2(\psi + 2\gamma) + 2t_0 \cos \gamma \cos(\psi + 2\gamma) + \sin^2 \gamma + t_0^2 \sin^2(\psi + 2\gamma) + 2t_0 \sin \gamma \sin(\psi + 2\gamma)$   
=  $1 + t_0^2 + 2t_0 (\cos \gamma \cos(\psi + 2\gamma) + \sin \gamma \sin(\psi + 2\gamma)) = 1 + t_0^2 + 2t_0 \cos(\psi + 2\gamma - \gamma) = 1 + t_0^2 + 2t_0 \cos(\psi + \gamma)$ 

da cui  $t_0^2 + 2t_0 \cos(\psi + \gamma) - 99 = 0$ . E' chiaro dalla figura che la radice che ci serve è quella positiva, ossia:

$$t_0 = -\cos(\psi + \gamma) + \sqrt{\cos^2(\psi + \gamma) + 99}$$
(3)

Ora segue da (2) che

$$P'' = (\cos \gamma + t_0 \cos(\psi + 2\gamma), \sin \gamma + t_0 \sin(\psi + 2\gamma))$$

Perciò l'angolo  $\phi=\phi(\psi)$  che il segmento dall'origine a P" forma con la direzione orizzontale è

$$\varphi = \varphi(\psi) = \operatorname{arctg} \frac{\sin \gamma + t_0 \sin(\psi + 2\gamma)}{\cos \gamma + t_0 \cos(\psi + 2\gamma)}.$$
(4)

Sostituendo (1) e (3) in questa equazione abbiamo l'espressione di  $\phi(\psi)$  come funzione solo di  $\psi$ .

In (1), quando  $\psi$  è piccolo (e dunque anche  $\gamma$ ) possiamo sviluppare il seno e il coseno fino al secondo ordine ed otteniamo

$$\frac{1}{2}\gamma^2\psi + \left(1 + \frac{1}{2}\psi^2\right)\gamma - \psi = 0$$

da cui, poiché abbiamo  $\gamma$  dello stesso ordine di  $\psi$  e poiché trascuriamo ordini maggiori del secondo, si ottiene  $\gamma \approx \psi$ .

Osserviamo che gli sviluppi di Taylor dei due angoli  $\psi$  e  $\gamma$  sono uguali sino al secondo ordine.

Sia P' il punto di S osservato, *r* la retta che passa per P' con deviazione angolare  $2\gamma + \psi = 3\psi$ , <u>v</u> il vettore direttore di tale retta, ovvero P' – O. Osserviamo che

$$r = \{ \underline{x} \in \mathbb{R}^2 : \underline{x} = \lambda \underline{v} + P' \}$$

$$\underline{v} = \begin{pmatrix} \cos(3\psi) \\ \sin(3\psi) \end{pmatrix} P' = \begin{pmatrix} \cos(\psi) \\ \sin(\psi) \end{pmatrix} \begin{cases} x = \lambda \cos(3\psi) + \cos(\psi) \cong \lambda \left( 1 - \frac{(3\psi)^2}{2} \right) + 1 - \frac{\psi^2}{2} \end{cases}$$

$$y = \lambda \sin(3\psi) + \sin(\psi) \cong 3\lambda \psi + \psi$$

da cui risulta

$$r = \left\{ (x, y) \in \mathbb{R}^2 : y = \psi(3x - 2) \right\}$$

Calcoliamo ora le intersezioni di r con la sfera B

$$P''=r \cap B = \begin{cases} y = \psi(3x-2) \\ x^2 + y^2 = 100 \end{cases} \Leftrightarrow x^2 + \psi^2(3x-2)^2 = 100 \end{cases}$$

da cui

$$x = 10 \left( 1 - \frac{98}{25} \psi^2 \right) \Longrightarrow y = 28\psi$$

Passando alle coordinate sferiche  $\phi$ , $\vartheta$  (ossia gli angoli di Eulero: ma ricordiamo che siamo nel piano  $\vartheta$ =0), si ha

$$\binom{x}{y} = \binom{10\cos\phi_{\psi}}{10\sin\phi_{\psi}} \Leftrightarrow \phi_{\psi} \cong \tan\phi_{\psi} = \frac{y}{x} = \frac{28\psi}{10\left(1 - \frac{98}{25}\psi^2\right)} = \frac{14}{5}\psi$$

quindi l'osservatore in P che guarda la sfera S con deviazione angolare  $\psi$  (piccolo) nel punto P' vede riflesso il punto P'' della sfera B tale che

$$P = \begin{pmatrix} 2\\0\\0 \end{pmatrix} \quad P_{g}' = \begin{pmatrix} \cos(\psi)\\\sin(\psi)\cos(\vartheta)\\\sin(\psi)\sin(\vartheta) \end{pmatrix} \quad P_{g}'' = \begin{pmatrix} 10\cos(\phi_{\psi})\\10\sin(\phi_{\psi})\cos(\vartheta)\\10\sin(\phi_{\psi})\sin(\vartheta) \end{pmatrix}$$

Parte 3.

Per quanto visto, supponendo  $\psi$  piccolo, si ha che il rapporto fra l'apertura angolare  $\phi_\psi$  e  $\gamma$  è  $\frac{14}{5}$ 

Vogliamo ora studiare l'andamento della derivata di  $\varphi$  rispetto a  $\psi$  per  $\psi$  che tende a  $\psi_0 = \pi/6$ , l'angolo di tangenza.

Osserviamo che:

$$\frac{d\gamma}{d\psi} = \frac{2\cos\psi}{\sqrt{1-4\sin^2\psi}} - 1$$

ii. dalla (3) si ha
$$\frac{dt_0}{d\psi} = (\gamma'+1) \left[ \sin(\psi+\gamma) + \frac{\cos(\psi+\gamma)\sin(\psi+\gamma)}{\sqrt{\cos^2(\psi+\gamma)+99}} \right]_{\psi \to \psi_0}^{\infty} (1+\gamma')$$

Nella (4) poniamo

$$\phi(\psi) = \arctan[\alpha(\psi)], \quad \alpha(\psi) = \frac{\sin \gamma + t_0 \sin(\psi + 2\gamma)}{\cos \gamma + t_0 \sin(\psi + 2\gamma)}$$

da cui

$$\frac{d\phi(\psi)}{d\psi} = \frac{1}{1 + \alpha^2(\psi)} \frac{d\alpha(\psi)}{d\psi}$$
(5)

е

$$\frac{d\alpha(\psi)}{d\psi} = \frac{\left(\gamma'\cos\gamma + \left[t_0'\sin(\psi+2\gamma) + t_0\cos(\psi+2\gamma)(1+2\gamma')\right]\right)\left[\cos\gamma + t_0\cos(\psi+2\gamma)\right]}{\left[\cos\gamma + t_0\cos(\psi+2\gamma)\right]^2} + \frac{\left\{\left[\sin\gamma + t_0\sin(\psi+2\gamma)\right]\left(-\gamma'\sin\gamma + t_0'\cos(\psi+2\gamma) - t_0\sin(\psi+2\gamma)(1+2\gamma')\right)\right\}}{\left[\cos\gamma + t_0\cos(\psi+2\gamma)\right]^2}$$

poiché

$$\lim_{\psi \to \psi_0} \gamma = \frac{\pi}{3} \quad ; \quad \lim_{\psi \to \psi_0} t_0 = 3\sqrt{11} \quad ; \quad \lim_{\psi \to \psi_0} \alpha = \alpha_0 = \sqrt{3} \frac{1 + \sqrt{33}}{1 - 3\sqrt{33}}$$

In base alla (ii) ora abbiamo

$$\frac{d\alpha}{d\psi} \underset{\psi \to \psi_0}{\approx} 2 \frac{(1+3\sqrt{33})^2 \left(\frac{1}{2}+\gamma'\right)+3(1+\sqrt{33})^2(1+\gamma')}{(1-3\sqrt{33})^2} = 2\left[\left(\frac{1}{2}+\gamma'\right)+\alpha_0^2(1+\gamma')\right]$$

Perciò, dalla (5)

$$\frac{d\phi(\psi)}{d\psi} \approx_{\psi \to \psi_0} \frac{2}{1 + \alpha_0^2} \left[ (1 + \alpha_0^2) \gamma' + \frac{1}{2} + \alpha_0^2 \right] = 2(1 + \gamma') - \frac{1}{1 + \alpha_0^2} = \frac{4\cos\psi}{\sqrt{1 - 4\sin^2\psi}} - \frac{1}{1 + \alpha_0^2} \xrightarrow{\psi \to \psi_0} + \infty$$

Vogliamo studiare l'ordine di infinito della derivata: pertanto poniamo

$$\psi = \psi_0 - x$$
  
 $\psi \rightarrow \psi_0 \Leftrightarrow x \rightarrow 0$ 
per cui

Facendo uso delle formule di duplicazione del seno e del coseno, possiamo sviluppare secondo Taylor la derivata appena calcolata, ed otteniamo

$$\frac{d\phi(\psi)}{d\psi} \underset{x \to 0}{\approx} \frac{k}{\sqrt{x}} \quad , \quad k = 2 \cdot \sqrt[4]{\frac{3}{4}} \tag{6}$$

Come ci aspettavamo, piccole variazioni di  $\psi$  portano a variazioni molto grandi dell'angolo  $\phi$  quando  $\psi$  si avvicina all'angolo di tangenza  $\psi_0$ . In prossimità dell'angolo di tangenza la mappa di riflessione produce quindi una forte distorsione delle aree, ampliandole.

Esercizio 2. Una scena consiste di un cubo con centro l'origine e lato 2, dalle pareti perfettamente speculari parallele agli assi coordinati, ed una sfera S con

centro l'origine e raggio 10, dipinta con i meridiani e i paralleli, come un mappamondo. L'osservatore si trova nel punto O = (2,2,2)

1) Sia F la faccia del cubo più vicina all'osservatore e parallela al piano xy. Quale valore di meridiano e parallelo (cioè quale latitudine e longitudine) l'osservatore vede riflessi al centro della faccia F? (Suggerimento: utilizzare la parametrizzazione della sfera data dagli angoli di Eulero).

2) Come cambia la risposta precedente se il cubo viene traslato in modo che il centro della faccia F sia l'origine?

Si consideri la faccia più vicina all'osservatore e parallela al piano xz.
 Si consideri il segmento I dato dall'intersezione di questa faccia con il piano yz

Trovare quali punti della sfera l'osservatore vede riflessi in l



Svolgimento.

Parte 1.

O = (2, 2, 2)

Centro della faccia c = (0,0,1)

Il vettore della direzione di visuale è  $V_1 = \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}$ 

Normalizzando:  $V_1 \rightarrow V = \left(\frac{2}{3}, \frac{2}{3}, \frac{1}{3}\right)$ Il versore normale alla faccia è N = (0,0,1).

Il versore del raggio riflesso è  $R_1 = 2(N,V)N - V = \begin{pmatrix} 0\\0\\\frac{2}{3}\\\frac{2}{3} \end{pmatrix} - \begin{pmatrix} \frac{2}{3}\\\frac{2}{3}\\\frac{1}{3}\\\frac{1}{3} \end{pmatrix} = \begin{pmatrix} -\frac{2}{3}\\-\frac{2}{3}\\\frac{1}{3}\\\frac{1}{3} \end{pmatrix}.$ 

L'equazione della sfera è  $x^2 + y^2 + z^2 = 100$ 

L'equazione parametrica della retta uscente da c in direzione R<sub>1</sub> è  $\begin{cases} x = -\frac{2}{3}t \\ y = -\frac{2}{3}t \\ z = 1 + \frac{1}{3}t \end{cases}$ Ricaviamo t sostituendo nell'equazione della sfera S<sub>1</sub>. Si ha  $\left(-\frac{2}{3}t\right)^2 + \left(-\frac{2}{3}t\right)^2 + \left(1 + \frac{1}{3}t\right)^2 = 100,$ da cui t = 9.6

Quindi il punto p<sub>1</sub> di intersezione della retta riflessa con la sfera è  $\begin{cases} x = -\frac{2}{3}(9.6) \\ y = -\frac{2}{3}(9.6) \\ z = 1 + \frac{1}{3}(9.6) \end{cases}$ ossia p<sub>1</sub> =(-5.8, -5.8, 3.2).

Con la parametrizzazione di Eulero possiamo scrivere  $p_1$ =  $(10\cos\theta\cos\varphi,$  $10\sin\theta\cos\varphi$ ,  $10\sin\varphi$ ), e quindi  $\cos\theta\cos\varphi = \frac{-5.7}{10} = -0.57$  $\sin\theta\cos\varphi = \frac{-5.7}{10} = -0.57$  $\sin \varphi = \frac{3.2}{10} = 0.32$  $\cos\varphi = \sqrt{1 - \sin^2\varphi} = 0.9$  $\cos\theta = \frac{\cos\theta\cos\varphi}{\cos\varphi} = -0.6$ da cui:  $\varphi = \arccos(-0.6) = 0.45$  $\theta = \arccos(0.9) = 2.21$ 

#### Parte 2.

Se la faccia utilizzata nel primo punto fosse traslata in (0, 0, 0) allora calcoli analoghi a quelli nella risposta alla parte 1 darebbero:

$$V_{2} = \left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right)$$
$$R_{2} = \left(-\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right)$$

L'equazione parametrica del raggio riflesso è

$$x = -\frac{1}{\sqrt{3}}t$$
$$y = -\frac{1}{\sqrt{3}}t$$
$$z = \frac{1}{\sqrt{3}}t$$

Ricaviamo t dall'equazione  $\left(-\frac{1}{\sqrt{3}}t\right)^2 + \left(-\frac{1}{\sqrt{3}}t\right)^2 + \left(\frac{1}{\sqrt{3}}t\right)^2 = 100$  della sfera S<sub>2</sub>: si ottiene

t = 10.

Quindi il punto di intersezione è p<sub>2</sub> =  $\left(-\frac{10}{\sqrt{3}}, -\frac{10}{\sqrt{3}}, \frac{10}{\sqrt{3}}\right)$ 

Ora nella parametrizzazione di Eulero possiamo scrivere  $p_2 = (10\cos\theta\cos\varphi, 10\sin\theta\cos\varphi, 10\sin\varphi)$ e quindi

$$\cos\theta\cos\varphi = -\frac{1}{\sqrt{3}}$$
$$\sin\theta\cos\varphi = -\frac{1}{\sqrt{3}}$$
$$\sin\varphi = \frac{1}{\sqrt{3}}$$

Allora:

$$\cos\varphi = \sqrt{1 - \sin^2\varphi} = \frac{\sqrt{2}}{\sqrt{3}}$$
$$\cos\varphi = \frac{\cos\theta\cos\varphi}{\cos\varphi} = -\frac{1}{\sqrt{2}}$$

da cui:

$$\varphi = 0.6$$
  
 $\theta = 2.3$ 

#### Parte 3.

Troviamo l'equazione cartesiana ax+by+cz=d del piano che passa per i punti m = (0,1,1), n = (0,1,-1) del segmento I e per il punto O<sub>1</sub> = (2, 2, 2) che si ottiene riflettendo il punto O rispetto al piano zy che passa per il segmento.



$$\begin{cases} -2a+2b+2c = -d \\ b+c = -d \\ b-c = -d \end{cases} \begin{cases} b = -d \\ c = 0 \\ a = -\frac{d}{2} \end{cases} x+2y=2$$

Quindi l'equazione del piano è x + 2y = 2 ossia x = 2(1 - y)La sostituiamo nell'equazione della sfera ed otteniamo  $4(1 - y)^2 + y^2 + z^2 = 100$ Sviluppando il quadrato abbiamo  $5y^2 - 8y + z^2 = 96$ Rammentando che il valore di x soddisfa l'equazione  $x^2 + y^2 + z^2 = 100$  otteniamo dunque  $\begin{cases} z = \sqrt{96 - 5y^2 - 8y} \\ x = 4 + 4y^2 - 8y \end{cases}$ , dove la y varia fra i valori y. e y<sub>+</sub> che

si ottengono riflettendo il raggio di visuale sui punti m = (0,1,1) e n = (0,1,-1) e tracciando questi raggi riflessi fino a colpire la sfera.

Eseguiamo i calcoli di riflessione sul punto m = (0,1,1):

$$N = (0,1,0)$$

$$V = \begin{pmatrix} 2\\2\\2\\2 \end{pmatrix} - \begin{pmatrix} 0\\1\\1 \end{pmatrix} = \begin{pmatrix} 2\\1\\1 \end{pmatrix} \qquad \text{che dopo la normalizzazione diventa } \left(\frac{2}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}\right)$$

$$R_{m} = 2(N,V)N - V = \left(-\frac{2}{\sqrt{6}}, \frac{1}{\sqrt{6}}, -\frac{1}{\sqrt{6}}\right)$$

L'equazione parametrica della retta uscente da m in direzione R<sub>m</sub> è

$$\begin{cases} x = -\frac{2}{\sqrt{6}}t\\ y = 1 + \frac{1}{\sqrt{6}}t\\ z = 1 - \frac{1}{\sqrt{6}}t \end{cases}$$

Calcoliamo l'intersezione di questa retta con la sfera  $x^2 + y^2 + z^2 = 100$ : da  $\left(-\frac{2}{\sqrt{6}}t\right)^2 + \left(1 + \frac{1}{\sqrt{6}}t\right)^2 + \left(1 - \frac{1}{\sqrt{6}}t\right)^2 = 100$ si ricava t=9.8 e per questo t si trova il punto di intersezione y- = (-8, 5, -3)

Ripetiamo riga per riga gli stessi calcoli per la riflessione sul punto n = (0,1,-1):

$$N = (0,1,0)$$

$$V = \begin{pmatrix} 2\\2\\2 \end{pmatrix} - \begin{pmatrix} 0\\1\\-1 \end{pmatrix} = \begin{pmatrix} 2\\1\\3 \end{pmatrix}$$
 che dopo la normalizzazione diventa  

$$\begin{pmatrix} \frac{2}{\sqrt{14}}, \frac{1}{\sqrt{14}}, \frac{3}{\sqrt{14}} \end{pmatrix}$$

$$R_n = 2(N,V)N - V = \left(-\frac{2}{\sqrt{14}}, \frac{1}{\sqrt{14}}, -\frac{3}{\sqrt{14}}\right)$$

L'equazione parametrica della retta uscente dal punto n in direzione  $R_n$  è

$$x = -\frac{2}{\sqrt{14}}t$$
$$y = 1 + \frac{1}{\sqrt{14}}t$$
$$z = -1 - \frac{3}{\sqrt{14}}t$$

Calcoliamo l'intersezione con la sfera  $x^2 + y^2 + z^2 = 100$ . Abbiamo  $\left(-\frac{2}{\sqrt{14}}t\right)^2 + \left(1 + \frac{1}{\sqrt{14}}t\right)^2 + \left(-1 - \frac{3}{\sqrt{14}}t\right)^2 = 100$ ,

da cui t=8.9, che corrisponde al punto di intersezione  $y_{+} = (-4.7, 3.4, -8)$ .

**Esercizio 3.** Una scena si compone di una stanza cubica di lato 3 metri con i muri opachi (coefficiente di riflettività 0 e di Lambert 1, coefficiente di Phong 1), pavimento rosso puro, pareti blu puro, soffitto verde puro, tutti e tre i colori alla massima intensità, e di uno specchio ideale (coefficiente di riflettività 1 e di trasmissione e di Lambert 0, coefficiente di Phong infinito) di forma quadrata, di diametro 1 metro, sospeso ad altezza 1 metro, disposto parallelamente al pavimento con i lati paralleli ai muri e con il centro posto verticalmente sopra il centro del pavimento. Supponiamo che l'origine coincida con uno dei vertici del pavimento, e che l'osservatore stia sul soffitto al punto E = (2, 2, 3). La luce è al punto L = (1.5, 1.5, 2.5). Che colore vede l'osservatore se guarda verso il centro dello specchio ma la riflessione è calcolata con la mappa di riflessione avente per centro il centro del pavimento?

Svolgimento. L'osservatore è in E = (2,2,3). Il centro dello specchio è al punto P = (1.5, 1.5, 1). Rispetto a questo punto, il versore dell'osservatore è proporzionale a E-P = (0.5, 0.5, 2), e più precisamente è il vettore

$$V = \frac{E - P}{\|E - P\|} = \frac{1}{\sqrt{4.5}} (0.5, 0.5, 2)$$

Il versore normale ovviamente è N = (0,0,1). Quindi il versore riflesso è

$$R = 2(N,V)N - V = 2\frac{2}{\sqrt{4.5}}(0,0,1) - \frac{1}{\sqrt{4.5}}(0.5,0.5,2) = \frac{1}{\sqrt{4.5}}(-0.5,-0.5,2)$$

Al centro dello specchio l'osservatore dovrebbe vedere il punto della parete dato dal punto di uscita dalla stanza cubica della retta P+Rt, ma se invece usiamo la mappa di riflessione centrata al centro del pavimento, che è il punto Q = (1.5, 1.5, 0), allora ciò che viene visto è il punto in cui la semiretta

$$Q + Rt = (1.5, 1.5, 0) + \frac{t}{\sqrt{4.5}} (-0.5, -0.5, 2) = \left(1.5 - \frac{0.5}{\sqrt{4.5}}t, 1.5 - \frac{0.5}{\sqrt{4.5}}t, \frac{2}{\sqrt{4.5}}t\right)$$

interseca le pareti della stanza cubica, per t>0. Questa semiretta tocca il piano x=0 per  $t = 3\sqrt{4.5}$ , e tocca per lo stesso valore di t il piano y=0. In altre parole, la semiretta colpisce le pareti laterali in un angolo, proprio sulla retta x=y=0, ossia l'asse z. L'intersezione avviene ad altezza z = 6, quindi fuori della stanza: pertanto prima di colpire le pareti laterali, la semiretta tocca il soffitto. Questo accade quando  $\frac{2t}{\sqrt{4.5}} = 3$ , ossia per  $t = \frac{3}{2}\sqrt{4.5}$ . Il punto del soffitto dove avviene l'intersezione è H=(<sup>3</sup>/<sub>4</sub>, <sup>3</sup>/<sub>4</sub>, 3). Il soffitto è verde puro ed ha coefficiente di Lambert 1. La normale al soffitto è –N=(0,0,-1). La direzione della luce rispetto a H è il versore K proporzionale a L-H = (<sup>3</sup>/<sub>4</sub>, <sup>3</sup>/<sub>4</sub>, -1/2), ossia

$$K = \sqrt{\frac{8}{11}} \left(\frac{3}{4}, \frac{3}{4}, -\frac{1}{2}\right)$$

Quindi il colore che l'osservatore vede, se si usa la mappa di riflessione, è il verde di intensità  $K = \sqrt{\frac{8}{11}} \frac{1}{2} = \sqrt{\frac{8}{11}} \frac{1}{4} \approx 0.4264$ .

*Esercizio* 4. Una scena si compone di un triangolo T rosso puro alla massima intensità, opaco, con vertici in (2,0,0), (2,2,0) e (0,0,2), e della calotta sferica data dalla parte S della superficie sferica  $\Sigma$  di raggio 5 con latitudine compresa fra  $\pi/4$  e  $\pi/2$  (equatore), verde pura alla massima intensità, opaca. Si calcoli la proiezione centrale della scena sulla sfera  $\Theta$  con centro l'origine e raggio 10 e centro di proiezione l'origine, come segue.

- (i) Si determinino le proiezioni su  $\Sigma$  dei tre vertici di T. (Suggerimento: basta rinormalizzare i tre vertici in maniera che abbiano norma uguale al raggio di  $\Sigma$ ).
- (ii) Si determinino i tre lati della proiezione di T. (Suggerimento: basta parametrizzare i lati, ad esempio come segmenti di rette in forma parametrica, e rinormalizzare come prima).
- (*iii*) Si determini la proiezione di S. (Suggerimento: si tratta della calotta sferica in  $\Theta$  dall'equatore al parallelo a nord a  $\frac{\pi}{4}$  radianti).
- (iv) Ora si calcoli la mappa di riflessione richiesta. Si può assumere che il colore di sfondo (ossia quello della sfera  $\Theta$  nelle aree su cui non si proietta nessun poligono della scena) sia nero. (Suggerimento: il triangolo è più vicino all'origine della calotta sferica, quindi la copre laddove le proiezioni si sovrappongono).

*Esercizio 5.* Si aggiunga alla scena una superficie puramente riflettente (specchio ideale) R data dall'emisfero con centro l'origine e raggio 1 nel semispazio  $\{z \ge 0\}$ , un osservatore nel punto (0, 0, 4).

- (i) Si parametrizzi R con gli angoli di Eulero e per ciascun suo punto  $\phi$ ,  $\theta$  si calcoli la direzione del versore  $\mathbf{r}$  ottenuto riflettendo rispetto alla normale il versore  $\mathbf{v}$  che individua la posizione dell'osservatore. (Suggerimento: il versore normale al punto di angoli di Eulero  $\phi$ ,  $\theta$  è il versore radiale (sin  $\phi$ , cos  $\phi$  sin  $\theta$ , cos  $\phi$  cos  $\theta$ )).
- (ii) Per ciascun tale punto si determini tramite mappa di riflessione cosa vede l'osservatore quando guarda quel punto, supponendo che ci sia solo una luce ambiente bianca di intensità 1, e che tutti i coefficienti di riflessione della luce ambiente valgano 1. (Suggerimento: i poligoni colorati dalla sola luce ambientale hanno colore costante; si usi la direzione del versore riflesso r come puntatore alla mappa di riflessione

precedentemente calcolata, che va espressa in termini degli angoli di Eulero).

*Esercizio 6.* Ora supponiamo che T abbia coefficienti di Lambert e di riflessione speculare entrambi uguali a  $\frac{1}{2}$  e che quelli di S siano entrambi uguali a  $\frac{1}{3}$ . Entrambi gli esponenti di Phong valgono 1. Lo specchio ideale Q ovviamente ha coefficiente di Lambert 0, di riflessione speculare 1 e di Phong infinito, e colore bianco. Tutti i coefficienti di riflessione ambientale sono 0. Lo sfondo è nero. C'è una sorgente di luce puntiforme che coincide con la posizione dell'osservatore. Lo sfondo è la sfera  $\Sigma$  dipinta di nero. Si calcoli il colore che vede l'osservatore quando guarda in direzione  $\left(\frac{1}{20}, \frac{1}{20}, -1\right)$ , utilizzando il Ray Tracing ricorsivo con una sola generazione di raggi. (Sugqerimento: si determini il piano dove qiace il triangolo T ed il punto  $\mathbf{p}$  dove il raggio proiettore lo interseca. Si provi che questo punto è interno a T (ad esempio proiettando su uno dei piani coordinati per ridurre il problema a dimensione due, oppure esprimendo p come combinazione convessa dei vertici di T). Si calcoli il contributo di p all'illuminazione grazie all'equazione di Phong (ma attenzione all'angolo fra la direzione di provenienza del raggio proiettore e quella del versore riflesso della luce: se è maggiore di 90 gradi il termine di Phong è nullo, e rimane solo quello di Lambert). Si trovi poi il versore riflesso  $\mathbf{r}$  del versore da  $\mathbf{p}$  alla posizione dell'osservatore  $\mathbf{v}$ , e si tracci il raggio riflesso, calcolando le intersezioni con il piano di base e con il piano orizzontale su cui giace il bordo superiore della calotta sferica. Si noti che l'intersezione con il piano di base è esterna a  $\Sigma$  mentre quella col piano superiore è interna - quindi il raggio proiettore deve intersecare la calotta S. Si calcoli il punto di intersezione ed il suo contributo al Ray Tracing ricorsivo).

#### Esercizio 7.

1. Una scena consiste di una stanza cilindrica avente per base un disco intorno all'origine di raggio 1 ed altezza 2; il pavimento è un cono retto che intrude nella stanza (la quale quindi non è convessa), di altezza pari al raggio. Il cono è uno specchio ideale, mentre tutti gli altri muri sono opachi. Consideriamo dentro la stanza punti  $p = (x_p, y_p, z_p)$  con  $x_p e y_p \neq 0$ . L'asse z è quello ortogonale al soffitto. Un osservatore è situato al centro del soffitto (rammentiamo che il soffitto è ad altezza 2). Quali sono le coordinate dei punti p interni alla stanza che l'osservatore può vedere riflessi nel cono? (Suggerimento: trasformare il problema in un problema equivalente bidimensionale. Per risolverlo, cominciare a determinare quali punti della stanza l'osservatore vede riflessi quando guarda:

- quasi verso il vertice del cono, ossia il più in alto possibile (vicino al vertice ma non proprio al vertice perché lì la normale cambia e si un punto con direzione riflessa eccezionale, ma un solo punto è irrilevante per il rendering )
- il più in basso possibile, ossia punti del cono vicino alla sua base (di nuovo, vicino alla base ma non proprio alla base, perché sui punti della base la normale cambia, ma si tratta di una curva eccezionale e non una superficie, quindi irrilevante per il rendering)

Poi ci si dovrebbe chiedere se l'angolo di riflessione varia in maniera monotòna quando l'osservatore fa variare il punto osservato fra questi due estremi, e dedurne quali punti vede.

2. Per ciascun tale punto p, verso quale punto del cono l'osservatore deve guardare per vederne il riflesso? (Suggerimento: parametrizzare il cono e scrivere per ciascun suo punto q il vettore normale ed il raggio riflesso del raggio che unisce l'osservatore a q; tracciare questo raggio riflesso e vedere per quale scelta di q esso passa per p. È molto conveniente trasformare il problema in un problema equivalente bidimensionale.)

# **RENDERING DELLE OMBRE**



E' un algoritmo a precisione d'oggetto a due passi per il rendering delle ombre. Le ombre vengono determinate un'unica volta, prima ancora della determinazione delle superfici visibili, dopo di che il calcolo rimane valido anche se l'osservatore cambia posizione (le ombre dipendono solo dalla posizione delle sorgenti di luce, non dell'osservatore: se le sorgenti di luce non cambiano in numero o in posizione le ombre rimangono le stesse). Per semplicità supponiamo che ci sia solo una sorgente di luce: se ce ne sono più di una si itera il procedimento per ciascuna di esse.

Il primo passo consiste nel calcolare la posizione delle ombre, applicando l'algoritmo di visibilità di <u>Atherton-Weiler</u> per determinare quali superficie non sono visibili alla sorgente di luce, e quindi in ombra.

Per far questo è necessario definire un volume di visuale (view volume) centrato rispetto alla sorgente di luce.

Nel caso in cui un unico volume di visuale non riuscisse a racchiudere tutti gli oggetti (si pensi ad esempio al caso di una luce che sia al centro di una stanza) è necessario costruire una famiglia di volumi di visuale, nel quale ognuno sia adiacente e disgiunto dall'altro, fino alla totale copertura di tutti gli oggetti nell' ambiente. Questa tecnica viene chiamata sezionamento, essa consiste nel ritagliare dai poligoni originali della scena nuovi poligoni che consistono dei loro frammenti illuminati dalla sorgente di luce considerata.

Questo primo passo termina con la produzione di una lista di poligoni illuminati, ognuno dei quali è accompagnato da un puntatore che riporta al poligono padre, permettendo quindi la fusione del poligono di illuminazione con il poligono originario sotto forma di poligono di colorazione di superficie, o meglio di illuminazione o ombreggiatura (e anche da un puntatore alla sorgente luminosa, per calcolare l'illuminazione relativa a quella specifica sorgente: per

ogni diversa sorgente si aggiunge una nuova lista di frammenti di poligoni, relativa al *view volume* centrato in tale sorgente ed ai puntatori ad essa).



Il secondo passo consiste nella determinazione delle superficie visibili dalla posizione dell'osservatore, effettuata sulla lista dei poligoni precedentemente modificata al primo passo (quindi inclusi i nuovi poligoni consistenti nei frammenti illuminati dei poligoni della scena, sorgente per sorgente).



Una volta determinato che un poligono è visibile attraverso un dato pixel, l'algoritmo di conversione proietta il poligono su quel pixel e verifica se su quel pixel cade o meno un poligono di illuminazione. Nel primo caso il pixel è illuminato dalla sorgente luminosa, mentre nel secondo caso è in ombra. Ecco il risultato:



## METODO DEL DOPPIO Z-BUFFER DI WILLIAMS (MAPPA D'OMBRA)

Il metodo della mappa d'ombra (shadow map, o z-buffer delle ombre) è un procedimento a precisione di immagine per il rendering delle ombre, introdotto in [L. Williams, Casting Curved Shadows on Curved Surface, SIGGRAPH 78, in Computer Graphics 12, August 1978, pagg. 270-274]. Il procedimento si basa sul confronto fra due z-buffer, quello consueto centrato sull'osservatore ed un altro centrato sulla sorgente di luce (se ci sono più sorgenti si usano ulteriori z-buffer). Anzitutto si calcola lo z-buffer dal punto di vista della sorgente di luce, che serve a determinare quali punti della scena sono direttamente visibili dalla sorgente, e quindi in luce. Poi si calcola lo z-buffer dal punto di vista dell'osservatore, ma nella variante seguente. Per ogni pixel, dopo aver determinato quale poligono della scena è visibile dall'osservatore, si trovano le coordinate tridimensionali del punto osservato (nel sistema di riferimento dell'osservatore: chiamiamole ( $x_0, y_0$ ,  $z_{0}$ ). Queste coordinate vengono trasformate grazie alla trasformazione prospettica che sposta l'osservatore nella sorgente. Scriviamo le coordinate così trasformate (x<sub>o</sub>', y<sub>o</sub>', z<sub>o</sub>'): esse sono ora coordinate nel sistema di riferimento centrato sulla sorgente di luce. Pertanto le prime due componenti identificano il pixel dello z-buffer centrato sulla luce attraverso cui passa il raggio proiettore che passa più vicino al punto della scena osservato: quindi permettono di recuperare il valore dello z-buffer della sorgente a quel pixel (chiamiamolo z<sub>i</sub>). La terza coordinata, z<sub>o</sub>', misura la profondità (e quindi la distanza) del punto osservato rispetto alla posizione della sorgente. Invece z<sub>I</sub> misura la distanza dalla sorgente del punto della scena ad essa più vicino lungo il raggio proiettore che attraversa quel pixel. Perciò se  $z_1 > z_0$ ' (ossia il punto illuminato è più prominente verso la sorgente rispetto al punto guardato dall'osservatore) qualche altro poligono si frappone fra la sorgente ed il punto osservato dall'osservatore: quindi tale punto è in ombra, ed allora nello z-buffer dell'osservatore si memorizza non il colore pieno del poligono illuminato, ma quello più scuro del poligono in ombra, che si calcola, ovviamente, a partire dall'equazione dell'illuminazione (se c'è una sola sorgente il valore memorizzato a questo pixel è quello che deriva dalla sola illuminazione ambientale).

In altre parole, lo z-buffer centrato sulla sorgente è uno *z*-buffer delle ombre. Si osservi che il punto osservato non giace di solito esattamente sul raggio proiettore che passa per il centro del pixel della luce, quindi il valore  $z_i$  si ricava tramite interpolazione a partire dai valori circostanti dello z-buffer della luce: se ne fa una media pesata in cui i pesi sono le combinazioni convesse che esprimono ( $x_o'$ ,  $y_o'$ ) in termini delle coordinate di questo buffer. In tal modo si attua una approssimazione, che è accurata purché non ci siano troppi punti con coordinate vicine o uguali a ( $x_o'$ ,  $y_o'$ ). Infatti, tutti questi punti risulterebbero con lo stesso valore di  $z_i$ , anche quando le profondità effettive  $z_o'$  sono molto diverse: in tal caso un oggetto interposto potrebbe coprire alcuni di questi punti ma non altri, e per assicurare una risoluzione adeguata sarebbe necessario un sovracampionamento.

Questo procedimento, come tutti quelli basati su z-buffer, è a precisione di immagine. Esso risente pesantemente degli inerenti arrotondamenti numerici ad esso inerenti: infatti si basa in maniera cruciale sul confronto fra i due valori  $z_1$  e  $z_0$ ', entrambi soggetti ad errori di arrotondamento, ed il secondo in misura ancora maggiore perché ottenuto attraverso una trasformazione matriciale che richiede calcoli aritmetici. Se valori di  $z_1$  e  $z_0$ ' sono vicini ed in conseguenza di tali arrotondamenti si scavalcano, allora un punto che avrebbe dovuto apparire illuminato appare in ombra o viceversa, generando una aberrazione evidente. Per ridurre questo effetto di aliasing si deve, come sempre, effettuare filtraggio, ed in questo caso si puo' usare anche dithering.

Invece non è possibile in questo metodo applicare un supercampionamento a riesaminiamo con maggiore attenzione variabile. Per vederlo. passo l'osservazione che abbiamo fatto poco fa circa la necessità di un sovracampionamento dove la risoluzione è insufficiente. Il metodo si basa su una trasformazione tridimensionale di coordinate che sposta il centro di prospettiva dalla posizione dell'osservatore a quella della sorgente di luce. I punti dello spazio da trasformare sono i punti della scena visti dall'osservatore tracciando raggi proiettori attraverso il centro dei pixel del viewport. Si tratta di un fascio di raggi uscenti dalla posizione dell'osservatore ed angolarmente equispaziati. Pero' i punti della scena che essi intersecano potrebbero non essere equidistribuiti nello spazio, ed ancora meno probabile è che lo siano dopo la proiezione sul rettangolo di pixel disposto di fronte alla posizione della luce che dobbiamo usare per costruire lo z-buffer della luce (questo ad esempio accade se poligoni diversi della scena che sottendono la stessa area quando proiettati sul viewport dell'osservatore coprono due aree molto diverse fra loro quando proiettati sul viewport della luce). In tali casi alcuni punti dopo la trasformazione risultano fittamente distribuiti sul viewport della luce ed altri risultano molto sparsi. Molti punti diversi possono cadere nello stesso pixel del viewport della luce, ed in altri pixel possono non caderne nessuno. Il campionamento fisso dato dai pixel del viewport della luce è allora sufficiente per calcolare uno z-buffer preciso nelle aree dove i punti da trasformare sono distribuiti in modo sparso, ma può essere molto insufficiente per una adeguata precisione nelle aree con punti da trasformare sono fittamente distribuiti. Una versione a passo variabile del campionamento richiede di infittire i raggi proiettori nelle aree in cui le variazioni di intensità di colore sono elevate, ma qui i punti da trasformare sono determinati dalle intersezioni con la scena dei proiettori attraverso i centri dei pixel di un altro viewport, quello dell'osservatore: quindi l'unico modo di infittire i proiettori in modo variabile è di tracciare raggi più fitti attraverso i pixel di questo viewport, nella fase iniziale prima della trasformazione: ma questo deve avvenire laddove i punti di intersezione sono più fitti quando visti dalla posizione della luce, ovvero dopo la trasformazione. Quali siano le zone della scena in cui dovrebbero arrivare più raggi proiettori si può solo sapere dopo la trasformazione, e si può solo determinare tracciando effettivamente raggi nello spazio per determinare le posizioni tridimensionali dei punti di intersezione, poi trasformare questi punti ed infine ritracciare i raggi a ritroso, un procedimento assai oneroso dal punto di vista numerico.

Come sempre con lo z-buffer, ogni poligono deve essere processato dal metodo, anche se dopo viene ricoperto da un altro. Pero' in guesto caso guesta ridondanza produce un dispendio di tempo di calcolo molto maggiore che con il solo z-buffer, perché per ogni pixel, oltre che la scansione necessaria per lo zbuffer, ora si deve anche effettuare una trasformazione di coordinate. Per ridurre la mole di calcolo Williams ha proposto la variante seguente: calcolare prima lo z-buffer consueto dal punto di vista dell'osservatore, poi per ciascun pixel trasformare le coordinate del punto della scena che lo z-buffer individua come visibile al'osservatore e confrontare i valori di z come prima. Se  $z_l < z_o$ ' allora al colore del pixel determinato dallo z-buffer dell'osservatore viene aggiunto un contributo di colore scuro di ombra (viene "dipinto d'ombra"). Questa variante del procedimento è molto più veloce, perché per ogni pixel dello z-buffer dell'osservatore viene trasformato solo il punto della scena visibile attraverso il pixel, e non tutti i punti intersecati dal raggio proiettore (uno per ciascun poligono man mano che si svolge la scansione); pero' è meno precisa, perché il primo passaggio di z-buffer determina l'ombreggiatura senza le ombre, e guindi determina anche gli incrementi di intensità di illuminazione dovuti ai contributi della riflessione speculare: laddove queste zone riflettenti chiare vengono dipinte d'ombra, la loro brillantezza viene attenuata, ma non del tutto annullata, come invece dovrebbe accadere dal momento che, essendo in ombra, non riflettono la luce. Pero', per scene che consistono di superficie prevalentemente opache e poco riflettenti, questa variante è vantaggiosa.

### Esercizi sul metodo del doppio z-buffer di Williams

**Esercizio 1.** Una scena si compone di un triangolo equilatero E con vertici in (1,0,0), (0,1,0) e (0,0,1) ed un triangolo T con vertici in (1,0,0), (0,1,0) e  $(1,0, \frac{1}{2})$ . L'osservatore è disposto lungo l'asse z positivo, all'infinito. C'è una sola sorgente di luce, all'infinito lungo l'asse x positivo: quindi una luce direzionale che emette un fascio di raggi paralleli all'asse x. Lo z-buffer dell'osservatore si basa su un viewport dato da una griglia di pixel quadrati di lato  $\frac{1}{4}$  sul piano z=1, con il pixel centrale che ha centro sull'asse z. Lo z-buffer della luce si basa su un viewport dato da una griglia di quadratini di lato  $\frac{1}{4}$  disposti sul piano x=1, con quello centrale che ha centro sull'asse x.

1. Calcolare lo z-buffer dell'osservatore nel suo pixel centrale (quale triangolo ivi proiettato sta più avanti verso l'osservatore e che profondità spaziale z₀ esso
ha su quel pixel). Quali sono le tre coordinate spaziali  $(x_o, y_o, z_o)$  del punto P osservato?

- Trasformare le coordinate in maniera che l'origine diventi la posizione L della sorgente, e scrivere le coordinate dei vertici dei due triangoli in queste nuove coordinate.
- 3. Calcolare le nuove coordinate  $(x_o', y_o', z_o')$  del punto P dopo la trasformazione.
- Determinare quale è il pixel (ossia quali sono le sue coordinate centrali (x<sub>o</sub>', y<sub>o</sub>')) che corrisponde al punto P sullo z-buffer della luce.
- 5. Trovare il valore della profondità z dello z-buffer della luce al pixel di coordinate centrali (x<sub>o</sub>', y<sub>o</sub>').
- 6. Da questi dati, decidere se il punto P è in luce o in ombra.

Si verifichi che le risposte non cambiano se l'osservatore è situato a distanza finita, ad esempio in O=(0, 0, 2).

#### Soluzione.

- L'osservatore guarda perpendicolarmente al piano z=1, quindi direttamente verso il basso. La proiezione sul suo viewport del triangolo T, che è disposto verticalmente, è il segmento da (1,0) a (0,1), ottenuto semplicemente scartando la coordinata z (proiezione ortogonale sull'asse z, dal momento che l'osservatore sta all'infinito); questa proiezione non passa per il pixel centrale in (0,0). Invece la proiezione del triangolo obliquo E, ottenuta nello stesso modo, è il triangolo di vertici (x, y)= (1, 0), (0, 1) e (0, 0) sul piano z=1: questa proiezione copre il pixel centrale ubicato alle coordinate (x, y) = (0, 0). Il corrispondente punto tridimensionale P è il vertice (0, 0, 1) di E, e quindi il valore di profondità spaziale z<sub>o</sub> vale 1. Si osservi che, se l'osservatore stesse a distanza finita, ad esempio in O=(0, 0, 2), allora si dovrebbe eseguire una proiezione centrale su questo punto invece che una proiezione ortogonale, e quindi la proiezione del triangolo T diventerebbe un triangolo invece che un segmento, ma chiaramente non coprirebbe il punto (0,0): quindi le risposte non cambierebbero.
- 2. La luce è disposta lungo l'asse x: quindi nel suo sistema di coordinate dobbiamo portare z in x, ad esempio scambiando z e x, ma per avere una terna destrorsa occorre anche riflettere y in -y. Questo porta alla trasformazione

$$x \rightarrow z$$
,  $z \rightarrow x$ ,  $y \rightarrow -y$ .

(*Nota*: Se la luce fosse ad un punto L a distanza finita sarebbe inoltre elegante (ma non indispensabile) effettuare una traslazione per spostare l'origine al punto L; ad esempio, se fosse L = (2, 0, 0), che giace sul vecchio asse x, dopo la trasformazione L si troverebbe sul nuovo asse z in posizione 2. Quindi la trasformazione diventerebbe:

x' = z, y' = -y, z' = x-2.

Ma in questo esercizio, invece, L si trova a distanza infinita e quindi non eseguiamo la traslazione.)

I vertici di E ora diventano (x', y', z') = (0, 0, 1), (0, -1, 0) e (1, 0, 0); quelli di T sono (x', y', z') = (0, 0, 1), (0, -1, 0) e  $(\frac{1}{2}, 0, 1)$ .

- 3. Poiché le coordinate originali di P sono  $(x_o, y_o, z_o) = (0, 0, 1)$ , le sue coordinate trasformate sono  $(x_o', y_o', z_o') = (1, 0, 0)$ . Quindi:
- 4. Questo punto si proietta sul pixel dello z-buffer della luce di coordinate centrali  $(x_0', y_0') = (1, 0)$ .
- 5. Ovviamente, per il precedente punto (4), sul pixel di coordinate centrali (x<sub>o</sub>', y<sub>o</sub>') = (1, 0) la proiezione di E proviene dal punto P, ossia (x<sub>o</sub>', y<sub>o</sub>', z<sub>o</sub>') = (1, 0, 0): questo fornisce un valore di profondità z' = 0. Nelle nuove coordinate, invece, il triangolo verticale T, che per il punto (2) ha vertici (x', y', z') = (0, 0, 1), (0, -1, 0) e (½, 0, 1), ha proiezione sul viewport della luce (x = 1, ovvero z' = 1) data dal triangolo proiettato i cui vertici si ottengono, come prima, scartando la terza coordinata z', e quindi sono (x', y') = (0, 0), (0, -1) e (½, 0). Si vede facilmente che il punto (x<sub>o</sub>', y<sub>o</sub>') = (1, 0) è esterno a tale triangolo (infatti il massimo valore di x' per i tre vertici vale ½, e quindi per convessità questo è anche il massimo valore di x' nell'intero triangolo proiettato T, che pertanto non raggiunge mai il valore x' = ½). Pertanto nello z-buffer della luce, in corrispondenza del pixel (x', y') = (0, 0) individuato dalla posizione di P, il valore z' è proprio il valore della profondità di P, ossia z' = z<sub>o</sub>' = 0.
- Il valore z' nello z-buffer della luce al pixel (x', y') corrispondente al punto osservato P è proprio il valore di profondità z<sub>o</sub>' di P. Quindi niente si frappone fra la sorgente di luce ed il punto P, e pertanto P è illuminato.

**Esercizio 2.** Nella stessa scena del problema precedente, adesso la sorgente di luce si trova sull'asse x a distanza finita, diciamo in L=(2,0,0). Come cambiano le risposte?

#### Soluzione.

Ora è piu' elegante effettuare la trasformazione di coordinate comprensiva della traslazione spiegata nella risposta al punto (2) dell'esercizio precedente (ma non è indispensabile): usiamo queste nuove coordinate. La proiezione sul viewport dello z-buffer della luce ora diventa una proiezione centrale, con centro in (x=2, y=0, z=0). Si vede subito che la proiezione centrale del punto P = (1, 0, 0) su questo viewport è il punto Q = (1, 0, ½) per similitudine dei triangoli, dal momento che il piano del viewport è proprio a metà fra P e L. La sua profondità nelle coordinate trasformate è z' = -2. Quindi il pixel del viewport della luce che occorre considerare è quello che contiene Q. Ma il punto Q, che giace sul viewport della luce, è anche il vertice alto del triangolo T: esso ha coordinate trasformate (x', y', z') = ( $\frac{1}{2}$ , 0, -1) e quindi profondità z' = -1, come ovvio perche questa è la

profondità del piano del viewport stesso. Quindi il valore di z-buffer z' = -1 invece che z' = -2 di P: pertanto P è in ombra, ma appena appena: sarebbe bastato spostare L indietro di una quantità arbitrariamente piccola e P sarebbe stato in luce.

**Esercizio 2.** Nella stessa scena del problema precedente, adesso la sorgente di luce si trova a distanza 4 dall'origine lungo la bisettrice dell'ottante positivo. Poniamo il viewport della luce a distanza 2 dall'origine, con asse x parallelo al piano di base. Rispondere alle stesse domande.

Suggerimento. Nel calcolo della trasformazione, ora occorre portare l'asse z nella direzione della suddetta bisettrice: quindi la trasformazione manda (0, 0, 1)

in  $\frac{1}{\sqrt{3}}$  (1,1,1). Gli assi x e y vanno in direzioni trasversali a tale bisettrice: l'asse x è

parallelo al piano di base, e quindi (1, 0, 0) va nel versore di terza coordinata 0 e prime due coordinate (ossia proiezione sul piano di base) perpendicolari alla proiezione della bisettrice, ossia a (1, 1): diciamo che (1, 0, 0) va nel versore

 $\frac{1}{\sqrt{2}}$ (1,-1,0). Il terzo vettore della base canonica iniziale va nel prodotto vettore di

questi due: è ovvio che la sua proiezione sul piano di base deve essere perpendiclare a quella del versore precedente, e quindi diretta come quella della bisettrice, ossia multipla di (1, 1), mentre la terza coordinata deve assicurare un angolo di  $90^{\circ}$  con la suddetta bisettrice, quindi il versore è diretto lungo la bisettrice dell'ottante con x e y negativi e z positivo. Pertanto esso è multiplo di

(-1, -1, 1), ossia vale  $\pm \frac{1}{\sqrt{6}}$ (1,1,-2) (cautela: il verso dovrebbe essere scelto in

modo che la terna sia destrorsa!). Alternativamente, tutte queste considerazioni geometriche si possono sviluppare in termini trigonometrici mediante gli angoli di Eulero, ossia esprimendo le coordinate come

 $x = r \cos\theta \cos\phi$ ,  $y = r \sin\theta \cos\phi$ ,  $z = r \sin\phi$ .

A questo punto non è difficile scrivere la trasformazione di coordinate (si rammenti che la posizione della luce è sì lungo la bisettrice, ma a distanza 4 dall'origine, ossia nel punto  $\frac{4}{\sqrt{3}}$  (1,1,1): non si dimentichi, nella trasformazione di coordinate, la necessaria traslazione che sposta l'origine su questo punto!) Il resto dell'esercizio si sviluppa come nell'esercizio precedente.

**Esercizio 4.** Nella stessa scena dei problemi precedenti, adesso la sorgente di luce si trova a distanza infinita dall'origine lungo l'asse x positivo, ma l'osservatore è disposto lungo la bisettrice del primo ottante, all'infinito. Lo z-buffer dell'osservatore si basa su un viewport dato da una griglia di pixel quadrati di lato ¼ sul piano perpendicolare alla bisettrice a distanza 1 dall'origine, con il pixel centrale che ha centro sulla bisettrice. Come cambiano le risposte circa il fatto che il punto che l'osservatore vede al centro del suo pixel centrale sia in luce od in ombra?

#### Esercizio 5.

La scena consiste di un cubo con centro l'origine e lato 2, dalle pareti perfettamente speculari, ed una sfera S con centro l'origine e raggio 10, dipinta con i meridiani ed i paralleli, come un mappamondo. L'osservatore si trova al punto (2,2,2), la sorgente di luce in (8,0,0). Si determini se l'osservatore vede il centro della faccia x=1 in luce o in ombra, mediante il meccanismo di trasformazione e confronto dell'algoritmo di Williams, ma (per evitare di calcolare a mano i due z-buffer per ogni pixel) qui modificato usando il tracciamento di raggi invece che lo z-buffer (ovviamente seguito dalla appropriata trasformazione di coordinate e dal confronto).

### Esercizio 6.

Una scena consiste di un pavimento infinito sul piano z=0 con una sfera di raggio 1 appoggiata all'origine, ed un disco D sul piano z=3 di raggio 1 e centro in (1,1,3). L'osservatore si trova al punto (0,0,6), la sorgente di luce al punto (5, 5, 5). Come procede il metodo di Williams a determinare se il punto che l'osservatore vede guardando in basso (ossia nella direzione del versore dell'asse z diretto verso la parte negativa di tale asse) è in luce o in ombra? Svolgere i calcoli necessari.

*Svolgimento.* L'osservatore si trova sull'asse z, al punto di altezza 6. Consideriamo lo z-buffer dell'osservatore: ad esempio possiamo immaginare che esso sia costruito su un quadrato ad esempio nel piano z=5, centrato in (0,0,5). L'osservatore guarda verso il basso, quindi nella direzione z negativa, ossia verso l'origine, e pertanto guarda verso il pixel centrale di questo z-buffer. Calcoliamo il contenuto del buffer. Per similitudine, si vede che la proiezione centrale sul piano z=5 del disco D, con punto di fuga nella posizione dell'osservatore (0,0,6) è il disco di centro (1/3, 1/3, 1) e raggio 1/3. Questo disco non contiene l'origine, perché ha raggio 1/3 ma il suo centro dista dall'origine  $\sqrt{2}/3 > 1/3$ . Invece la proiezione centrale della sfera (il cui disco equatoriale orizzontale è z=1) sul piano del buffer z=5 è contenuta nella proiezione del suo disco equatoriale, ossia del disco di centro (0,0) e raggio 1, che è il disco orizzontale ad altezza z=5 di centro (0,0): questo disco ovviamente contiene l'origine, nella direzione del suo disco ovviamente contiene l'origine, centro (0,0): questo disco ovviamente contiene l'origine, centro (0,0): questo disco ovviamente contiene l'origine, centro (0,0): questo disco ovviamente contiene l'origine, nella direzione di centro (0,0): questo disco ovviamente contiene l'origine, nella direzione del suo disco ovviamente contiene l'origine, centro (0,0): questo disco ovviamente contiene l'origine, nella direzione del suo disco di centro (0,0): questo disco ovviamente contiene l'origine, nella direzione del suo disco di centro (0,0): questo disco ovviamente contiene l'origine, nella direzione del suo disco di centro (0,0): questo disco ovviamente contiene d'origine, nella direzione del suo disco di centro disco di centro disco di centro disco disco ovviamente contiene d'origine, nella direzio

di osservazione vede la sfera, la quale in quel punto (il suo polo nord) ha altezza (ovvero profondità z) uguale a 2. Il punto osservato, quindi, è (0,0,2).

Ora cambiamo coordinate per centrarle sulla sorgente di luce, al punto (5,5,5). Basiamo lo z-buffer della luce su un quadrato perpendicolare al segmento da (0,0,2) a (5,5,5): allora il punto osservato si proietta sul centro dello z-buffer della luce, ossia (4,4,4). Si tratta di verificare se tale segmento interseca il disco D. Ma D giace sul piano z=3. Il segmento sta sulla retta

$$\mathbf{r}(t) = (0,0,2) + t(5,5,3) = (5t, 5t, 2+3t).$$

Questa retta interseca il piano z=3 per t=2/3, e quindi nel punto  $\mathbf{q} = (x,y,z) = (10/3, 10/3, 3)$ . Il disco D ha centro nel punto x=1, y=1 e raggio 1. Quindi, affinché  $\mathbf{q}$  sia dentro D dovrebbe valere  $1 \ge (10/3 - 1)^2 + (10/3 - 1)^2 = 2(7/3)^2$ . Ovviamente ciò è falso, e quindi il disco non si frappone frs la posizione della luce e quella del punto osservato: perciò tale punto è in luce.

**Nota**: abbiamo usato uno z-buffer perpendicolare al segmento dal punto osservato alla sorgente di luce. Questo rende facile la soluzione ma non è molto corretto: qui stiamo solo interessandoci all'unico punto osservato, ma in un caso di rendering vero i punti osservati sono tanto e lo z-buffer della luce uno solo, non lo possiamo spostare quando osserviamo un altro punto, e quindi la maggior parte dei punti osservati cade fuori dal centro dello z-buffer della luce. In effetti, abbiamo risolto il problema con un Ray Tracing a partire dalla posizione della luce invece che con uno z-buffer.

Recependo l'osservazione in questa nota, ripetiamo il calcolo basandolo solo su z-buffer. Costruiamo lo z-buffer della luce su un guadrato perpendicolare al segmento, diciamo, dall'origine a (5,5,5), ossia alla bisettrice del primo ottante: ad esempio il piano ortogonale a tale bisettrice che la interseca in (4,4,4) (si tratta del piano  $\pi$  di equazione x+y+z=4). Eseguiamo la proiezione centrale di disco e sfera sul piano dello z-buffer della luce, con punto di fuga nella posizione I = (5,5,5) della luce. A questo scopo, osserviamo che la proiezione ortogonale di un punto  $\mathbf{r} = (x_0, y_0, z_0)$  sul piano x+y+z=4 si ottiene componendo  $\mathbf{r} = \mathbf{r_d} + \mathbf{r_n}$ , dove  $r_d$  è la componente lungo la diagonale e  $r_n$  è quella trasversale. Calcoliamo  $r_d$ . Sia  $d_0 = x_0 + y_0 + z_0$ . Il punto  $\mathbf{r} = (x_0, y_0, z_0)$  giace sul piano parallelo a  $\pi$  di equazione x+y+z=  $d_0$ , quindi la proiezione sulla bisettrice è  $\mathbf{r}_d = (d_0, d_0, d_0)/3$ . Pertanto  $\mathbf{r}_n = (x_0 - d_0/3, y_0 - d_0/3, z_0 - d_0/3)$ , un punto del piano  $\pi_0$  parallelo a  $\pi$  che passa per l'origine, e la proiezione centrale di  $\mathbf{r} = \mathbf{r}_d + \mathbf{r}_n$  sul piano  $\pi$  è  $\mathbf{r} = 4/5$  I + rn /(5- d<sub>0</sub>)) (di nuovo per similitudine, perché il piano di proiezione si trova a distanza  $4\sqrt{3}$  dall'origine, mentre il punto  $\mathbf{r}_{d}$  è a distanza  $d_{0}\sqrt{3}$  ed il punto di fuga I è a distanza  $5\sqrt{3}$  quindi la compressione di scala è di un fattore  $(5-4)/(5-d_0) =$  $1/(5- d_0)).$ 

Quindi la componente trasversale  $\mathbf{r}_n$  si proietta sul piano  $\pi$  sul punto 4/5  $\mathbf{I} + \mathbf{r}_n/(5-d_0)$ : per vedere il punto  $\mathbf{r} = (x_0, y_0, z_0)$  dalla ubicazione della sorgente di luce bisogna guardare nella direzione data dal pixel in posizione 4/5  $\mathbf{I} + \mathbf{r}_n/(5-d_0)$ 

nello z-buffer della luce. Ora concretizziamo la nostra attenzione al punto  $\mathbf{r} = (0,0,2)$  che l'osservatore sta osservando: in tal caso d<sub>0</sub>=2,  $\mathbf{r}_n = (-2/3, -2/3, 4/3)$ , e 4/5  $\mathbf{l} + \mathbf{r}_n/(5-d_0)$  dista dal centro dello z-buffer della luce (che è il punto 4/5  $\mathbf{l} =$ 

(4,4,4)) una distanza pari a 
$$\frac{\|r_n\|}{5-d_0} = \frac{5}{9}$$
.

Invece, il disco D ha per bordo la circonferenza  $(x-1)^2 + (y-1)^2 = 1$ , z=3. Scriviamo i punti del bordo di D come  $\mathbf{p} = (x,y,z)$ . Procedendo come prima, vediamo che  $\mathbf{p} = \mathbf{p_d} + \mathbf{p_n}$  con  $\mathbf{p_d} = (x+y+z)(1/3,1/3,1/3)$  e  $\mathbf{p_n} = (2x-y-z, 2y-x-z, 2z-x-y)/3$ , ed i punti del disco hanno proiezione centrale sul piano  $\pi$  nei punti 4/5 I +  $\mathbf{p_n}/(5-x-y-z) = (4,4,4) + (2x-y-z, 2y-x-z, 2z-x-y)/(3/(5-x-y-z))$ . La deviazione laterale rispetto al centro (4,4,4) dello z-buffer è il vettore (2x-y-z, 2y-x-z, 2z-x-y)/(3/(5-x-y-z)). Si tratta di verificare se la norma di questo vettore è inferiore a 5/9 allorché z=3 e  $(x-1)^2+(y-1)^2=1$ . Per z=3 il vettore di deviazione dal centro dello z-buffer diventa (2x-y-3, 2y-x-3, 6-x-y)/(3/(5-x-y-3)). La disuguaglianza da verificare è quindi una disequazione quadratica in x e y. Per ogni x dobbiamo verificare che non ci sono soluzioni reali in y. Lasciamo al lettore i tediosi calcoli.

# **RENDERING DELLA TRASPARENZA**

# MODELLAZIONE DELLA TRASPARENZA SENZA RIFRAZIONE

Il rendering della trasparenza con un procedimento che non tiene conto delle leggi della rifrazione ha senso solo quando le superficie trasparenti sono piane e sottili, e non si è interessati a rendere la separazione dei colori dovuta al fatto che la rifrazione provoca deviazioni angolari differenti a colori monocromatici di diverse lunghezze d'onda. Se le superficie sono costituite da lastre trasparenti piane, ogni raggi che le attraversa ne esce su una retta parallela alla direzione di ingresso (incidente), quindi con una deflessione ma senza una deviazione angolare. Se la lastra è sottile la deflessione è piccola, e spesso trascurabile nella resa di una scena. Quindi ha senso trascurarla, e pertanto trascurare l'effetto della rifrazione. Invece bisogna rendere l'attenuazione dovuta alla perdita di energia nell'attraversamento della lastra, e la colorazione che si ottiene se la lastra è colorata. La modellazione empirica di questi fenomeni, senza ricorso alle leggi dell'ottica relative alla rifrazione ed alla riflessione, si chiama la modellazione della *trasparenza non rifrattiva*.

Ci sono due diverse modellazioni della trasparenza non rifrattiva: *interpolata* oppure *filtrata*.

### Trasparenza interpolata

La trasparenza interpolata modella la situazione in cui una superficie frontale (assumeremo per semplificare la terminologia che si tratti di un poligono) semitrasparente (tipicamente una griglia sottile fitta), che denotiamo come poligono 1, ne copre (in tutto od in parte) una posteriore, che denotiamo con 2. Il modello assume che i colori dei due poligoni si mescolino tramite una appropriata combinazione convessa (cioè un'interpolazione) dei loro valori: l'equazione risultante per l'illuminazione è, per ogni lunghezza d'onda  $\lambda$ ,

$$\mathbf{I}_{\lambda} = (\mathbf{1} - \mathbf{k}_{t1}) \mathbf{I}_{\lambda 1} + \mathbf{k}_{t1} \mathbf{I}_{\lambda 2}$$

dove il coefficiente di trasmissione  $k_{t1}$  è la percentuale di luce che il poligono 1 lascia passare per trasparenza, e varia fra 0 e 1: quando vale 0 il poligono è completamente opaco, quando vale 1 esso è completamente trasparente e non contribuisce in niente al colore risultante. Questa modellazione è accurata nel caso si stia rendendo un poligono posteriore al quale si antepone una griglia sottile colorata, in modo che il colore risultante, pixel per pixel, sia una media del colore della griglia e di quello del poligono retrostante, con coefficienti di interpolazione che variano a secondo di quanto è fitta la griglia: si noti che, se la superficie anteriore è una griglia, i raggi di luce che la colpiscono vengono riflessi coloorati del suo colore, ma quelli che invece passano attraverso le maglie della griglia e colpiscono la superficie posteriore vengono colorati da questa con il proprio colore, senza alcun nesso al colore della griglia. In tal caso il colore di ciascun pixel è una combinazione convessa dei colori delle due superficie visibili attraverso di esso, in proporzioni ldate dalla percentuale di raggi di luce che colpisce la griglia oppure che la attraversa, e quindi da qaunto è fitta la sua maglia.

### Trasparenza filtrata

La trasparenza filtrata è il modello per la situazione in cui la superficie (poligono) anteriore è un filtro semitrasparente che ha il suo proprio colore, ed inoltra colora (con un colore di trasmissione magari diverso) una superficie (poligono) posteriore. Al variare della lunghezza d'onda  $\lambda$ , l'equazione dell'illuminazione diventa

$$\mathbf{I}_{\lambda} = \mathbf{I}_{\lambda 1} + \mathbf{k}_{t1} \mathbf{O}_{t\lambda 1} \mathbf{I}_{\lambda 2}$$

dove il coefficiente di trasmissione  $k_{t1}$  è definito come per la trasparenza interpolata, e  $O_{t\lambda 1}$  è il colore di trasparenza del poligono frontale, cioè il colore che esso dà a ciò che sta dietro. Questo modello è di solito più realistico della trasparenza interpolata, perché nella maggior parte dei casi le superficie semitrasparenti sono lastre di materiale continuo (filtri) e non griglie..

Per entrambi i modelli, se più superficie semitrasparenti insistono sullo stesso pixel, esse vanno rese dal dietro in avanti ricorsivamente (o iterativamente) perché le formule si compongano nel modo giusto.

In prossimità dei bordi di una superficie trasparente curva la modellazione finora esposta è meno soddisfacente, non per colpa degli algoritmi, ma a causa del fatto che la modellazione dei filtri come sottili lastre piane di vetro non è sufficientemente precisa, perché in quelle zone il raggio proiettore attraversa uno spessore di vetro maggiore, a causa della curvatura, e quindi la luce si attenua di più. Una formula empirica non lineare per modellare questo effetto è proposta in *[D.S. Kay, D. Greenberg, Transparency for Computer Synthesized Images, SIGGRAPH 79,* in *Computer Graphics 13, August 1979, pagg. 158-164].* In questa formula, il coefficiente di trasmissione k<sub>t</sub> viene espresso come un opportuno valore intermedio fra i coefficienti massimo e minimo della superficie, che che chiamiamo k<sub>t min</sub> e k<sub>t max</sub> e che si ottengono nei punti di spessore minimo o massimo, rispettivamente. Il valore intermedio è determinato da una espressione non lineare che involve la componente z<sub>n</sub> nella direzione della profondità del versore normale alla superficie dopo la trasformazione prospettica. La formula è la seguente:

$$k_t = k_t \min + (k_t \max - k_t \min)[1 - (1 - z_n)^m]$$

Qui m è un esponente empirico che dipende dal materiale di cui è composta la superficie, e di solito ha valore fra 2 e 3. Più la superficie è sottile, più m è grande.

Si tratta di un procedimento a <u>precisione di immagine</u> per il rendering della trasparenza filtrata, introdotto in *[A. Mammen, Transparency and Antialiasing Algorithms Implemented with the Virtual Pixel maps Technique, Computer Graphics and Applications 9, July 1989, 43-55].* 

Abbiamo visto che, per il rendering della trasparenza filtrata, i poligoni nella pila che insiste sullo stesso pixel deve essere trattati dal dietro in avanti. Il metodo di Mammen perviene all'ordinamento corretto utilizzando vari <u>z-buffer</u>. Dapprima si effettua lo z-buffer consueto per trattare tutti i poligoni opachi (e solo quelli). Poi vengono scanditi i poligoni trasparenti in vari z-buffer, a piu' passi, nel modo seguente. Lo z-buffer viene modificato per aggiungervi, oltre al colore del poligono visibile a ciascun pixel ed alla sua profondità, anche il valore di trasparenza ed un bit logico (una *flag*), inizialmente posto uguale a *falso*. A differenza dallo z-buffer solito, qui la variabile di profondità viene inizializzata al valore più alto possibile, cioè quello più vicino all'osservatore (la profondità del *viewport*, il piano di visuale). La memorizzazione della profondità

in questo z-buffer dei poligoni trasparenti avviene nel modo seguente: durante la scansione di un poligono trasparente, se ne memorizza la profondità ad un pixel solo se essa è maggiore (poligono trasparente più vicino all'osservatore) di quella del poligono opaco più vicino (che viene letta nello z-buffer dei poligoni opachi), ma minore poligono trasparente più lontano) di tutti gli altri poligoni trasparenti a quel pixel. Oltre alla profondità si memorizzano il colore ed il valore di trasparenza, e si pone *true* la flag. Al termine della scansione di tutti i poligoni trasparenti, il valore di z in questo z-buffer misura, pixel per pixel, la profondità del poligono trasparente *più lontano* dall'osservatore.

Si ripete il procedimento per determinare i poligoni trasparenti successivamente più vicini, cioè, per ogni dato pixel, la pila dei poligoni trasparenti via via più vicini, che vengono memorizzati in analoghi z-buffer successivi: ma solo pedr quei pixel la cui flag era diventata *true* al primo passo (in corrispondenza degli altri evidentemente non c' è alcun poligono trasparente). In ciascuno di questi zbuffer successivi non viene memorizzato il valore di colore e trasparenza del poligono che viene messo nel frame buffer, bensì il valore che risulta dall'applicazione della equazione di illuminazione applicata a quel poligono ed al valore risultante allo stesso pixel nello z-buffer precedente. In tal modo, nel corso della scansione, ogni z-buffer acquisisce il valore di colore risultante dai poligoni trasparenti e opachi sottostanti nella pila, cioè quello dato dalla sovrapposizione dei successivi filtri colorati (semi)trasparenti. Al termine del procedimento l'ultimo z-buffer contiene i valori finali di colore.

#### Esercizio

Una scena consiste dei seguenti poligoni, visti frontalmente nel modo in cui si ricoprono. I quattro rettangoli orizzontali o verticali si indicano con 0, 1, 2, 3 rispettivamente, in senso antiorario a partire da quello verticale a sinistra. Le profondita' dei vertici sono le seguenti:

0: vertici alti z=0, vertici bassi z=12

- 1: vertici di sinistra z= 0, vertici di destra z=12
- 2: vertici bassi z= 6, vertici alti z=12
- 3: vertici di destra z=4, vertici di sinistra z=8

Disegnare i grafici dei valori di profondità (al variare dei pixel in ascissa) dello zbuffer nelle due righe di scansione tratteggiate, dopo la scansione del primo poligono, del secondo, e così via fino all'ultimo, supponendo di scandirli nell'ordine seguente: prima il rettandolo orizzontale alto, poi quello basso, poi gli altri tre nell'ordine da sinistra a destra.



## MODELLAZIONE DELLA TRASPARENZA CON LA LEGGE DELLA RIFRAZIONE

All'attraversare la superficie di confine fra due materiali (evidentemente semitrasparenti, visto che lasciano passare la luce), un raggio di luce si *rifrange*. Il raggio incidente (cioè quello che transita nel primo materiale) viene deviato in maniera da soddisfare la seguente relazione fra l'angolo  $\theta_i$  che esso con la normale alla superficie e l'analogo angolo  $\theta_t$  formato dal raggio rifratto, cioè trasmesso al secondo materiale *(legge di Snell)*:

 $\eta_{i\lambda} \sin \theta_i = \eta_{t\lambda} \sin \theta_t$ .

Qui  $\eta_{i\lambda}$  e  $\eta_{t\lambda}$  sono i rispettivi *indici di rifrazione* dei due materiali, definiti come il rapporto tra la velocità della luce nel vuoto e nel materiale: essi variano con la lunghezza d'onda  $\lambda$  della luce. Il vuoto ha quindi indice di rifrazione 1, i materiali hanno indice di rifrazione più elevato.

E' utile derivare dalla legge di Snell una formula per la direzione del raggio trasmesso, cioè per il suo versore che denotiamo con **T**.

Per prima cosa, determiniamo il versore **M** perpendicolare al versore normale **N** che giace nel piano generato da **N** e dal versore della direzione di incidenza **I**, e punta verso il lato opposto di **I** rispetto a **N**. La proiezione di **I** lungo **N** vale **N** cos  $\theta_i$ . Perciò il vettore **N** cos  $\theta_i$  - **I** è diretto orizzontalmente, e quindi è perpendicolare a **N**.

Poiché **N** e **I** hanno norma 1 (sono versori!) e sono ortogonali, per il teorema di Pitagora la norma di **N** cos  $\theta_i$  - **I** vale  $(1 - \cos^2 \theta_i)^{1/2} = \sin \theta_i$  (il seno è positivo perché l'angolo  $\theta_i$  è fra  $0^0$  e  $90^0$ ). Quindi, normalizzando, si ottiene

 $\mathbf{M} = (\mathbf{N} \cos \theta_i - \mathbf{I}) / \sin \theta_i .$ 



Ora, per la legge di Snell, il versore del raggio trasmesso è





D'altra parte, sempre per la legge di Snell, si ha

$$\sin\theta_t I \sin\theta_i = \eta_{i\lambda} I \eta_{t\lambda}$$
.

Quindi, ponendo  $\eta_{\tau\lambda} = \eta_{i\lambda} / \eta_{t\lambda}$ , otteniamo

$$\mathbf{T} = (\eta_{\tau\lambda} \cos\theta_i - \cos\theta_t) \mathbf{N} - \eta_{\tau\lambda} \mathbf{I}$$

ed inoltre  $\cos\theta_i$  è il prodotto scalare < N, I >, e

$$\begin{split} \cos\theta_t &= (1 - \sin^2\theta_t)^{1/2} = (1 - \eta_{\tau\lambda}^2 \sin^2\theta_i)^{1/2} \\ &= [1 - \eta_{\tau\lambda}^2 (1 - \cos^2\theta_i)]^{1/2} = [1 - \eta_{\tau\lambda}^2 (1 - \langle \mathbf{N}, \mathbf{I} \rangle^2)]^{1/2} \;. \end{split}$$

Pertanto possiamo scrivere T unicamente in termini di prodotti scalari:

$$\mathbf{T} = (\eta_{\tau\lambda} < \mathbf{N}, \mathbf{I} > - [1 - \eta_{\tau\lambda}^{2} (1 - \langle \mathbf{N}, \mathbf{I} \rangle^{2})]^{1/2}) \mathbf{N} - \eta_{\tau\lambda} \mathbf{I}.$$

#### Riflessione totale interna

Nelle figure precedenti abbiamo considerato un raggio di luce che passa da un mezzo ad indice di rifrazione basso (ad esempio l'aria) ad un mezzo con indice di rifrazione più elevato (ad esempio l'acqua o il vetro): dalla legge di Snell segue che il raggio rifratto si avvicina alla direzione della normale alla superficie di separazione dei due mezzi. Nel caso contrario, di un raggio di luce che passa dall'acqua all'aria, esso dovrebbe invece allontanarsi dalla normale. C'è pero' un'eccezione a questa regola. Consideriamo un raggio di luce nell'aria pressoché radente alla superficie di separazione con l'acqua: esso penetra nell'acqua con l'angolo più aperto possibile, cioè più lontano possibile dalla normale: quest'angolo, che viene chiamato angolo limite, dalla legge di Snell risulta valere  $\operatorname{arccos}(\eta_{t\lambda}/\eta_{i\lambda})$ . Consideriamo allora un altro raggio, che viaggia dall'acqua all'aria, ma con angolo superiore all'angolo limite. Per la reversibilità dei percorsi ottici, questo raggio non può essere trasmesso all'aria (ed in effetti, se lo fosse, dovrebbe avere angolo di trasmissione superiore a 90<sup>0</sup>, e quindi essere riflesso indietro all'acqua!). In effetti, l'ottica prevede che un tale raggio sia interamente riflesso indietro al mezzo di provenienza, secondo le leggi della riflessione (riflessione totale). Di questo si deve tener conto nel rendering: in una scena subacquea un osservatore che quarda verso la superficie dell'acqua vede la luce e la scena dell'aria soprastante solo entro il cono di ampiezza pari all'angolo limite: al di fuori invece vede la riflessione interna della scena subacquea.

**RAY TRACING RICORSIVO** 

# **RAY TRACING RICORSIVO**

Ora estendiamo l'algoritmo di Ray Tracing per includere ombre, riflessioni e rifrazioni. Questa estensione dell'algoritmo è dovuta a Whitted [*T. Whitted*, <u>An Improved Illumination</u> <u>Model for Shaded Display</u>, Comm.ACM, 23 giugno 1980, 343-349] e Kay [D.S. Kay, <u>Transparency</u>, <u>Reflection and Ray Tracing for Computer Synthesized Images</u>, Master of Sciences Thesis, Program of Computer Graphics, Cornell University, Ithaca, NY, gennaio 1979].

Questa variante dell'algoritmo di Ray Tracing determina il colore di un pixel tramite il colore del punto di intersezione fra il raggio di proiezione (cioè proveniente dal punto di osservazione) che passa per il centro di quel pixel ed il primo oggetto della scena che esso interseca, in base ai consueti modelli d'illuminazione (tipicamente l'equazione dell'illuminazione di Phong), ma iterando ricorsivamente la procedura per generare altri raggi a partire dal punto di intersezione: uno riflesso ed uno rifratto. Ciascuno di questi due raggi viene tracciato per determinare (se e) guale oggetto della scena interseca per primo: per tale oggetto, al punto di intersezione, si calcola l'intensità di luce data dall'equazione dell'illuminazione di Phong, considerando ora come direzione dell'osservatore quella individuata dalla direzione di provenienza del raggio: in altre parole, si considera il contributo all'illuminazione del punto inizialmente osservato dato dalla luce inviata verso di esso dall'oggetto più vicino nella direzione speculare rispetto all'osservatore (e nella direzione rifratta, se il punto giace su una superficie parzialmente trasparente). Ovviamente attenuiamo le rispettive illuminazioni moltiplicandole per il fattore (compreso fra 0 e 1) di riflettività o di rifrazione del materiale al punto originalmente osservato. Poi si procede ricorsivamente: si considerano i due raggi generati ed ai due punti della scena che hanno toccato li si riflette e rifrange nuovamente, quindi ciascuno di essi genera due nuovi raggi, che vengono tracciati per trovare ulteriori punti di intersezione con la scena e sommarne i contributi di luce (che sono quindi contributi di due riflessioni o rifrazioni consecutive, ossia del secondo ordine), e così via.

Inoltre, per calcolare le ombre, tracciamo un altro raggio addizionale, dal punto di intersezione con l'oggetto, verso ciascuna delle sorgenti di luce (questi raggi si chiamano *raggi d'ombra*). Se uno di questi raggi d'ombra interseca un oggetto lungo la sua traiettoria, e quindi non raggiunge la sorgente di luce, allora il punto dell'oggetto da cui il raggio è partito è in ombra rispetto alla data sorgente di luce, ed in tal caso l'algoritmo di rendering ignora il contributo del raggio luminoso proveniente da quella sorgente.

Quale è il motivo di aggiungere un nuovo raggio riflesso ed uno rifratto? Per spiegare questo punto, consideriamo l'equazione dell'<u>illuminazione di Phong</u>. Essa determina, sia pure in via euristica e non fisica, l'intensità di luce nel punto osservato causata dalla luce ambientale, quella diffusa generata dalle sorgenti e quella riflessa specularmente a partire dalle sorgenti: però non tiene conto della luce che arriva sul punto osservato non direttamente dalle sorgenti, bensì riflessa da altri oggetti della scena (o meglio, ne tiene conto implicitamente nel termine di illuminazione ambientale, che però è impreciso perché non direzionale: il nostro modello di luce ambientale non dipende dalla posizione degli oggetti e delle luci, mentre la luce riflessa da un oggetto sugli altri ne risente). Tracciare dal punto osservato un raggio nuovo raggio nella direzione riflessa e sommare alla illuminazione del punto della scena colpito da questo nuovo raggio significa tenere conto della luce riflessa dall'oggetto colpito, ma solo di quella proveniente dalla direzione esattamente speculare rispetto alla direzione di osservazione. Riassumendo, il metodo aggiunge all'illuminazione diretta un ulteriore contributo dovuto alla riflessione da altri oggetti, ma

solo quella proveniente dalla direzione di massima riflessione, trascurando contributi di riflessione in direzioni deflesse rispetto a quella perfettamente speculare. Pertanto il Ray Tracing ricorsivo, già al primo livello di ricorsività, accentua l'illuminazione diretta aggiungendovi l'effetto di una riflessione speculare. Iterando una seconda volta il procedimento aggiungiamo alla luce proveniente dalla riflessione perfettamente speculare da parte di un altro oggetto l'ulteriore contributo di illuminazione che quel dato oggetto riceve a sua volta dalla direzione perfettamente speculare rispetto a quella in cui viene osservato (la direzione del primo raggio riflesso). Quindi l'iterazione ricorsiva, passo dopo passo, raffina e rende via via più preciso il rendering degli effetti delle riflessioni secondarie fra oggetti, ma sempre limitatamente alla specularità perfetta, non a quella alle varie deviazioni angolari (se si intendessero includere contributi di riflessioni non perfettamente speculari si dovrebbe utilizzare un metodo più sofisticato del Ray Tracing, ossia un metodo che genera raggi non solo nella direzione speculare ma anche in direzioni distribuite, possibilmente vicine a questa: ciò si può fare mediante il Ray Tracing ricorsivo stocastico, ovvero distribuito statisticamente). Analoga evidenziazione avviene per i contributi della interrifrazione fra oggetti. Perciò il Ray Tracing ricorsivo è un metodo ideale per evidenziare gli effetti perfettamente speculari e rifrattivi, e non per quelli diffusi: la variante di Ray Tracing ricorsivo stocastico permette un trattamento più accurato della riflessione e rifrazione con parziale diffusione, ma richiede un maggior numero di raggi a meno di non rassegnarsi ad elevati livelli di rumore. Un procedimento stocastico di illuminazione globale, che tiene conto di tutti i raggi riflessi e non solo di quelli speculari, basato su una equazione integrale ricorsiva, è stato introdotto in [J.T.Kajiya, The Rendering Equation, SIGGRAPH 86, in Computer Grahics 20(4), Agosto 1986, pagg. 143-150].

Ora studiamo in maggior dettaglio i raggi secondari tracciati per effettuare il rendering della riflessione speculare e della trasparenza rifrattiva.

I raggi d'ombra, di riflessione e di rifrazione sono chiamati raggi secondari, per distinguerli dai raggi di proiezione primari provenienti dall'osservatore Se l'oggetto intersecato da un raggio riflette in modo speculare, allora viene generato un raggio di riflessione, diretto verso la <u>direzione R speculare a quella di incidenza</u>. Se l'oggetto è trasparente, e se non si verifica una <u>riflessione totale interna</u>, allora viene inviato un raggio di rifrazione nell'oggetto nella direzione T stabilita dalla <u>legge di Snell sulla rifrazione</u>.



Inoltre tutti questi raggi di riflessione e rifrazione, qualora intersechino un nuovo oggetto, possono produrre in maniera ricorsiva raggi di ombra, riflessione e rifrazione. Così i raggi formano una struttura gerarchica ad albero. I nodi dell'albero sono i punti di intersezione,

ed i segmenti sono i raggi. Si prefissa il numero massimo di generazioni (cioè di fasi di ricorrenza) che verranno considerate in quest'albero, in base ad una soglia prefissata di energia luminosa al di sotto della quale ci si deve necessariamente ridurre dopo un tale numero di generazioni, a causa della attenuazione complessiva dei contributi ricorsivi, causata, generazione dopo generazione, dalla moltiplicazione successiva dei coefficienti di riflessività e di trasmissibilità rifrattiva. Si può usare una stima uniforme, basata sul valore massimo di tali coefficienti fra i materiali della scena: alla n-sima generazione ciascun contributo non può superare questo valore elavato alla potenza n. Oppure si può optare per una scelta di numero massimo di generazioni a passo variabile, ossia adattiva, tenendo conto degli effettivi valori dei coefficienti di attenuazione per i materiali incontrati da ciascun raggio nel corso delle sue riflessioni o rifrazioni alle varie generazioni, e fermando la ricorrenza quando il risultato diventa inferiore ad una soglia prefissata.



Nell'algoritmo di Whitted, un "ramo" dell'albero termina nei casi seguenti: se i raggi riflessi e rifratti non intersecano alcun oggetto, se si raggiunge il massimo numero predefinito di generazioni o se il sistema esaurisce la memoria.

Per calcolare l'illuminazione di un pixel, i rami dell'albero che hanno inizio con il raggio proiettore che passa per il centro di quel pixel sono percorsi ricorsivamente dal basso in alto: a ciascun nodo (cioè punto di intersezione) l'intensità è determinata in base all'intensità dei suoi immediati discendenti. In tal modo, risalendo l'albero, ogni raggio accumula energia, e la somma di tutti i raggi derivati da quello che passa per il centro del pixel è l'illuminazione totale del pixel.



L'equazione dell'illuminazione usata da Whitted è la seguente equazione ricorsiva:

$$I_{\lambda} = I_{a\lambda}k_a O_{d\lambda} + \sum_{1 \le i \le m} S_i f_{att_i} I_{p\lambda_i} \Big[ k_d O_{d\lambda} (N, L_i) + k_s O_{s\lambda} (R, V)^n \Big] + k_r I_{r\lambda} + k_i I_{t\lambda}$$

dove le sorgenti di luce sono m, e per ogni sorgente di luce i:

- $I_{r\lambda}$  è l'intensità del raggio riflesso,  $I_{t\lambda}$  quella del raggio trasmesso (rifratto),  $I_{a\lambda}$  è l'intensità della luce ambientale,  $I_{p\lambda i}$  è l'intensità della i-sima sorgente, ed i corrispondenti coefficienti O rappresentano la distribuzione spettrale del colore di queste forme di illuminazione al variare della frequenza  $\lambda$ ;
- k<sub>a</sub> e k<sub>s</sub> sono i coefficienti di attenuazione per la luce ambientale e la luce riflessa specularmente, k<sub>r</sub> e k<sub>t</sub> sono i coefficienti di riflessione e di rifrazione: tutti questi coefficienti possono variare fra 0 e 1 (il coefficiente k<sub>r</sub> è nient'altro che il coefficiente di riflessione speculare k<sub>s</sub> del materiale al punto in cui è stato generato il raggio riflesso, ed analogamente per k<sub>t</sub> ed il raggio rifratto);
- per ogni raggio riflresso o rifratto, il versore del punto di osservazione cambia generazione dopo generazione: volta per volta è quello diretto verso il punto della scena in cui sono stati generati tali raggi.

Per ogni punto della scena toccato da un raggio proiettore proveniente dall'osservatore, i valori di  $I_{r\lambda}$  e  $I_{t\lambda}$  sono determinati da questa stessa equazione applicata questa volta ai punti della superfice più vicina che vengono toccati dai raggi riflessi e trasmessi (da qui la ricorsività del procedimento).

Per approssimare l'attenuazione dovuta alla distanza, per ogni raggio l'algoritmo di Whitted moltiplica il valore calcolato di  $I_{\lambda}$  per l'inverso della distanza percorsa dal raggio.

Il valore di *Si* dovrebbe essere 1 oppure 0 a seconda del fatto, che la prima sorgente di luce sia visibile oppure no dal punto illuminato (cioè che il punto sia illuminato o in ombra).

Invece che trattare *Si* come una funzione a valori 0 e 1, il metodo la sostituisce con una funzione continua del coefficiente di trasmissione  $k_t$  degli altri oggetti precedentemente attraversati in modo che gli oggetti parzialmente trasparenti assorbano meno luce di quelli opachi.

Questo algoritmo presenta alcuni problemi che presentiamo in una pagina separata.

```
PSEUDOCODICE:
```

```
Seleziona l'osservatore (centro di proiezione) e la finestra;
Per ogni scan line nell'immagine {
       Per ogni pixel nella scan line {
               Determina il raggio che proviene dell'osservatore e che attraversa il pixel;
               pixel=RT trace (raggio, 1);
       }
}
/*intersezione
RT_colore RT_trace (RT_raggio raggio, int profondità) {
       Determina l'intersezione più vicina tra il raggio e un oggetto;
       Se (l'oggetto è colpito) {
               Calcola la normale all'intersezione;
               restituisci RT_ombra (l'oggetto colpito più vicino, l'intersezione, la normale, la
               profondità);
       }
       altrimenti
               restitusici il valore dello sfondo;
}
/*ombre, riflessioni e rifrazioni
RT_colore RT_ombra (RT_oggetto oggetto1, RT_raggio raggio1, RT_punto punto1,
                          RT_normale normale1, int profondità)
{
       RT colore colore1;
       RT_raggio Araggio, Braggio, Craggio;
       RT colore Acolore, Bcolore;
       colore1= ambiente;
       Per ogni luce {
               Craggio = raggio di luce che proviene da un punto;
               Se il prodotto della normale con la direzione della luce è positivo {
                      Calcola quanta luce è bloccata dalle superfici opache o trasparenti,
                       usa il termine di diffusione e quello speculare e aggiungili a colore1;
               }
       Se la profondità è minore della profondità massima {
               Se l'oggetto è riflettente {
                      Araggio=raggio in direzione della riflessione;
                      Acolore=RT_trace (Araggio, profondità+1);
                      scala Acolore con il coefficiente di riflessione speculare ed aggiungilo a
                      colore1;
               }
               Se l'oggento è trasparente {
                      Braggio = raggio in direzione della rifrazione;
                      Se non si ha una riflessione interna totale {
                              Bcolore=RT_trace (Braggio, profondità +1);
                              scala Bcolore con il coefficiente di trasmissione ed aggiungilo a
                              colore1;
                       }
               }
       }
       restituisci colore1;
}
```

```
Spiegazione dello pseudocodice dell'algoritmo del Ray Tracing Ricorsivo
```

Il metodo RT\_trace determina il punto di intersezione più vicino al punto di visuale di un raggio di proiezione con un oggetto della scena, e richiama il metodo RT\_ombra per

determinare l'iluminazione in quel punto. Dapprima, RT\_ombra calcola la componente di luce ambientale dell'illuminazione, poi traccia raggi d'ombra verso ogni sorgente di luce per determinare se il punto di intersezione sia in ombra rispetto a qualcuna delle sorgenti, che quindi non dà apporto all'illuminazione. Poi il metodo verifica se l'oggetto intersecato è riflettente od opaco, e se è opaco o (semi)trasparente. Un oggetto opaco scherma totalmente la luce, mentre un oggetto riflettente lascia continuare il raggio incidente in un raggio riflesso, ed uno trasparente in un raggio rifratto: per tracciare questi eventuali nuovi raggi viene richiamato ricorsivamente su entrambi il metodo RT\_trace. La direzione del raggio riflesso dipende solo dalla direzione di incidenza e dalla direzione normale alla superficie; per calcolare la direzione del raggio rifratto sono necessari gli indici di rifrazione dei due materiali per cui il raggio incidente sta transitando, che vengono inclusi fra i dati di ogni raggio generato. RT\_trace costruisce l'albero del raggio proiettore, ma lo mantiene solo il tempo necessario a determinare il colore corrente del pixel.

#### Esercizi sul Ray Tracing ricorsivo

Per ragioni di leggibilità, nello svolgimento del primo esercizio indichiamo il prodotto scalare con  $\langle \cdot, \cdot \rangle$  invece che con  $(\cdot, \cdot)$ , ed indichiamo i vettori con lettere con sopra una barra.

*Esercizio 1.* Una scena consiste di due sfere  $T_1 \in T_2$  con centro l'origine e raggio rispettivamente 1 e 7, dalle pareti perfettamente speculari (cioè con coefficiente di riflettività pari a 1), ed un quadrato rosso puro con coefficiente di riflessione  $\frac{1}{2}$  sul piano z = 2, con i lati paralleli agli assi coordinati, con centro in (0,0,2) e lato pari a 3. L'osservatore si trova al punto (1,0,2), la luce a (0,4,0) ed è bianca. Il coefficiente di diffusione di Lambert del quadrato è  $\frac{1}{2}$ , tutti gli altri sono zero.



Si determini cosa vede l'osservatore quando guarda verso il punto (0,0,1); si ricavi il risultato mediante il procedimento di ray tracing ricorsivo, con due generazioni di tracciamento di raggi (oltre a quello originale). L'esponente di Phong di ciascun oggetto è 2.

Svolgimento.

Troviamo  $\overline{V}_1$ , il versore del raggio incidente in p:

$$\overline{V}_1 = \frac{(o-p)}{\|o-p\|} = \frac{(1,0,2) - (0,0,1)}{\|(1,0,1\|)} = \frac{(1,0,1)}{\sqrt{2}}$$

La normale in p è nella direzione del raggio:

$$\overline{N}_1 = (0, 0, 1)$$

Il raggio proiettore riflesso in p è dato da una espressione del tipo

$$\overline{R} = 2\langle \overline{N}, \overline{V} \rangle \overline{N} - \overline{V}$$

e quindi, poiché  $\langle \overline{V}_1,\overline{\overline{N}}_1\rangle=\frac{1}{\sqrt{2}},$ troviamo

$$\overline{R}_1 = \frac{2}{\sqrt{2}}(0,0,1) - \left(\frac{1}{\sqrt{2}},0,\frac{1}{\sqrt{2}}\right) = \left(-\frac{1}{\sqrt{2}},0,\frac{1}{\sqrt{2}}\right)$$

Il punto p è in ombra perché la retta che congiunge il punto l (luce) con p interseca prima un altro punto della sfera. Quindi non bisogna calcolare il raggio d'ombra. Se non stessimo usando il Ray Tracing ricorsivo, il punto sarebbe stato in ombra e l'osservatore avrebbe visto un colore nero.

**Prima generazione.** Per calcolare il contributo della generazione successiva, spostiamo l'osservatore in p. Troviamo il punto di intersezione q del raggio riflesso in p con il quadrato rosso.

$$r(t) = p + t\overline{R}_1 = \begin{cases} x = -\frac{1}{\sqrt{2}}t\\ y = 0\\ z = 1 + \frac{1}{\sqrt{2}}t \end{cases}$$

Mettiamo a sistema  $p + t\overline{R}_1$  con il piano z = 2:

$$\begin{cases} x = -\frac{1}{\sqrt{2}}t\\ y = 0\\ 1 = \frac{1}{\sqrt{2}}t \end{cases}$$

quindi $t=\sqrt{2},$ e pertanto il punto osservato è q=(-1,0,2) Il versore normale è

$$\overline{N}_2 = -\overline{N}_1 = (0, 0, -1)$$

Troviamo  $\overline{V}_2$ , il versore della direzione del raggio proiettore di prima generazione incidente in q:

$$\overline{V}_2 = \frac{p-q}{\|p-q\|} = \frac{(0,0,1) - (-1,0,2)}{\|(-1,0,2\|)} = \frac{(1,0,-1)}{\sqrt{2}} = -\overline{R}_1$$

Questo versore è quindi il nuovo versore dell'osservatore, che ora immaginiamo localizzato al punto p. Troviamo la direzione del proiettore riflesso  $\overline{R}_2$  al punto q:

$$\overline{R}_2 = 2\left\langle \left(\overline{V}_2, \overline{N}_2\right) \overline{N}_2 - \overline{V}_2 \right.$$
$$= 2\left\langle \left(\frac{1}{\sqrt{2}}, 0, \frac{-1}{\sqrt{2}}\right), (0, 0, -1) \right\rangle (0, 0, -1) - \left(\frac{1}{\sqrt{2}}, 0, \frac{-1}{\sqrt{2}}\right)$$
$$= \left(\frac{-1}{\sqrt{2}}, 0, \frac{-1}{\sqrt{2}}\right)$$

Troviamo il versore della direzione della luce  $\overline{L}_2$  :

$$\overline{L}_2 = \frac{l-q}{\|l-q\|} = \frac{(0,4,0) - (-1,0,2)}{\|(1,4,-2)\|} = \frac{(1,4,-2)}{\sqrt{21}}$$

Infine, calcoliamo il versore  $\overline{L'}_2$  riflesso speculare del versore della luce:

$$\overline{L'_2} = 2 \left\langle \overline{L_2}, \overline{N_2} \right\rangle \overline{N_2} - \overline{L_2} \\= 2 \left\langle \frac{(1, 4, -2)}{\sqrt{21}}, (0, 0, -1) \right\rangle (0, 0, -1) - \frac{(1, 4, -2)}{\sqrt{21}} \\= \frac{(-1, -4, -2)}{\sqrt{21}}$$

Poiché la luce è bianca, il quadrato è rosso ed i suoi coefficienti di diffusione e di riflessione sono  $\frac{1}{2}$ , il contributo del raggio di prima generazione alla luminosità è

$$\frac{1}{2} \langle \overline{N_2}, \overline{L_2} \rangle (1_R, 0_G, 0_B) + \frac{1}{2} \langle \overline{V_2}, \overline{L'_2} \rangle^2 (1_R, 1_G, 1_B) \\
= \frac{1}{\sqrt{21}} (1_R, 0_G, 0_B) + \frac{2}{21} (1_R, 1_G, 1_B) \\
= (0.313_R, 0.095_G, 0.095_B) \quad (1)$$

(il primo termine è quello di Lambert ed il secondo quello di Phong).

**Seconda generazione.** Supponiamo che l'osservatore stia ora in q. Troviamo il punto di intersezione w tra la retta r uscente da q in direzione  $\overline{R}_2$  e la sfera  $T_2$ . L'equazione parametrica della retta è  $r(t) = q + t\overline{R}_2$ , ossia

$$\begin{cases} x = -1 - \frac{t}{\sqrt{2}} \\ y = 0 \\ z = 2 + \frac{t}{\sqrt{2}} \end{cases}$$

Intersecando con la sfera  $x^2 + y^2 + z^2 = 49$  troviamo il parametro t del punto di intersezione:

$$(-1 - \frac{t}{\sqrt{2}})^2 + (2 - \frac{t}{\sqrt{2}})^2 = 49$$
$$1 + \frac{t^2}{2} + \frac{2t}{\sqrt{2}} + 4 + \frac{t^2}{\sqrt{2}} - 2\frac{2t}{\sqrt{2}} = 49$$
$$t^2 - \sqrt{2}t - 44$$

Le soluzioni sono:

$$t_{1,2} = \frac{\sqrt{2} \pm 4\sqrt{89}}{2}$$
 cioè  $t_1 = 19.5$  e  $t_2 = -18.13$ 

Un valore di t negativo corrisponde ad un'intersezione che avviene prima dell'inizio del raggio riflesso: quindi usiamo il valore positivo.

Il punto è: w = (-6.2, 0, -3.2)Troviamo la normale in w:

$$\overline{N}_3 = \frac{(0-w)}{\|0-w\|} = (\frac{6,2}{7}, 0, \frac{3,2}{7})$$

Troviamo  $\overline{V}_3$ , il versore del raggio proiettore di seconda generazione incidente in w (ossia il nuovo vettore della direzione dell'osservatore che si è spostato in q):

$$\overline{V}_3 = \frac{(q-w)}{\|q-w\|} = \frac{(-1,0,2)(-6.2,0,-3.2)}{\|(5.2,0,5.2)\|} = \left(\frac{1}{\sqrt{2}},0,\frac{1}{\sqrt{2}}\right) = -\overline{R}_2$$

Troviamo il versore  $\overline{L}_3$ :

$$\overline{L}_3 = \frac{l-w}{\|l-w\|} = \frac{(0,4,0)(-6.2,0,-3.2)}{\|(6.2,4,3.2)\|} = \frac{(6.2,4,3.2)}{\sqrt{64.7}}$$

Troviamo la direzione  $\overline{L'}_3$  riflessa speculare di  $\overline{L}_3$  rispetto a  $\overline{N}_3$ :

$$\overline{L'}_3 = 2 \left\langle \frac{(6.2, 0, 3.2)}{7}, \frac{(6.2, 4, 3.2)}{\sqrt{64.7}} \right\rangle \left(\frac{6.2}{7}, 0, \frac{3.2}{7}\right) - \frac{(6.2, 4, 3.2)}{\sqrt{64.7}} \\ = 2 \frac{48.68}{7\sqrt{64.7}} \frac{(6.2, 0, 3.2)}{7} - \frac{(6.2, 4, 3.2)}{\sqrt{64.7}} = 0.247(6.2, 0, 3.2) - 0.124(6.2, 4, 3.2) \\ = (0.763, 0.498, 0.394)$$

Poiché la luce è bianca ed i coefficienti di diffusione e di riflessione della sfera esterna sono rispettivamente 0 e 1, ma il raggio proiettore si era riflesso al punto q che appartiene ad un materiale di coefficiente di riflessione uguale a 1, il contributo del raggio di seconda generazione alla luminosità è

$$\frac{1}{2} \langle \overline{V_3}, \overline{L'_3} \rangle^2 (1_R, 1_G, 1_B) = (0.335_R, 0.335_G, 0.335_B)$$
(2)

Infine calcoliamo l'intensità della luce che vede l'osservatore, sommando i risultati della prima e seconda generazione, ottenuti in (1) e (2):

$$\begin{split} I_p &= 0 + \left[\frac{1}{2} \langle \overline{N_2}, \overline{L_2} \rangle (1_R, 0_G, 0_B) + \frac{1}{2} \langle \overline{V_2}, \overline{L'_2} \rangle^2 (1_R, 1_G, 1_B)\right] \\ &+ \frac{1}{2} \left[ \langle \overline{V_3}, \overline{L'_3} \rangle^2 (1_R, 1_G, 1_B) \right] \\ &= (0.313_R, 0.095_G, 0.095_B) + (0.335_R, 0.335_G, 0.335_B) \\ &= (0.648_R, 0.43_G, 0.43_B) \end{split}$$

*Esercizio* 2. Un osservatore è collocato nel punto O = (2, 2, 2). Una sorgente isotropa di luce è collocata nel punto L = (0, 0, 10). L'osservatore guarda verso l'origine. La scena si compone di una lastra metallica speculare sul piano z = 0, uno strato di materiale trasparente di coefficiente di rifrazione  $\sqrt{2}$  che occupa la striscia 0 < z < 1, un quadrato sul piano z = 3 di estensione  $\{-1.5 \le x \le 1.5, -1.5 \le y \le 1.5\}$  ed una sfera con centro (5, -5, 5) e raggio 2. Si segua il raggio di osservazione, si determini il punto di intersezione p con la lastra trasparente, e da tale punto si traccino:

- il raggio riflesso  $R_1$
- il raggio rifratto  ${\cal T}_0$
- il raggio  $V_{sorg}$  che si dirige verso la sorgente di luce.

Inoltre

- 1. Per il terzo raggio, si determini se esso raggiunge la sorgente di luce oppure se prima interseca qualche altro oggetto della scena. Il punto p è in luce o in ombra?
- 2. Per il primo raggio, si determini se esso interseca qualche altro oggetto della scena o si perde nello sfondo.
- 3. Per il secondo raggio, si trovi il punto q in cui esso tocca la lastra speculare. Dal punto q si tracci il suo raggio riflesso  $t_1$ , e si tracci un nuovo raggio diretto verso la luce . Questo nuovo raggio raggiunge la luce oppure prima interseca qualche altro oggetto? Il punto q è in luce o in ombra?
- 4. Si determini il punto s in cui il raggio  $t_1$  esce dalla lastra trasparente, e si segua il raggio rifratto (che esce quindi dalla lastra ed entra nell'aria). Tale raggio colpisce qualche altro oggetto? Dal punto s si tracci un nuovo raggio diretto verso la sorgente di luce. Tale raggio arriva alla sorgente di luce o viene fermato da qualche altro oggetto? Il punto s è in luce o in ombra?



Svolgimento.

• Per trovare il punto p tracciamo la retta passante per l'osservatore e diretta verso l'origine, quindi con direzione  $\overline{V}_1$  data da

$$\overline{V}_1 = \frac{(o - oss)}{\|o - oss\|} = \frac{(0, 0, 0) - (2, 2, 2)}{\|(-2, -2, -2\|)} = \left(-\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}\right)$$
$$r(t) = oss + t\overline{V}_1 = \begin{cases} x = 2 + t\frac{-1}{\sqrt{3}} \\ y = 2 + t\frac{-1}{\sqrt{3}} \\ z = 2 + t\frac{-1}{\sqrt{3}} \end{cases}$$

Il punto p di intersezione della retta con la superficie dello strato trasparente è la soluzione del sistema della equazione della retta  $oss+t\overline{V}_1$ e del pianoz=1:

$$\begin{cases} x - 2 = \frac{-t}{\sqrt{3}} \\ y - 2 = \frac{-t}{\sqrt{3}} \\ -1 = \frac{-t}{\sqrt{3}} \end{cases}$$

quindi p = (1, 1, 1)

• Per trovare il versore del raggio riflesso  $\overline{R}_1$ , usiamo come sempre l'identità  $\overline{R}_1 = 2 < \overline{N}_1, \overline{V}_p > \overline{N}_1 - \overline{V}_p$ . Qui  $\overline{V}_p$  è il versore della direzione dell'osservatore visto dal punto p, ed ovviamente è l'opposto del versore  $\overline{V}_1$  calcolato prima. Infatti

$$\overline{V}_p = \frac{(oss - p)}{\|oss - p\|} = \left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right)$$
$$\overline{N}_1 = (0, 0, 1)$$
$$(\overline{N}_1, \overline{V}_p) = \frac{1}{\sqrt{3}}$$

Quindi si ottiene

$$\overline{R}_1 = \frac{2}{\sqrt{3}}(0,0,1) - \left(\frac{1}{\sqrt{3}},\frac{1}{\sqrt{3}},\frac{1}{\sqrt{3}}\right) = \left(-\frac{1}{\sqrt{3}},-\frac{1}{\sqrt{3}},\frac{1}{\sqrt{3}}\right)$$

• Punto 2: troviamo la retta passante per p e con direzione  $\overline{R}_1$ .

$$\begin{cases} x = 1 - \frac{t}{\sqrt{3}} \\ y = 1 - \frac{t}{\sqrt{3}} \\ z = 1 + \frac{t}{\sqrt{3}} \end{cases}$$

Mettiamo a sistema con il quadrato (z = 3):

$$\begin{cases} x = 1 - \frac{t}{\sqrt{3}} \\ y = 1 - \frac{t}{\sqrt{3}} \\ 2 = \frac{t}{\sqrt{3}} \end{cases}$$

quindi  $p_1 = (-1, -1, 3).$ 

• Per trovare il versore del<br/>raggio rifratto  $T_0$ , usiamo l'identità

$$T_0 = \left[\eta < N_1, V_p > -\sqrt{1 - \eta^2 (1 - \langle N_1, V_p \rangle)}\right] N_1 - \eta V$$

dove  $\eta = \frac{1}{\sqrt{2}}$ . Si trova:

$$\begin{bmatrix} \frac{1}{\sqrt{2}}(\frac{1}{\sqrt{3}}) - \sqrt{1 - \frac{1}{2}(1 - \frac{1}{3})} \end{bmatrix} (0, 0, 1) - \frac{1}{\sqrt{2}} \left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right)$$
$$= \left(-\frac{1}{\sqrt{6}}, -\frac{1}{\sqrt{6}}, -\frac{2}{\sqrt{6}}\right)$$

• Ecco il versore del raggio d'ombra  $V_{sorg}$  diretto verso la sorgente di luce:

$$\overline{V}_{sorg} = \frac{(L-p)}{\|L-p\|} = \frac{(0,0,10) - (1,1,1)}{\|(-1,-1,9\|)} = (\frac{-1}{\sqrt{83}}, \frac{-1}{\sqrt{83}}, \frac{9}{\sqrt{83}})$$

• Punto 1:  $\overline{V}_{sorg}$  raggiunge la sorgente di luce in L o interseca qualche oggetto?

Trovo la retta passante per p e con direzione  $\overline{V}_{sorg}$ .

$$\begin{cases} x = 1 - \frac{t}{\sqrt{83}} \\ y = 1 - \frac{t}{\sqrt{83}} \\ z = 1 + \frac{9t}{\sqrt{83}} \end{cases}$$

Mettiamo a sistema con il quadrato (z = 3):

$$\begin{cases} x = 1 - \frac{t}{\sqrt{83}} \\ y = 1 - \frac{t}{\sqrt{83}} \\ 2 = \frac{9t}{\sqrt{83}} \end{cases}$$

quindi  $p_2 = (\frac{7}{9}, \frac{7}{9}, 3).$ 

Il punto appartiene al quadrato perché esso ha estensione  $\{-1.5 \le x \le 1.5, -1.5 \le y \le 1.5\}$ ; pertanto il punto p è in ombra.

• Punto 3: determiniamo il punto q. Troviamo la retta passante per p e con direzione  $\overline{T}_0$ .

$$\begin{cases} x = 1 - \frac{t}{\sqrt{6}} \\ y = 1 - \frac{t}{\sqrt{6}} \\ z = 1 - \frac{2t}{\sqrt{6}} \end{cases}$$

Mettiamo a sistema con la lastra speculare  $\left(z=0\right)$  :

$$\begin{cases} x = 1 - \frac{t}{\sqrt{6}} \\ y = 1 - \frac{t}{\sqrt{6}} \\ \frac{1}{2} = \frac{t}{\sqrt{6}} \end{cases}$$

quindi  $q=(\frac{1}{2},\frac{1}{2},0)$ 

• Per trovare il riflesso  $t_1$ :

$$t_1 = 2 < \overline{N}_2, \overline{T}_q > \overline{N}_2 - \overline{T}_q$$
$$\overline{T}_q = \frac{(p-q)}{\|p-q\|} = \left(\frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{2}{\sqrt{6}}\right)$$
$$\overline{N}_2 = (0, 0, 1)$$
$$(\overline{N}_2, \overline{T}_q) = \frac{2}{\sqrt{6}}$$

quindi

$$\overline{t}_1 = \frac{4}{\sqrt{6}}(0,0,1) - \left(\frac{1}{\sqrt{6}},\frac{1}{\sqrt{6}},\frac{2}{\sqrt{6}}\right) = \left(-\frac{1}{\sqrt{6}},-\frac{1}{\sqrt{6}},\frac{2}{\sqrt{6}}\right)$$

• Il punto q in è ombra?

$$\overline{L} = \frac{(L-q)}{\|L-q\|} = \frac{(0,0,10) - \left(\frac{1}{2},\frac{1}{2},0\right)}{\sqrt{\frac{201}{2}}} = \frac{\left(-\frac{1}{2},-\frac{1}{2},10\right)}{\sqrt{\frac{201}{2}}}$$
$$\begin{cases} x = \frac{1}{2} + t\frac{-2}{\sqrt{201}}\\ y = \frac{1}{2} + t\frac{-2}{\sqrt{201}}\\ z = 0 + t\frac{10\sqrt{2}}{\sqrt{201}} \end{cases}$$

Mettiamo a sistema con il quadrato (z = 3):

$$\begin{cases} x = \frac{1}{2} + t \frac{-2}{\sqrt{201}} \\ y = \frac{1}{2} + t \frac{-2}{\sqrt{201}} \\ t = \frac{3\sqrt{201}}{10\sqrt{2}} \end{cases}$$

quindi  $p_L = \left(\frac{5\sqrt{2}-6}{10\sqrt{2}}, \frac{5\sqrt{2}-6}{10\sqrt{2}}, 3\right) = (0.15, 0.15, 3)$ . Il punto appartiene al quadrato perché ha estensione  $\{-1.5 \le x \le 1.5, -1.5 \le y \le 1.5\}$ ; Pertanto il punto q è in ombra.

• Punto 4: Troviamo il punto s dato dall'intersezione di  $\overline{T}_1$  con la lastra trasparente.

Retta passante per q con direzione  $\overline{T}_1$ :

$$\begin{cases} x = \frac{1}{2} - \frac{t}{\sqrt{6}} \\ y = \frac{1}{2} - \frac{t}{\sqrt{6}} \\ z = 0 + \frac{2t}{\sqrt{6}} \end{cases}$$

Mettiamo a sistema con la lastra trasparente (z = 1):

$$\begin{cases} x = \frac{1}{2} - \frac{t}{\sqrt{6}} \\ y = \frac{1}{2} - \frac{t}{\sqrt{6}} \\ \frac{1}{2} = \frac{t}{\sqrt{6}} \end{cases}$$

quindi s = (0, 0, 1)

• Troviamo il versore del raggio rifratto  $\overline{T}_2$ :

$$T_2 = \left[\eta < N_3, T_s > -\sqrt{1 - \eta^2 (1 - \langle N_3, T_s \rangle)}\right] N_3 - \eta T_s$$

dove  

$$\eta = \frac{\sqrt{2}}{1}$$
  
 $\overline{N}_3 = (0, 0, -1)$   
 $\overline{T}_s = \frac{(q-s)}{\|q-s\|} = \frac{(\frac{1}{2}, \frac{1}{2}, 0) - (0, 0, 1)}{\frac{\sqrt{3}}{\sqrt{2}}} = (\frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, -\frac{2}{\sqrt{6}})$   
 $(\overline{N}_3, \overline{t}_s) = \frac{2}{\sqrt{6}}$   
Quindi

$$\overline{T}_2 = \left[\sqrt{2}\left(\frac{2}{\sqrt{6}}\right) - \sqrt{1 - 2\left(1 - \frac{4}{6}\right)}\right] (0, 0, -1) - \sqrt{2}\left(\frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, -\frac{2}{\sqrt{6}}\right)$$
$$= \left(-\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right)$$

• Calcoliamo la retta passante per s e direzione  $\overline{T}_2$ .

$$\begin{cases} x = 0 - \frac{t}{\sqrt{3}} \\ y = 0 - \frac{t}{\sqrt{3}} \\ z = 0 + \frac{2t}{\sqrt{3}} \end{cases}$$

Mettiamo a sistema con il quadrato (z = 3):

$$\begin{cases} x = 0 - \frac{t}{\sqrt{3}} \\ y = 0 - \frac{t}{\sqrt{3}} \\ 2 = \frac{t}{\sqrt{3}} \end{cases}$$

quindi il punto di intersezione è n = (-2, -2, 3), che non appartiene al quadrato  $\{-1.5 \le x \le 1.5, -1.5 \le y \le 1.5\}$  e non interseca nemmeno la sfera (avente centro in (5,-5,5) e raggio 2).

• Troviamo la retta che passa per s e si dirige verso la sorgente.

$$\overline{L}_2 = \frac{(L-s)}{\|L-s\|} = \frac{(0,0,10) - (0,0,1)}{9} = (0,0,1)$$
$$\begin{cases} x = 0\\ y = t\\ z = 1+t \end{cases}$$

Mettiamo a sistema con il quadrato(z = 3) : t = 2

$$\begin{cases} x = 0\\ y = 0\\ z = 3 \end{cases}$$

quindi  $s_L = (0, 0, 3)$  é il punto in cui la retta incontra il quadrato. Pertanto il punto q è in ombra. **Esercizio 3**. Nella scena dell'Esercizio 2 si sposti la sorgente di luce al punto (0,10,10). Inoltre, si assuma che la lastra speculare abbia coefficiente di riflettivita' 1, esponente di Phong 3 e coefficiente di Lambert 0, la sfera sia rossa con coefficiente di Lambert ½ e tutti gli altri coefficienti zero, il quadrato sia nero ed opaco, lo strato trasparente abbia coefficiente di Lambert 0, coefficiente di riflettività ¼, coefficiente di trasmissione ½ ed esponente di Phong 1. Come cambiano le risposte?

### Svolgimento.

Ricapitoliamo il problema. L'osservatore è collocato nel punto O = (2, 2, 2). Una sorgente isotropa di luce è collocata nel punto L = (0, 10, 10). L'osservatore guarda verso l'origine. La scena si compone di una lastra metallica speculare sul piano z=0, uno strato di materiale trasparente di coefficiente di rifrazione 2 che

occupa la striscia 0<z<1, un quadrato sul piano z=3 di estensione  $\{-1.5 \le x \le 1.5 \le 1.5 \le 1.5 \le 1.5 \le x \le 1.5 \le 1.5$ 

# 1.5, -1.5 $\leq$ y $\leq$ 1.5 $\}$ ed una sfera con centro (5,-5,5) e raggio 2.

Si segua il raggio di osservazione, si determini il punto di impatto **p** con la lastra trasparente, e da tale punto si traccino:

- il raggio riflesso R1
- il raggio rifratto To
- il raggio V<sub>sorg</sub> che si dirige verso la sorgente di luce
- 1) Per il terzo raggio, si determini se esso raggiunge la sorgente di luce oppure se prima interseca qualche altro oggetto della scena. Il punto **p** è in luce o in ombra?
- 2) Per il primo raggio, si determini se esso interseca qualche altro oggetto della scena oppure si perde nello sfondo.
- 3) Per il terzo raggio, si trovi il punto q in cui esso tocca la lastra speculare. Dal punto q si tracci il suo raggio riflesso t1, e si tracci un nuovo raggio diretto verso la luce. Questo nuovo raggio raggiunge la luce oppure prima interseca qualche altro oggetto? Il punto q è in luce o in ombra?
- 4) Si determini il punto f in cui il raggio t<sub>1</sub> esce dalla lastra trasparente, e si segua il raggio rifratto (che esce quindi dalla lastra ed entra nell'aria). Tale raggio colpisce qualche altro oggetto? Dal punto f si tracci un nuovo raggio diretto verso la sorgente di luce. Tale raggio arriva alla sorgente di luce o viene fermato da qualche altro oggetto? Il punto s è in luce o in ombra?
- 5) Si determini l'illuminazione vista dall'osservatore sommando i contributi di questi raggi.

Dati del Problema:

Lastra Metallica Specu	lare →	z=0 K <sub>r</sub> = 1	$K_d = 0$	γ = 3		
Sfera Rossa	÷	$(x-5)^2$ K <sub>r</sub> = 0	+ $(y+5)^2$ + $(z-5)^2$ K <sub>d</sub> = $\frac{1}{2}$	$(5)^2 = 4$		
Quadrato Nero Opaco	<i>→</i>	$Q: \begin{cases} F \\ K_r = 0 \end{cases}$	$P(x, y, z) \mid z = 3$ $K_{d} = 1$	$3; -\frac{3}{2} \le x \le$	$\leq \frac{3}{2}; -\frac{3}{2};$	$\leq y \leq \frac{3}{2}$
Strato Trasparente	÷	Tra z: K <sub>r</sub> = ¼	=0 e z=1 K <sub>d</sub> =0	$K_t = \frac{1}{2}$	γ = 1	η = 1
Osservatore:	O (2, 2,	2)				
Luce:	S (0, 10, 10)					

Direzione di Osservazione:  $\overline{W}$  (-2, -2, -2)

Svolgimento. Determiniamo il punto di osservazione tracciando la retta direzione  $\overline{W}$  passante per il punto O.

$$\begin{cases} x = 2 - 2t \\ y = 2 - 2t \\ z = 2 - 2t \end{cases}$$

Intersechiamo la retta con lo strato trasparente, ovvero con il piano di equazione z = 1:

$$\begin{cases} x = 2 - 2t \\ y = 2 - 2t \\ z = 2 - 2t \\ z = 1 \end{cases}$$

Si ottiene t=1/2.

Quindi il punto osservato è P=(1, 1, 1).

Verifichiamo se il punto P è in luce o in ombra. Dobbiamo tracciare la retta che passa per il punto P e ha per direzione il vettore L che da P si dirige verso la sorgente di luce.

$$L_{0} = \frac{S - P}{\|S - P\|} = \frac{(0, 10, 10) - (1, 1, 1)}{\| \cdot \|} = \frac{(-1, 9, 9)}{\sqrt{163}}$$
  
r: 
$$\begin{cases} x = 1 - t \\ y = 1 + 9t \\ z = 1 + 9t \end{cases}$$
 retta che si dirige verso la sorgente di luce

Verifichiamo se questa retta interseca qualche oggetto.

La retta non può intersecare la sfera poiché il vettore direzione ha coordinata y positiva mentre la sfera ha centro in y=-5 e raggio 2, quindi al massimo la sfera ha coordinata y=-3.

Verifichiamo quindi solo l'intersezione con il quadrato posto in z=3:

$$r:\begin{cases} x=1-t\\ y=1+9t\\ z=1+9t\\ z=3 \end{cases}$$
$$t=\frac{2}{9} \longrightarrow x=\frac{7}{9} \qquad y=3 > \frac{3}{2}$$

Poiché il punto trovato non appartiene al quadrato, il punto P non è oscurato da nessun altro oggetto, quindi è il luce.

Per calcolare il contributo di illuminazione diretto abbiamo bisogno del raggio riflesso e del raggio rifratto.

Raggio Riflesso:
$$R_0 = 2 \langle N_0, V_0 \rangle N_0 - V_0$$
dove $N_0 = (0,0,1)$ 

$$V_{0} = \frac{O - P}{\|O - P\|} = \frac{(2, 2, 2) - (1, 1, 1)}{\| \cdot \|} = \frac{1}{\sqrt{3}} \begin{pmatrix} 1\\1\\1\\1 \end{pmatrix}$$
$$R_{0} = \frac{2}{\sqrt{3}} \begin{pmatrix} 0\\0\\1 \end{pmatrix} - \frac{1}{\sqrt{3}} \begin{pmatrix} 1\\1\\1 \end{pmatrix} = \frac{1}{\sqrt{3}} \begin{pmatrix} -1\\-1\\1\\1 \end{pmatrix}$$
г Raggio Rifratto:

$$\mathbf{\Gamma}_{0} = \mathbf{N}_{0} \left[ \eta \left\langle \mathbf{N}_{0}, \mathbf{V}_{0} \right\rangle - \sqrt{1 - \eta^{2} \left(1 - \left\langle \mathbf{N}_{0}, \mathbf{V}_{0} \right\rangle^{2}\right)} \right] - \eta \mathbf{V}_{0}$$

dove

N<sub>0</sub> e V<sub>0</sub> sono i versori calcolati al passo precedente  $\eta = \frac{1}{\sqrt{2}}$  perchè il raggio passa dall'area verso il materiale

$$T_{0} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \left[ \frac{1}{\sqrt{2}} \frac{1}{\sqrt{3}} - \sqrt{1 - \frac{1}{2} \left(1 - \frac{1}{3}\right)} \right] - \frac{1}{\sqrt{6}} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{6}} \begin{pmatrix} -1 \\ -1 \\ -2 \end{pmatrix}$$

Calcoliamo ora il contributo di illuminazione:

$$I_{p} = K_{r} \langle R_{0}, L_{0} \rangle \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + K_{d} O_{d\lambda} \langle L_{0}, N_{0} \rangle + K_{t} \langle T_{0}, L_{0} \rangle \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

 $L_0$  è il versore che indica la direzione della luce dove

$$\left\langle R_0, L_0 \right\rangle = \frac{1}{\sqrt{489}}$$
$$\left\langle T_0, L_0 \right\rangle = \frac{-26}{\sqrt{978}}$$

Il prodotto scalare tra T<sub>0</sub> e L<sub>0</sub> è negativo quindi l'angolo compreso tra i due vettori è maggiore di 90°. Allora, non c'è contributo dato dalla rifrazione, e non c'è neanche la diffusione perché il coefficiente di Lambert è zero. L'illuminazione nel punto P diventa:

$$I_{p} = \frac{1}{4} \frac{1}{\sqrt{489}} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.01 \\ 0.01 \\ 0.01 \end{pmatrix}$$

Vediamo ora i contributi di illuminazione dati dai punti colpiti rispettivamente dal raggio riflesso e del raggio rifratto.

Tracciamo la retta che passa per P e ha direzione Ro e vediamo se interseca qualche oggetto nella scena.

$$r:\begin{cases} x=1-t\\ y=1-t\\ z=1+t \end{cases}$$

Controlliamo le intersezioni con il quadrato posto in z=3

$$\begin{cases} x = 1 - t \\ y = 1 - t \\ z = 1 + t \\ z = 3 \\ t = 2 \rightarrow x = -1 \quad y = -1 \end{cases}$$

Il punto C(-1, -1, 3) appartiene al quadrato. Dovremmo calcolare il contributo di illuminazione fornito alla generazione precedente ma, dato che il quadrato è opaco e nero, il suo contributo sarà nullo.

Seguiamo ora il raggio rifratto e vediamo in quale punto colpisce la lastra speculare posta sul piano z=0

$$r:\begin{cases} x=1-t\\ y=1-t\\ z=1-2t \end{cases}$$
 retta per P direzione T<sub>0</sub>

Intersezione retta r e piano z=0 :

$$\begin{cases} x = 1 - t \\ y = 1 - t \\ z = 1 - 2t \\ z = 0 \end{cases}$$
$$t = \frac{1}{2} \longrightarrow x = \frac{1}{2} \qquad y = \frac{1}{2}$$

Il punto di intersezione Q ha coordinate:  $Q = \left(\frac{1}{2}, \frac{1}{2}, 0\right)$ 

Controlliamo se il punto Q è in luce tracciando la retta direzione  $L_1$  passante per Q e vediamo se tale retta interseca il quadrato.

$$L_{1} = \frac{S - Q}{\|S - Q\|} = \frac{(0,10,10) - (\frac{1}{2}, \frac{1}{2}, 0)}{\|\cdot\|} = \frac{(-\frac{1}{2}, \frac{19}{2}, 10)}{\sqrt{762}} = \frac{1}{\sqrt{762}}(-1,19,20) = \begin{pmatrix}-0.04\\0.69\\0.72\end{pmatrix}$$

$$\begin{cases} x = \frac{1}{2} - t \\ y = \frac{1}{2} + 19t \\ z = 20t \\ z = 3 \end{cases}$$
$$t = \frac{3}{20} \qquad \rightarrow \qquad y = \frac{1}{2} + \frac{3}{20} > \frac{3}{2} \quad \text{quindi il punto non appartiene al quadrato}$$

Il punto Q è in luce.

Troviamo ora tutti i vettori necessari per il calcolo del contributo di illuminazione.

$$N_{1} = (0,0,1)$$

$$V_{1} = -T_{0} = \frac{1}{\sqrt{6}} \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}$$

$$T_{1} = 2 \langle N_{1}, V_{1} \rangle N_{1} - V_{1} = \frac{4}{\sqrt{6}} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} - \frac{1}{\sqrt{6}} \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix} = \frac{1}{\sqrt{6}} \begin{pmatrix} -1 \\ -1 \\ 2 \end{pmatrix}$$
 raggio riflesso

Il contributo di illuminazione del punto Q è dato dalla seguente equazione:

$$I_{Q} = K_{i} \begin{bmatrix} K_{r} \langle T_{1}, L_{1} \rangle^{3} \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} \end{bmatrix}$$

dove

$$\langle T_1, L_1 \rangle^3 = (0.32)^3 = 0.034$$

$$I_{Q} = \frac{1}{2} \begin{bmatrix} 1 * 0.034 \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} = \begin{pmatrix} 0.01 \\ 0.01 \\ 0.01 \end{pmatrix}$$

Questo vettore è stato moltiplicato per il coefficiente di rifrazione della generazione di raggi precedente perché la luce nel punto Q non arriva in modo diretto ma solo dopo la rifrazione subito sullo strato sovrastante.

Seguiamo la retta che passa per Q e ha direzione  $T_1$  e troviamo il punto di intersezione di questa retta con lo strato trasparente posto in z = 1

$$r:\begin{cases} x = \frac{1}{2} - t \\ y = \frac{1}{2} - t \\ z = 2t \\ z = 1 \end{cases}$$
$$t = \frac{1}{2} \longrightarrow x = 0 \quad y = 0 \quad z = 1$$

Il punto di intersezione F ha coordinate F=(0,0,1)Verifichiamo se il punto è in luce.

$$L_{2} = \frac{S - F}{\|S - F\|} = \frac{(0, 10, 10) - (0, 0, 1)}{\|\cdot\|} = \frac{(0, 10, 9)}{\sqrt{181}}$$

$$r:\begin{cases} x=0\\ y=10t\\ z=1+9t \end{cases}$$

Intersechiamo la retta con il piano z=3 per verificare che il punto F non sia coperto dal quadrato:

$$\begin{cases} x = 0 \\ y = 10t \\ z = 1 + 9t \\ z = 3 \end{cases}$$
$$t = \frac{2}{9} \longrightarrow y = \frac{20}{9} > \frac{3}{2}$$

Quindi il punto F è in luce. Troviamo il raggio rifratto:

$$N_{2} = (0, 0, -1)$$
$$V_{2} = -T_{1} = \frac{1}{\sqrt{6}} \begin{pmatrix} 1 \\ 1 \\ -2 \end{pmatrix}$$

 $\mathbf{T}_{2} = N_{2} \left[ \eta \left\langle N_{2}, V_{2} \right\rangle - \sqrt{1 - \eta^{2} \left(1 - \left\langle N_{2}, V_{2} \right\rangle^{2}\right)} \right] - \eta V_{2}$ Raggio Rifratto: N<sub>2</sub> e V<sub>2</sub> sono i versori calcolati al passo precedente

dove

 $\eta = \sqrt{2}$  perchè il raggio passa dal materiale all'aria

$$\mathbf{T}_{2} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} \left[ \sqrt{2} \frac{2}{\sqrt{6}} - \sqrt{1 - 2\left(1 - \frac{4}{6}\right)} \right] - \frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ 1 \\ -2 \end{pmatrix} = \frac{1}{\sqrt{3}} \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix}$$

Calcoliamo ora il contributo di illuminazione di F:

$$I_{F} = K_{r} \left[ K_{t} \langle T_{2}, L_{2} \rangle \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \right]$$

dove

$$\left\langle T_2, L_2 \right\rangle = \frac{-1}{\sqrt{543}}$$

Poiché il prodotto scalare tra T<sub>2</sub> e L<sub>2</sub> è negativo, il punto F non fornisce contributo di illuminazione.

Calcoliamo ora l'illuminazione totale del punto P:

$$I_{TOT} = I_P + I_Q + I_F = \begin{pmatrix} 0.01\\ 0.01\\ 0.01 \end{pmatrix} + \begin{pmatrix} 0.01\\ 0.01\\ 0.01 \end{pmatrix} + 0 = \begin{pmatrix} 0.02\\ 0.02\\ 0.02 \end{pmatrix}$$

Esercizio 4. Un triangolo T nello spazio ha vertici in (1,1,0), (2,2,3), (1,2,m). C'è una sorgente di luce in (-1,-1,1). Il piano z=0 è speculare. Un raggio di luce parte da L e si muove verso l'origine: quando raggiunge l'origine si riflette, e la sua nuova direzione di movimento ovviamente si ottiene cambiando il segno della sola componente z della direzione che il raggio aveva prima della riflessione. Il raggio riflesso colpisce in triangolo T? Se sì, dove?

#### Svolgimento.

La direzione del raggio che va verso l'origine è data dal vettore -L=(1,1,-1). Il raggio riflesso si propaga dall'origine in direzione R=(1,1,1). Il triangolo giace in un piano ax+by+cz=d, i cui coefficienti sono tali che il piano contiene i tre vertici di T, ossia soddisfano le identità

> a+b=d 2a+2b+3c=d

#### a+2b+mc=d.

Sostituendo la seconda identità nella prima si trova d+3c=0; sostituendola nella terza si ha b+mc=0. Quindi c=-d/3 e b=md/3. Pertanto, dalla prima delle tre identità sopra, si ha a=(1-m/3)d. Siamo liberi di dilatare dello stesso fattore tutti e quattro i coefficienti (in tal caso il piano non cambia), quindi prendiamo d=1. Allora i coefficienti del piano di T sono

$$a = 1 - m/3$$
,  $b = m/3$ ,  $c = -1/3$ .

Il piano quindi è

(1-m/3)x + m/3 y - 1/3z = 1

(per verifica, si controlli direttamente che i tre vertici soddisfano questa equazione). La retta uscente dall'origine in direzione R ha equazioni cartesiane x=y=z e quindi equazioni parametriche x=t, y=t, z=t). Osserviamo che (1-m/3) + m/3 - 1/3 = 2/3. Quindi la retta interseca il piano quando il parametro t soddisfa 1=(1-m/3)t + m/3 t - 1/3 t = 2/3 t, ossia

t=3/2. Il punto di intersezione è P=(3/2, 3/2, 3/2). Il punto P giace dentro T o fuori? Proiettiamo T sul piano x,y, ossia poniamo z=0. Il punto P giace in T se e solo se la proiezione di P su questo piano giace nella proiezione di T (perché il triangolo è convesso). La proiezione di T è il triangolo nel piano con vertici in (1,1), (2,2), (1,2). La proiezione di P è il punto (3/2, 3/2). Questo punto è esattamente il punto medio del segmento che congiunge i vertici (1,1) e (2,2), quindi il punto medio di uno dei lati della proiezione di T, pertanto esso giace nella proiezione di T. Concludiamo quindi che P giace in T.

**Esercizio 5.** Un triangolo T nello spazio ha vertici in (1,0,0), (0,1,0), (0,0,1). Esso ha coefficiente di diffusione pari a  $\frac{1}{2}$  e coefficiente di riflessione di luce ambientale pari a  $\frac{1}{2}$ , ma il coefficiente di riflessione speculare è 0; è colorato con un gradiente lineare che ha colore rosso puro nel vertice (1,0,0), verde puro in (0,1,0) e blu puro in (0,0,1). C'è una sorgente di luce in (-1,-1,1). Il piano z=0 è una superficie bianca parzialmente speculare con coefficiente di riflessione speculare 4/5, coefficiente di diffusione di Lambert 1/5, coefficiente di riflessione per la luce ambientale 1/10 ed esponente di Phong 2. L'osservatore si trova nel punto (-1,-1,2) e guarda verso l'origine. La luce ambientale, di intensità 1, è bianca.

i) Che colore vede l'osservatore se si tiene conto solo della luce ambientale?

ii) Che colore vede l'osservatore se nel calcolo si usa il ray Tracing ricorsivo a due generazioni (due generazioni di raggi riflessi)?

#### Svolgimento.

La direzione del raggio riflesso che si propaga dall'origine è, come nel problema precedente, R=(1,1,1). Invece ora il triangolo giace nel piano x+y+z=1: il punto P di intersezione è quindi

$$\mathsf{P} = (1/3, 1/3, 1/3).$$

Ovviamente P giace dentro il triangolo, perché questo triangolo sottende l'intero ottante positivo. Osserviamo che i vertici del triangolo sono i tre vettori canonici di

base  $e_1$ ,  $e_2$ ,  $e_3$ , ed il punto P è la loro combinazione convessa ( $e_1+e_2+e_3$ )/3 (in effetti, è il baricentro del triangolo). Osserviamo che la normale al triangolo T che punta verso il semipiano che contiene la sorgente di luce è la stessa in tutti i punti del triangolo e coincide con uno dei due versori normali al piano del triangolo, e precisamente con

$$N_{P} = (-1, -1, -1)/3^{\frac{1}{2}}$$

i) Se si tiene conto solo della\luce ambientale e l'osservatore guarda verso l'origine, vede una superficie bianca con fattore di riflessone alla\luce ambientale 1/10, e quindi vede il colore grigio di intensità 1/10.

ii) Il contributo all'illuminazione dato dal raggio iniziale è 1/10 di grigio per ;la luce ambiente (come visto al punto precedente), più il contributo di Lambert, più il contributo di Phong. Poiché il coefficiente di diffusione del materiale del piano di base è 1/5 ed il versore L che individua la sorgente di luce è (-1, -1, 1)/3<sup>1/2</sup>, il contributo di Lambert è grigio di intensità 1/5 (L,N) = 1/5 1/3<sup>1/2</sup> 0.1156. Come nel problema precedente, il versore riflesso è R=(1,1,1)/3<sup>1/2</sup>, mentre il versore V che individua l'osservatore è il normalizzato di (-1, -1, 2), ossia V=(-1, -1, 2)/6<sup>1/2</sup> Poiché il coefficiente di riflessione speculare è 0.8 e l'esponente di Phong 2, il contributo del termine di Phong per il raggio originale è grigio di intensità 0.8 (R,V)<sup>2</sup> = 0.8 (2<sup>1/2</sup>/3)<sup>2</sup> = 1.6/9 0.1778.

Quindi il contributo totale del raggio originale è 1/10 + 0.1156 + 0.1778 = 0.3333.

Per calcolare il contributo del raggio riflesso di prima generazione, dobbiamo considerare il punto P del triangolo T che esso colpisce ed il suo colore, e dobbiamo aggiungere per questo punto i contributi di Lambert e di Phong, attenuati tramite il coefficiente di riflettività del raggio riflesso, ossia il coefficiente di riflettività speculare del materiale che l'ha generato, che vale 4/5=0.8. Si potrebbe aggiungere a questi due contributi anche quello dovuto alla luce ambientale, ma di norma la luce ambientale è considerata una luce di sottofondo tenue ed uguale per tutti i materiali, e quindi non la si aggiunge in maniera ricorsiva ai contributi speculari calcolati nelle generazioni successive.

Determiniamo anzitutto il colore del triangolo T al punto P. Per linearità, il colore di P, in ciascun canale, è la corrispondente combinazione convessa dei colori dei vertici. Quindi, nel canale rosso, il colore di P ha intensità 1/3, perché un vertice in questo canale ha intensità 1 e gli altri due hanno intensità zero. Analogamente nei canali verde e blu: quindi il colore di P è (1/3, 1/3, 1/3), ossia grigio di intensità 1/3. Il coefficiente di diffusione di Lambert del triangolo T vale  $\frac{1}{2}$ .

Rispetto al punto P la direzione della luce si ottiene normalizzando il vettore L-P: indichiamo con  $L_P$  il vettore normalizzato, che vale

$$L_{P} = (-5, -5, -1)/51^{\frac{1}{2}}$$
.

Analogamente, la direzione dell'osservatore è il versore  $V_P$  ottenuto normalizzando V-P, e vale

$$V_{P} = (-5, -5, 1)/51^{\frac{1}{2}}$$
.

Ne segue che il contributo di Lambert del primo raggio riflesso generato (da moltiplicare poi per il fattore di riflessione speculare 0.8 della superficie sul piano

z=0 che ha generato tale raggio riflesso) è  $\frac{1}{2}$  (L<sub>P</sub>,V<sub>P</sub>) moltiplicato per il colore del punto P. In ciascun canale tale colore ha intensità 1/3, e quindi il contributo di Lambert dovuto a tale raggio è

0.8 1/3 
$$\frac{1}{2}$$
 (25+25-1)/ 51 <sup>$\frac{1}{2}$</sup>  = (2/15) (49/51 <sup>$\frac{1}{2}$</sup> ) ~ 0.9148.

Poiché il coefficiente di riflessione speculare del triangolo vale 0, il primo raggio riflesso non produce alcun contributo di riflessione speculare (termine di Phong), ed inoltre esso non si riflette in un nuovo raggio riflesso. Ne segue che l'illuminazione totale del Ray Tracing ricorsivo a due generazioni vale 0.3333 + 0.9148 = 1.2481 (naturalmente, dovremo poi riscalare questi valori di illuminazione dividendo per il massimo valore di illuminazione fra tutti i pixel).

**Esercizio 6.** Si consideri una ciotola K di forma emisferica di raggio 1 appoggiata, nel suo punto più basso, all'origine, ossia tangente all'origine al piano {z=0} e contenuta nel semispazio superiore { $z \ge 0$ }: quindi il bordo della ciotola è un cerchio J sul piano {z=1}. Sia  $\varphi$  l'angolo di latitudine visto dal suo centro, orientato in modo che il bordo J corrisponda a  $\varphi = \pi/2$  ed il polo sud (ossia l'origine) a  $\varphi = 0$ . La ciotola è colorata con un gradiente di rosso, con intensità uguale a cos  $\varphi$ , ha coefficiente di diffusione di Lambert e di riflessione entrambi  $\frac{1}{2}$ , esponente di Phong 1 ed è riempita di acqua (coefficiente di rifrazione 1.33) fino a metà della sua altezza. Un osservatore sta nel punto v=(1,1,3/2) e guarda verso il punto (0,0,1/2). Una lampada puntiforme che emette luce bianca alla massima intensita' in maniera isotropa è collocata al punto (0,0,1).

- a. Ponendo uguale a 0 il coefficiente di riflessione, di luce ambientale e di Lambert e ad 1 il coefficiente di trasmissione dell'acqua, che colore vede l'osservatore, nel modello del Ray Tracing con equazione di illuminazione di Phong? (Ovviamente, si tracci il raggio proiettore finché non tocca la ciotola, ma senza generare a quel punto ulteriori raggi riflessi e rifratti: però, per arrivare alla ciotola, il proiettore deve venire rifratto una volta.
- b. Ora poniamo il coefficiente di riflessione della ciotola pari a ½, quello dell'acqua pari a 1/10 e quello di trasmissione dell'acqua 9/10. Immaginiamo che la scena sia completata da uno sfondo nero non riflettente che la avvolge completamente, e che non ci sia luce ambientale. Rieseguiamo il calcolo con il Ray Tracing ricorsivo, con due generazioni di raggi. Come cambia la risposta?

Svolgimento. Per la parte (a), basta rammentare che il versore del raggio rifratto è

$$T = (\eta_{\tau\lambda} < N, I > - [1 - \eta_{\tau\lambda}^2 (1 - \langle N, I \rangle^2)]^{1/2}) N - \eta_{\tau\lambda} I,$$

dove  $\eta_{\tau\lambda} = 1/1.33 = \frac{3}{4} = 0.75$ , ed il versore incidente I vale, nel caso presente,

$$I = \frac{1}{\sqrt{3}}(1, 1, 1)$$

Pertanto

$$T = \left(\frac{3}{4}\frac{1}{\sqrt{3}} - \sqrt{\left(1 - \frac{9}{16}\left(1 - \frac{1}{3}\right)\right)}\right)N - \frac{3}{4}I = \left(\frac{\sqrt{3}}{4} - \sqrt{\frac{5}{8}}\right)N - \frac{3}{4}I = \left(\frac{\sqrt{3}}{4} - \frac{\sqrt{5}}{2\sqrt{2}}\right)N - \frac{3}{4}I$$
$$= -\frac{\sqrt{3}}{4}(N+I) = -\left(\frac{\sqrt{3}/4}{\sqrt{3}/4}\right)$$
$$\frac{\sqrt{3}}{2\sqrt{2}}$$

Il raggio proiettore colpisce la superficie dell'acqua al punto  $(0,0,\frac{1}{2})$ , e da lì continua, senza perdita di energia, lungo la semiretta

$$r(t) = \begin{pmatrix} 0 \\ 0 \\ 1/2 \end{pmatrix} - t \begin{pmatrix} \sqrt{3}/4 \\ \sqrt{3}/4 \\ \frac{\sqrt{5}}{2\sqrt{2}} \end{pmatrix} = \begin{pmatrix} -\frac{\sqrt{3}}{4}t \\ -\frac{\sqrt{3}}{4}t \\ \frac{1}{2}\left(1 - \sqrt{\frac{5}{2}t}\right) \end{pmatrix}$$

con t > 0. La semisfera ha equazione  $x^2 + y^2 + (z-1)^2 = 1$ , con z ≤ 1. Il punto di intersezione quindi è dato dall'equazione

$$1 = \frac{3}{8}t^{2} + \frac{1}{4}\left(1 + \sqrt{\frac{5}{2}}t\right)^{2} = \frac{3}{8}t^{2} + \frac{\frac{5}{2}t^{2} + \sqrt{10}t + 1}{4} = t^{2} + \frac{\sqrt{5}}{2\sqrt{2}}t + \frac{1}{4}$$

La soluzione con t > 0 è

$$t_0 = -\frac{\sqrt{5}}{4\sqrt{2}} + \sqrt{\frac{5}{16} + \frac{3}{4}} = -\frac{\sqrt{5}}{4\sqrt{2}} + \frac{\sqrt{17}}{4} = \frac{1}{4} \left(\sqrt{17} - \sqrt{\frac{5}{2}}\right) \sim 0.6355$$

Troviamo quindi il punto di intersezione  $r(t_0) \sim (-0.2752, -0.2752, 0.1847)$ . L'altezza dell'intersezione è quindi circa il 18.5% dell'altezza totale dell'emisfero, ossia una discesa rispetto al centro della sfera di circa 81.5%. Questo corrisponde ad un angolo  $\varphi$  con coseno uguale a 0.185. Poiché l'emisfero è colorato con una sfumatura di rosso pari a cos  $\varphi$ , l'intensità del colore al punto osservato è 0.185 del rosso puro, ossia rosso al 18.5% di luminosità. A questo dato bisogna applicare l'equazione di illuminazione di Phong, ricordando per' che dopo la generazione del raggio rifratto il calcolo va fatto *come se l'osservatore si spostasse nel punto dove viene generato tale raggio*, quindi in (0,0,1/2): in altre parole, la direzione del versore V dell'osservatore, per il calcolo dell'equazione di Phong al punto r(t\_0), è l'opposto del nuovo vettore incidente, ossia

$$V = \begin{pmatrix} \sqrt{3}/4 \\ \sqrt{3}/4 \\ \frac{\sqrt{5}}{2\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 0.4330 \\ 0.4330 \\ 0.7906 \end{pmatrix}$$

Inoltre, la luce è collocata nel centro dell'emisfero, e quindi il versore della direzione della luce coincide con il versore radiale N, che è proporzionale a A = (0,0,1) -  $r(t_0) = (0.2752, 0.2752, 0.8153)$ . Si ha quindi ||A||=0.903429, e normalizzando si trova

$$N = L = \begin{pmatrix} 0.3046\\ 0.3046\\ 0.9025 \end{pmatrix}$$

Poiché L coincide con N, anche il versore R di massima riflessione coincide con L e N. Quindi, a meno dei rispettivi coefficienti, il termine di Lambert è < L , N > e quello di Phong è < R , V > = < L , V >. Poiché entrambi i coefficienti di diffusione e di riflessione valgono  $\frac{1}{2}$ , il risultato dell'equazione di Phong è il colore rosso al 18.5% moltiplicato per

$$\frac{1}{2}$$
 < R , N+V >=  $\frac{1}{2}$  < N , N+V > =  $\frac{1}{2}$  < N , N > +  $\frac{1}{2}$  < N , V > =  $\frac{1}{2}$  (1+ 0.9773) = 0.9887

Il risultato è che il colore rosso della ciotola viene visto con intensità 18.26%.

Per la parte (b), i calcoli sono tediosi ma banali (purché ci si ricordi che ad ogni generazione bisogna spostare l'osservatore all'origine dei raggi generati), e sono lasciati al lettore.

# MALFUNZIONAMENTI INERENTI AL RAY TRACING RICORSIVO

Il Ray Tracing ricorsivo è molto sensibile agli errori di arrotondamento. Infatti le cordinate del punto in cui il raggio principale interseca un oggetto sono usate come origine di un nuovo raggio d'ombra, che quindi in quel punto ha parametro t con valore 0. Se un arrotondamento numerico dà a t un valore piccolo ma negativo, l'algoritmo di Ray tracing Ricorsivo deduce che l'oggetto interferisce col nuovo raggio d'ombra, oscurando sé stesso (cioè proiettando ombra su sé stesso). In tal caso vengono generate ombre sbagliate sull'oggetto. Non si può risolvere il problema evitando di eseguire il test di intersezione fra un oggetto e i raggi di ombra da esso generati, perchè ci sono oggetti che possono proiettare davvero ombre su sé stessi: gli oggetti concavi. Bisogna invece predefinire una soglia di tolleranza ed eseguire il test di intersezione solo per i valori di t inferiori a quella soglia (in valore assoluto).

La seguente figura illustra un problema del Ray Tracing ricorsivo dovuto alla rifrazione.



Il raggio d'ombra L3 (generato quando il raggio di luce T1 che sta attraversando l'oggetto solido semitrasparente sulla destra ne esce) viene inviato in direzione della sorgente di luce. Però questo raggio esce anch'esso dal solido semitrasparente prima di arrivare alla sorgente. Nella realtà (cioè nel modello fisico della visibilità della sorgente) all'uscita dall'oggetto anche il raggio L3

dovrebbe subire una deviazione dovuta alla rifrazione. Il metodo ignora questa deviazione perchè al momento in cui L3 viene generato si conosce la direzione della sorgente di luce, ma non la direzione verso cui inviare il raggio per farlo arrivare alla sorgente di luce tenendo conto della successiva rifrazione all'uscita del solido (questa successione di raggio d'ombra interno al solido nella direzione giusta seguito da raggio rifratto esterno che colpisce esattamente la sorgente di luce è illustrata in rosso nella Figura). Quest'ultima avviene con un angolo che dipende dall'angolo che il raggio forma con la frontiera a quel punto. Per esattamente si potrebbero usare alcuni accorgimenti calcolarla sul campionamento come inviare non un raggio d'ombra, ma un pennello di raggi, tracciarli tutti e interpolare (questo è un procedimento possibile ma oneroso) o sull'algoritmo stesso. Invece si utilizza un Ray Tracing a ritroso, cioè a partire dalle sorgenti luminose.

# VARIANTI DEL RAY TRACING RICORSIVO PER AUMENTARE L'EFFICIENZA

L'algorimo di Ray tracing ricorsivo è molto oneroso, perchè ad ogni intersezione un raggio ne genera m+2 (un raggio riflesso, uno rifratto e m raggi d'ombra, dove m è il numero di sorgenti di luce). Se ci si limita a n passi di ricorrenza, il numero di nuovi raggi creati da ogni intersezione è dell'ordine di m2<sup>n</sup>. Ecco alcuni metodi per migliorare l'efficienza.

#### Buffer di memoria di oggetti (item buffer)

Per ciascun pixel si considera il raggio che parte dal punto di visuale ed attraversa il centro del pixel. Invece che provare ad intersecare il raggio con tutti gli oggetti della scena, si determina l'oggetto intersecato per primo mediante un algoritmo meno oneroso, come ad esempio lo <u>z-buffer</u> (applicato a quel pixel) e poi si calcola il punto di intersezione tridimensionale con solo questo oggetto, e se ne ricava la luminosità con il metodo del Ray tracing ricorsivo.

#### Mappa di riflessione (reflection map)

E' un'approssimazione che ha lo scopo di ridurre la mole di calcoli sui raggi riflessi, cioè sugli oggetti che non sono direttamente visibili. Questi oggetti vengono suddivisi in due gruppi, quelli con visibilità riflessa alta o bassa, in base ad una soglia prefissata. La valutazione di quanto sia elevata la visibilità indiretta di un oggetto Q è determinata dall'angolo solido che il complesso di tutti gli oggetti direttamente visibili sottende quando li si guarda dal baricentro di Q.

Per gli oggetti di visibilità elevata si applica il normale Ray tracing ricorsivo. Per gli altri, invece, si usa una mappa di riflessione: la direzione del raggio riflesso che li interseca si usa come puntatore in una mappa di riflessione che contiene l'informazione sulla luminosità. Naturalmente, il rendering in questo modo risulta distorto, perché l'informazione sulla luminosità è calcolata a partire dal baricentro di Q invece che dal punto di visuale. Però questa approssimazione dà ottimi risultati perchè le immagini riflesse e rifratte sono spesso assai distorte.

La mappe di riflessione si calcola proiettando tutti gli oggetti su una superficie sferica centrata nel centro della scena, ed abbastanza grande perchè nessun oggetto sia troppo vicino alla superficie sferica (il che genererebbe distorsioni eccessive). Per una migliore precisione, si dovrebbe calcolare una diversa mappa di riflessione per ogni punto che è il baricentro di un oggetto Q non direttamente visibile, o almeno di quelli fra questi oggetti che sono più grandi od ubicati più centralmente nei gruppi di oggetti non direttamente visibili. Le direzioni uscenti da questi punti sono parametrizzati in maniera diretta dai punti delle sfere corrispondenti: invece, se si usa una sola sfera, si ha un errore di parallasse tanto maggiore quanto più l'oggetto Q è lontano dal centro della sfera e vicino alla sua superficie: per questo si sceglie una sfera grande. Se la sfera ha raggio infinito l'errore di parallasse si annulla.

# Adattamento variabile del controllo della profondità gerarchica nell'albero dei raggi riflessi e rifratti

Il Ray tracing ricorsivo ha il suo ambito naturale ambito di applicazione nel rendere scene molto riflettenti o semitrasparenti. Ma di solito solito solo una piccola parte della scena consiste in oggetti riflettenti o semitrasparenti. Allora si possono abbreviare i calcoli adattando il livello massimo di generazioni nell'albero binario dei raggi riflessi e rifratti generati a un dato pixel al contributo che i raggi riflessi e rifratti danno all'intensità di illuminazione a quel dato pixel [R.Hall, Hybrid Techniques for Rapid Image Synthesis, in Image Rendering Tricks: Course Note 16 for SIGGRAPH 86, Dallas, TX, agosto 1986, T.Whitted & R.Cook, editors]. Più precisamente, per ogni raggio riflesso e rifratto si calcola il fattore di perdita di luminosità causata dai coefficienti dell'equazioni di illuminazione. Ad ogni sua intersezione sui moltiplicano questi fattori percentuali, e quando la luminosità arriva sotto una soglia prefissata ci si ferma. Questo procedimento funziona bene per molte immagini, ma non per tutte, perchè molti raggi il cui contributo alla luminosità è più basso della soglia prefissata possono complessivamente produrre un contributo di illuminazione elevato, che nell'algoritmo di Hall verrebbe omesso.

### Buffer delle sorgenti luminose (light buffer)

C'è una differenza importante tra raggi riflessi o rifratti e raggi d'ombra: i raggi d'ombra sono diretti solo verso pochi oggetti (le sole sorgenti luminose). In *[E.A.Haines, D.P.Greenberg, <u>The Light Buffer: A Shadow-Testing Accelerator</u>, <i>IEEE Computer Graphics & Applications 6(9), settembre 1986, pagg. 6-16]* questo fatto viene usato per aumentare la velocità di elaborazione dei raggi d'ombra. Viene introdotto un *buffer delle sorgenti luminose*, ossia un cubo intorno ad ogni sorgente, con i lati paralleli agli assi delle coordinate della scena (*world coordinates*) e quadrettati con tessere quadrate identiche.

Per ogni tessera si calcola una lista dei poligoni visibili attraverso di essa dalla sorgente di luce, ordinata al crescere della distanza. La lista viene ottenuta attraverso la scansione di tutti gli oggetti rispetto al centro di proiezione dato dalla sorgente di luce. Viene incluso nella lista ogni poligono visibile attraverso la tessera quadrata, anche se la sua sovrapposizione con tale tessera è minoritaria.

A questo punto per ogni raggio d'ombra (diretto verso una data sorgente) si determina per quale tessera passa, e si calcolano le eventuali intersezioni solo con gli oggetti del buffer di luce corrispondenti a quella tessera, nell'ordine di vicinanza alla sorgente.

Se uno degli oggetti copre completamente una tessera, si eliminano dalla lista, per quella tessera, tutti gli oggetti più lontani (i raggi d'ombra generati da questi ultimi sono infatti bloccati dall'oggetto più vicino). Inoltre si omettono dalla lista i lati posteriori di ogni oggetto opaco (*back face culling*).



#### Campionamento di area

Anche il Ray Tracing ricorsivo ha problemi di aliasing, dovuti alla regolarità della griglia di campionamento. Per risolvere questi problemi, in *[T. Whitted, <u>An Improved Illumination</u> <u>Model for Shaded Display</u>, CACM 23(6), giugno 1980, 343-349] viene introdotta una procedura di campionamento di area non pesato (<i>unweighted area sampling*), nella quale si rimpiazza il *raggio proiettore* (raggio uscente dal punto di osservazione) che passa al centro di un pixel con la piramide con vertice nel punto di osservazione e spigoli dati dalle semirette che passano per i quattro angoli del pixel; allo stesso modo si usano tronchi di piramide per i raggi secondari riflessi e trasmessi (ottenuti applicando ai quattro raggi di spigolo le leggi dell'ottica geometrica: riflessione e rifrazione). Questa implementazione risulta troppo costosa in termini di consumo di risorse, ma è la base di partenza di molte varianti efficienti.

#### Tracciamento di coni

Questo metodo, introdotto in *[J. Amanatides, <u>Ray Tracing with Cones</u>, SIGGRAPH 84, in Computer Graphics 18(3), luglio 1984]*, modella i raggi proiettori come coni con origine nel punto di osservazione e di raggio grande abbastanza da coprire il rispettivo pixel. Quando il cono interseca qualche oggetto si calcola quale frazione del cono è stata bloccata da oggetti precedentemente intersecati lungo il percorso. I coni riflessi e rifratti sono determinati sulla base delle leggi dell'ottica: per le superficie curve, si applica un incremento dell'ampiezza dei coni riflessi e rifratti per simulare l'allargamento dei coni dato dalla dispersine laterale dovuta alla curvatura. Il raggio di ombra da un punto di intersezione P ad una sorgente S è un cono con vertice in P e di ampiezza pari alla sezione trasversale di S (vista da P). L'intensità di luce in P si determina sulla base di quale frazione del raggio d'ombra viene bloccata da oggetti incontrati nel suo percorso verso S.

#### Tracciamento di fasci

Si tratta di un metodo a precisione d'oggetto che si applica a scene modellate tramite facce poligonali. I raggi sono sostituiti da piramidi. Si parte con la piramide con vertice nel punto di osservazione e spigoli che passano per i vertici del viewport. Essa viene intersecata con tutti i poligoni della scena, ordinati dal davanti al dietro, utilizzando il metodo di z-buffer (i poligoni vengono proiettati sulla sedzione trasversale della piramide, verso il suo vertice); per ogni poligono intersecato, cioè visibile, si sottrae la corrispondente porzione della piramide mediante un efficiente algoritmo di clipping (di solito si usa l'algoritmo di Weiler [K.Weiler, Polygon Comparison using a Graph Representation, SIGGRAPH 80, in Computer Graphics 14(3), Juglio 1980, 10-18], lo stesso che viene impiegato nel metodo di rimozione di aree nascoste di Atherton-Weiler). Ogni poligono intersecato genera due nuovi fasci piramidali, uno riflesso ed uno rifratto, con vertice al punto di intersezione del poligono con l'asse centrale del fascio piramidale incidente. Si procede ricorsivamente: i calcoli per i nuovi raggi si ricoducono al caso precedente mediante una trasformazione di coordinate che trasporta il punto di intersezione nel punto di osservazione, e l'asse centrale nel nuovo tronco di piramide in quello della piramide originale (cioè nel raggio di proiezione che passa per il centro del viewport). Ogni raggio generato produce un ulteriore mcontributo (speculare o rifratto) che viene aggiunto all'illuminazione del frammento di poligono che l'ha generato. Si usano gli stessi criteri di arresto della ricorrenza già adottati nel Ray Tracing ricorsivo. Per tutti i pixel inclusi in un fascio piramidale dell'ultima generazione l'illuminazione è la stessa, per coerenza: questo dà luogo ad una notevole velocizzazione dei calcoli. Il metodo fu introdotto in [P.S. Heckbert, P. Hanrahan, Beam Tracing Polygonal Objects, SIGGRAPH 84, in Computer Graphics 18(3), luglio 1984].

#### Ray Tracing distribuito stocasticamente

II Ray Tracing distribuito si basa sul sovracampionamento con una distribuzione casuale e non regolare dei campioni, in modo tale che gli errori visivi dovuti all'antialiasing vengano sostituiti da rumore, che ci appare meno fastidioso. L'utilizzo di un campionamento distribuito in modo casuale può portare ad effetti come la profondità di campo, il rendering di sorgenti luminose estese, la riflessione su superficie ruvide e la sfocatura di movimento (motion blur, sbavatura che si verifica fotografando un oggetto in movimento con un tempo basso di esposizione: per questo effetto si deve distribuire il campionamento su diversi fotogrammi consecutivi). Il sovracampionamento non può avvenire semplicemente rimpiazzando la griglia regolare con un numero uguale di raggi distribuiti in maniera casuale con densità uniforme, perché così si formerebbero addensamenti e lacune. E' necessario utilizzare per il campionamento una distribuzione che assicuri che due punti non distino meno di una soglia minima prefissata, come ad esempio la distribuzione di Poisson di minima distanza. La distribuzione di Poisson viene approssimata con un metodo chiamato *jittering* (tremolio), consistente nello spostare ogni punto di una griglia regolare dalla sua posizione originale di una piccola quantità casuale (strettamente inferiore alla metà del lato delle celle della griglia, per evitare addensamenti). Per un campionamento distribuito bidimensionale, ciò significa applicare ad ogni punto di griglia (x,y) due spostamenti dx e dy,uno per la x e l'altro per la y, distribuiti con una varianza così piccola da impedire la sovrapposizione dei punti risultanti. Aire dalla griglia sovracampionata per la correzione dell'aliasing, il valore di intensità di ciascun pixel viene determinato filtrando con maschere centrate nel pixel. Se il jitter è piccolo rispetto all'ampiezza del filtro si possono utilizzare maschere di 4x4 sottopixel, ottenute tramite suddivisioni a partire dalla posizione dei campioni prima dello spostamento.



Quindi, dopo il jitter, i punti del campionamento si collocano ciascuno in una cella quadrata separata.

•×	ו	×.
•*	×.	×.
ו	×	×

Questo tipo di campionamento si dovrebbe chiamare campionamento a celle, ma invece viene di solito indicato con campionamento a strati. Questo nome è dovuto al fatto che usiamo spesso il suo analogo unidimensionale per approssimare integrali di funzioni *f* di una variabile dopo aver ripartito l'intervallo di integrazione con una griglia equispaziata, campionando la funzione *f* su un punto  $x_j$  a caso in ciascun sottointervallo  $I_j$  determinato dalla griglia. In tal modo l'integrale viene approssimato con la somma di aree di strati rettangolari che hanno per base i sottointervalli  $I_j$  ed altezza *f*( $x_j$ ).

*Campionamento basato sull'importanza* Consiste nel campionare in maniera più fitta le aree dell'immagine che contengono i dettagli più importanti.

# RAY TRACING DALLE SORGENTI DI LUCE (*LIGHT TRACING*)

II Ray Tracing ricorsivo non riproduce correttamente scene che contengono specchi o lenti, perché i raggi d'ombra vengono inviati nella direzione della sorgente luminosa e quindi non seguono il percorso corretto della luce, che invece subisce rifrazioni e riflessioni. Quindi l'algoritmo rende una scena che include uno specchio in modo tale che la luce riflessa dallo specchio non proietti ombre; inoltre le ombre di oggetti semitrasparenti non danno luogo a rifrazione. C'è un esempio di errore ancora più grave, che si verifica nelle scene con lenti che fanno convergere la luce su una superficie, creando zone ristrette di alta luminosità chiamate caustiche (un esempio sono le chiazze di luce ed ombra create dalle onde sul fondo del mare vicino a riva). Poiché le caustiche sono molto luminose e si presentano in aree ristrette, esse danno luogo ad un elevato gradiente di luminosità, per visualizzarle correttamente è necessario che queste aree siano raggiunte da molti raggi proiettori, il che generalmente non è vero (si sub-pixel). procedimento caustiche di dimensioni Un pensi а si sottocampionamento adattivo può non funzionare: ad esempio, se la caustica è di dimensioni inferiori ad un pixel, è probabile che il procedimento adattivo non se ne accorga e non si attui.

Per correggere questi problemi si effettua una variante del Ray Tracing, questa volta a partire dalle sorgenti luminose verso l'osservatore. Però non è opportuno implementare l'intero metodo a rovescio, cioè a partire dalle sorgenti di luce, perchè non si sa a priori (prima di tracciarli) quali raggi passeranno per i pixel del piano di visuale, e si dovrebbero quindi tracciare moltissimi raggi di prova alla fine inutili. Però il tracciamento a ritroso serve per completare l'informazione ottenuta dal tracciamento normale. Il metodo del tracciamento dei fasci può essere implentato in questo spirito, osservando che le generazioni successive dell'albero dei fasci di luce rappresentano frammenti di poligoni illuminati in modo indiretto. Questi frammenti si possono aggiungere al database dei poligoni iniziali della scena, e così si riesce a rendere l'illuminazione riflessa specularmente o rifratta. Un altro algoritmo, introdotto in [J.Arvo, Backward Ray Tracing, in Developments in Ray Tracing Course Notes 12, Dallas, TX, agosto 1986, A.H. Barr, editor, traccia dei raggi nell'ambiente a partire da ogni sorgente luminosa; ogni raggio possiede una quota iniziale di energia. Ad ogni intersezione con un oggetto che diffonde la luce, il raggio deposita una piccola quantità di energia che viene accumulata in una griglia rettangolare regolare applicata all'oggetto. I vertici della griglia fungono da contatori, cioè accumulatori dell'energia ricevuta. Ogni contributo viene ripartito in maniera lineare tra i quattro contatori che circondano la cella della griglia colpita dal raggio. Successivamente si esegue il Ray Tracing convenzionale, utilizzando per determinare la riflessione diffusa sia i valori precedentemente accumulati, sia le intensità di luce delle di sorgenti visibili determinate tramite Ray Tracing. Così si risolve il problema legato al fatto che, dato un punto di intersezione di un raggio proveniente dall'osservatore con un oggetto della scena, è improbabile sparare a caso un raggio da una sorgente di luce che colpisca prorpio quel punto (di solito questo raggio non precede direttamente verso il punto di interezione, per via delle rifrazioni e riflessioni): si sparano raggi dalla sorgente, e per ciascuno di essi si distribuiscono i contributi su una griglia di contatori vicini al punto effettivo di intersezione, accumulando i risultati. Purtroppo, se un raggio luminoso colpisce la parte invisibile di un oggetto in vicinanza del bordo di visibilità, quel raggio può influenzare un contatore che interessa la parte visibile, e quindi modificare la luminosità della parte visibile: l'algoritmo non gestisce questo errore, peraltro di solito lieve.

# ESERCIZI SULLE STIME DEL TEMPO DI RENDERING DEL RAY TRACING RICORSIVO ABBINATO A MAPPE DI RIFLESSIONE E DI TESSITURA

Nel rendering di un fotogramma con risoluzione di 1 milione di pixel, sia T0 il tempo necessario per tracciare un raggio ed intersecarlo con tutti i poligoni della scena, tr il tempo necessario per calcolare il vettore riflesso o rifratto o il versore della direzione della luce (supponiamo ci sia una sola sorgente di luce), tl il tempo necessario per calcolare l'equazione dell'illuminazione di Phong ad un punto della scena, Tg il tempo necessario per interpolare i valori di illuminazione dai vertici dei poligoni ai punti colpiti dai raggi proiettori, e t<sub>0</sub> il tempo necessario per la lettura di un valore della mappa di tessitura. Supponiamo che il rapporto di scala di questi tempi sia t<sub>0</sub> : tr : tl : Tg : Tt: T0 = 1/10 :1: 1 : 10 : 100 : 1000.

a. Supponiamo dapprima che la scena non abbia poligoni semitrasparenti, e che il numero medio delle generazioni di nuovi raggi nel rendering con il Ray Tracing ricorsivo sia 3 (tre nuove generazioni dopo il primo hit di ogni raggio proiettore). Quanto tempo è necessario per il rendering, come funzione di t<sub>0</sub>?

b. Se invece tutto il calcolo delle riflessioni speculari è sostituito dall'uso della mappa di riflessione, quanto diventa il tempo di calcolo? Qual è il risparmio di tempo?

c. Supponiamo ora che la scena abbia anche poligoni semitrasparenti, quindi ad ogni generazione deve essere generato anche un raggio rifratto. Come cambia il tempo di rendering, come funzione di  $t_0$ ?

d. Poiché l'uso della mappa di riflessione consiste nell'usare il versore riflesso come puntatore al buffer della mappa, essa si può usare anche per il calcolo approssimato dei contributi di rifrazione (in tal caso, oltre all'errore di parallasse, c'è anche un errore di deviazione nei casi in cui il raggio rifratto tracciato esce dal materiale semitrasparente e dovrebbe deviare rifrangendosi di nuovo, mentre ovviamente l'uso del versore rifratto come puntatore non tiene conto di questa ulteriore deviazione). Supponiamo che, in aggiunta ai poligoni riflettenti, ci siano anche poligoni semitrasparenti. Come cambiano le risposte precedenti?

e. Se nei punti precedenti, per evitare salti netti del colore agli spigoli, la lettura dell'illuminazione di Phong è attuata tramite interpolazione di Gouraud, come cambiano le risposte?

f. Supponiamo che una frazione  $\alpha$  dei raggi proiettori sia gestita tramite ray tracing ricorsivo a tre generazioni, una frazione  $\beta$  tramite mappa di riflessione ed il resto, 1- $\alpha$ - $\beta$ , tramite mappa di tessitura (ossia applicando una tessitura e poi andando a leggere il valore dell'equazione dell'illuminazione di Phong al punto colpito dal raggio proiettore, senza alcuna ulteriore generazione di raggi o lettura della mappa di riflessione). Cosa diventano le risposte precedenti, come funzione di t<sub>0</sub>,  $\alpha \in \beta$ ?

g. Supponiamo  $t_0 = 10^{-9}$  secondi,  $\alpha = 1/3 = \beta$ . Rispetto al caso a, qual è il risparmio di tempo, in percentuale, nei casi b, c, d ed e, rispettivamente?

#### Svolgimento.

a. Ci sono  $10^6$  raggi proiettori. Per ciascuno occorre determinarne la direzione (tempo necessario tr), tracciarlo ed intersecarlo con tutti i poligoni della scena e leggere il valore dell'equazione di Phong al punto di intersezione più vicino (tempo richiesto T0+tl), e bisogna anche tracciare un raggio d'ombra calcolandone le intersezioni con tutti i poligoni della scena (il tempo necessario per questo è B=T0+tr). Quindi fino a questa fase per ciascun proiettore originario il tempo è A+B, dove A=T0+tl+tr e B=T0+tr: il totale finora fa A+B=2T0+tl+2tr. Ora bisogna calcolare il versore riflesso, tracciare il raggio riflesso, calcolare le intersezioni e l'equazione di Phong: il tempo necessario è di nuovo A=T0+tl+tr, ma c'è anche un raggio d'ombra, per il quale non si deve calcolare il contributo di Phong però si deve eseguire il tracciamento e calcolare le intersezioni, e per esso il tempo è B=T0+tr. Questo si ripete per altre due generazioni, e porta ad un tempo 4A+4B al termine della terza generazione. Ovviamente questo è il tempo per ciascun proiettore originale: poi bisogna moltiplicare per  $10^6$ .

b. Ora non occorre più generare raggi riflessi. Il tempo per pixel necessario per il rendering è quello della prima fase (tracciamento di proiettori), più quello per la lettura della mappa di riflessione: totale A+  $t_0$ . Prima però bisogna fare una volta per tutte il calcolo della mappa: il tempo necessario per questo è circa lo stesso che per la prima fase, ossia A moltiplicato per il numero di raggi che usiamo per la mappa di riflessione (ossia la sua risoluzione), il quale certamente è molto minore del numero di pixel del display (dobbiamo dopotutto rendere i riflessi di aree speculari che tipicamente sono piccole: se sono grandi sappiamo che l'errore di parallasse inerente alla mappa di riflessione rende il procedimento inopportuno per eccesso di distorsione). Chiamiamo  $\Lambda$  la risoluzione della mappa di riflessione: riassumendo, essa consiste del numero di direzioni in cui campioniamo un grande emisfero per determinarvi i valori della mappa di riflessione. Tipicamente  $\Lambda$  è dell'ordine di  $10^2$  o  $10^3$ : diciamo  $10^3$ , nel caso peggiore. Totale,  $10^3A + 10^6(A + t_0)$ , ossia  $10^6((1+10^{-3})A+t_0)$ .

c. La fase iniziale, che riguarda i raggi proiettori originali, è come prima nel punto a: il tempo è A+B=2T0+tl+2tr. Ora però ad ogni generazione i raggi si raddoppiano: bisogna calcolare i versori riflesso *e rifratto*, rifare il tracciamento e le intersezioni e calcolare l'equazione di Phong (tempo A=tr+T0 +tl) per per ciascuno dei due raggi generati (riflesso e rifratto). Inoltre quando i due raggi colpiscono il poligono della scena piu' vicino, per ciascuno di essi bisogna anche tracciare un raggio d'ombra ed intersecarlo con tutti i poligoni della scena: tempo B=T0+tr. Quindi dopo la prima generazione il tempo impiegato è (A+B)+(2A+2B)=3A+3B.

Per ciascuno dei due raggi della prima generazione occorre ripetere il procedimento: a questo punto ci sono 4 raggi nella genealogia (riflesso ed rifratto di ciascuno dei raggi riflesso e rifratto della prima generazione) e 2 raggi d'ombra. Pertanto dopo la seconda generazione il tempo passato è (A+B)+(2A+2B)+(4A+4B)=7A+7B. Dopo la terza generazione (altri 8 raggi) arriviamo al tempo A+(2A+2B)+(4A+2B)+(8A+8B) = 15A+15B. Tutto questo da moltiplicare per  $10^6$ .

d. Ora si deve sostituire il termine tl con il tempo di interpolazione Tg: se scriviamo A'=T0+tg+tr, la risposta diventa 4A'+4B con solo il raggio riflesso, e 15A'+15B con raggi riflesso e rifratto. Tutto questo da moltiplicare per  $10^6$ .

e. Se si usa solo la mappa di tessitura, il tempo necessario è quasi lo stesso che con la mappa di riflessione, però non occorre calcolare una volta per tutte la mappa di riflessione (tempo A), bensì caricare la mappa di tessitura, che richiede un tempo molto più breve, Tt. Quindi il tempo diventa A+ t<sub>0</sub>+Tt. Siccome abbiamo proporzioni  $\alpha$ ,  $\beta \in 1-\alpha-\beta$ , i tempi per pixel (che poi devono tutti essere moltiplicati per 10<sup>6</sup> per avere il tempo complessivo del rendering) diventano:

senza interpolazione di Gouraud:

al punto a:  $\alpha (4A+4B) + \beta ((1+10^{-3})A+t_0) + (1-\alpha-\beta) (A+t_0+Tt)$ al punto d:  $\alpha (15A+15B) + \beta ((1+10^{-3})A+t_0) + (1-\alpha-\beta) (A+t_0+Tt)$ se si usa l'interpolazione di Gouraud: al punto a:  $\alpha (4A'+4B) + \beta (A'+10^{-3}A+t_0) + (1-\alpha-\beta) (A'+t_0+Tt)$ 

al punto d:  $\alpha (15A'+15B) + \beta (A'+10^{-3}A+t_0) + (1-\alpha-\beta) (A'+t_0+Tt)$ 

(si noti che l'ombreggiatura di Gouraud si applica solo al contributo di illuminazione di Phong, che dipende dalla geometria dei versori, e quindi anche alla mappa di tessitura, ma non al calcolo iniziale della mappa di riflessione, che si basa solo su una proiezione radiale non interpolata).

f. Con  $t_0 = 10^{-9}$  secondi, si ha A=0.000001002, B=0.000001001, Tt=0.0000001, A'=0.000001011 secondi. Bisogna moltiplicare per 10<sup>6</sup> e poi applicare le proporzioni 1/3, 1/3, 1/3. Il tempo al punto a (solo ray tracing ricorsivo senza rifrazione) diventa  $10^{6}(4A+4B)=4.008+4.004=8.012$  secondi. AL punto d (ray tracing ricorsivo con rifrazione) si ottiene 15.0165+15.015=30.0315 secondi. Usando anche l'interpolazione di Gouraud si trova invece circa 4.16+4.004=8.12 nel caso a. e 15.165+15.015=30.18 nel caso b.

Per la sola mappa di riflessione il tempo è  $10^{6}(1.001A + t_{0})=1.004$  con o senza Gouraud (con Gouraud la differenza è sulla quinta cifra: 1.004011).

Per la mappa di tessitura si ottiene 1.003 secondi senza Gouraud e 1.012 secondi con Gouraud.

Quindi la mappa di riflessione offre una riduzione dei tempi che diventano circa  $1/8.012 \sim 12.48\%$  senza Gouraud,  $(1/8.12 \sim 12.32\%)$  con Gouraud nel caso non ci siano materiali semitrasparenti, e  $1/30.03 \sim 3.33\%$  senza Gouraud ( $1/30.18 \sim 3.31\%$  con Gouraud) se c'è trasparenza. La mappa di

tessitura ha tempi simili. Nel caso di proporzioni 1/3, 1/3, 1/3 si ottiene una riduzione del tempo che diventa il 41.7% circa nel caso senza rifrazione, ed il 35.3% nel caso di rifrazione.

Ovviamente queste stime sono per eccesso: stiamo supponendo che per ogni ogni raggio che tocca la scena venga generato un raggio riflesso ed uno rifratto. In una scena tipica sono non molti i poligoni riflettenti, e pochi quelli rifrangenti. Lasciamo come esercizio al lettore una stima più accurata, assumendo che solo una percentuale fissata di raggi si riflettano effettivamente, ed un'altra percentuale, piccola, individui i raggi che si rifrangono: la stima deve dipendere esplicitamente da queste due percentuali. RADIOSITÀ

# INTRODUZIONE ALLA RADIOSITÀ

Questo modello per la luce diffusa, introdotto in [*C.M.Goral, K.E.Torrance, D.P.Greenberg e B.Bataille, Modeling the Interaction of Light between Diffuse Surfaces, SIGGRAPH 84, in Computer Graphics 18(3), luglio 1984, 213-222], sposta l'attenzione dallo studio delle riflessioni dirette ad una analisi accurata dell'effetto cumulativo delle interriflessioni tra gli oggetti della scena.* 

I primi algoritmi in questo campo si occupavano della radiazione del calore dentro motori ed altri ambienti chiusi, e da questo ambito furono poi trasportati allo studio della diffusione della luce. Si calcola quanta parte dell'energia emessa o riflessa da ogni superficie raggiunge ogni altra superficie, venendone ulteriormente riflessa o assorbita. Una parte della porzione riflessa raggiunge nuovamente la superficie originale, e quindi ne modifica la luminosità. Questo porta ad una successione di correzioni successive, che si traducono nella risoluzione con metodi iterativi di un appropriato sistema lineare. La soluzione dà la luminosità di ogni superficie della scena in conseguenza dell'illuminazione che la raggiunge direttamente dalle sorgenti di luce ed indirettamente dalla diffusione da parte delle altre superficie. La potenza luminosa totale di una superficie (emessa e/o riflessa) si chiama radiosità.

Il metodo di calcolo suddivide le superficie in pezzi più piccoli di luminosità uniforme, chiamati elementi (*patch*), e risolve per iterazione il sistema lineare di cui sopra per trovare la luminosità dei vertici di ciascun elemento generata dalla luce diretta e da quella diffusa dalle altre superficie. Il procedimento iterativo aggiunge a ciascun elemento la luminosità diffusa dalle successive. Quindi nei primi passi iterativi la luminosità globale risulta ridotta, perché si perde la luminosità dovuta ai contributi degli elementi ancora non considerate. Per rendere più rapida la convergenza dell'iterazione, viene quindi eseguito un procedimento di correzione della luminosità globale per recuperare la perdita iniziale.

# L' EQUAZIONE DELLA RADIOSITÀ

Le superficie considerate nel metodo di radiosità possono emettere luce propria, a differenza di quelle considerate nel ray tracing. In particolare, le sorgenti di luce in questo metodo diventano superficie (il metodo è ideale, in particolare, per trattare sorgenti di luce estese), ma ciascuna superficie, oltre ad emettere luce propria, diffonde quella che le proviene dalle altre.

Quindi ciascuna superficie della scena è modellata come una sorgente luminosa estesa, avente un'area finita. Suddividiamo l'ambiente in un numero finito di sottosuperficie sufficientemente piccole, che chiameremo elementi, che emettono e riflettono luce in modo uniforme sull'intera area. Se consideriamo ogni elemento come una sorgente di luce che, oltre ad emettere la propria luce, diffonde quella proveniente dalle altre sorgenti (ma non la riflette specularmente: è una superficie opaca), allora, per ogni i = 1,..., n, abbiamo la seguente equazione:

$$b_i = e_i + \rho_i \sum_{j=1,\dots,n} b_j F_{j-i} A_j / A_i$$

Qui  $b_i e b_j$  sono le radiosità - cioè le potenze totali per unità di area ( = tasso di energia emessa per unità di tempo e di area) - uscenti dagli elementi i e j,  $A_i$  ed  $A_j$  sono le aree degli elementi i e j,  $e_i$  la potenza totale per unità di area emessa dall'elemento i,  $p_i$  la riflettività dell'elemento i e  $F_{j-i}$  il fattore di forma dell'elemento i rispetto all'elemento j, che misura quale percentuale della potenza emessa dall'elemento j arriva all'elemento i. Il fattore di forma misura quindi quanto è grande l'angolo solido coperto dalla porzione visibile dell'elemento i quando si guarda la scena dal centro dell'elemento j (la frazione dell'angolo solido coperto dalla porzione visibile dell'elemento i); sarebbe più accurato fare la media degli angoli solidi coperti dall'elemento i al variare del punto di osservazione sull'elemento j, ma questo porterebbe ad una mole di calcoli eccessiva. Il j-simo addendo della sommatoria rappresenta l'energia per unità di tempo e di area che l'elemento i riceve da quello j. Questa energia, moltiplicata per il coefficiente di riflettività dell'elemento i, è la frazione proveniente dall'elemento i.

Riassumendo, questa equazione indica semplicemente la relazione esistente tra l'energia emessa per unità di tempo dall'elemento i e l'energia per unita di tempo entrante in questo elemento, vista come somma della componente di luce propria e quella di luce diffusa proveniente dagli altri elementi.

E' importante notare che il 100% dell'energia emessa da ciascun elemento deve arrivare agli altri elementi (considerando anche lo sfondo come uno o più elementi). In altre parole si ha la seguente equazione della conservazione dell'energia (ovvero dell'angolo solido totale):

$$\Sigma_j F_{i-j} = 1$$

Come accennato più sopra, quest'ultima quantità viene determinata moltiplicando per il coefficiente di riflessione dell'elemento i la somma della luce incidente proveniente dagli altri elementi, cioè la somma al variare di j della luce che lascia un'unità di area dell'elemento j e che raggiunge l'elemento i.

Come vedremo, vale la seguente relazione di reciprocità tra i fattori di forma (reversibilità dei percorsi ottici):

$$A_i F_{i-j} = A_j F_{j-i}$$

Essenzialmente questa relazione esprime, in termini della luce inviata da un elemento all'altra, la reversibilità dei percorsi ottici: se un punto di un elemento è visibile da un punto dell'altro, allora è vero anche il viceversa.

Pertanto possiamo riscrivere l'equazione precedente così:

$$b_i = e_i + \rho_i \sum_{j=1,\dots,n} b_j F_{i\text{-}j}$$

Riordinando i termini e considerando tutti i possibili valori di i arriviamo al seguente sistema lineare:

$$\begin{pmatrix} 1 - \rho_{1}F_{1-1} & -\rho_{1}F_{1-2} & \cdots & -\rho_{1}F_{1-n} \\ -\rho_{2}F_{2-1} & 1 - \rho_{2}F_{2-2} & \cdots & -\rho_{2}F_{2-n} \\ \cdots & \cdots & \cdots \\ -\rho_{n}F_{n}F_{n-1} & -\rho_{n}F_{n-2} & \cdots & 1 - \rho F_{n-n} \end{pmatrix} \begin{pmatrix} b_{1} \\ b_{2} \\ \vdots \\ b_{n} \end{pmatrix} = \begin{pmatrix} e_{1} \\ e_{2} \\ \vdots \\ e_{n} \end{pmatrix}$$

Il sistema è quindi il seguente:

dove **b** ed **e** sono rispettivamente i vettori delle radiosità e delle potenze emesse, ed M è la matrice

$$M_{ij} = \delta_{ij} - \rho_i F_{i\text{-}j}$$

(come sempre, il simbolo di Kronecker  $\delta_{ij}$  vale 1 se i=j e 0 altrimenti). E' importante rendersi conto per i termini diagonali possiamo supporre

$$M_{ii} > 0$$

dal momento che

 $\rho_{i} < 1$ 

(la radiosità serve a studiare scene diffusive, non perfettamente speculari!), e

$$F_{i-i} < 1$$

a meno che l'elemento i-simo non sia l'unico elemento della scena incluso lo sfondo, nel qual caso la scena è banale: l'unico elemento ha illuminazione costante, e non occorrono metodi numerici per il rendering.

Questo sistema di equazioni, risolvibile usando il metodo iterativo di <u>Gauss-Seidel</u>, porta ad ottenere i valori di radiosità per ogni elemento, i quali, calcolati in funzione della lunghezza d'onda, possono essere interpretati come la distribuzione spettrale dell'intensità dell'elemento nell'immagine, e sono utilizzabili nell'algoritmo di determinazione delle superfici visibili.

Nell'articolo [M.F.Cohen e D.P.Greenberg, <u>The Hemi-Cube: A Radiosity Solution</u> for Complex Environments, SIGGRAPH 85, in Computer Graphics 19(3), luglio 1985, pagg. 21-40] viene sviluppato un metodo per determinare la radiosità di un vertice della scena poligonale, utilizzabile poi nello shading interpolato (secondo i metodi di <u>Gouraud</u> o <u>Phong</u>, ma qui di solito si usa il metodo di Gouraud perché stiamo studiando un problema di luce diffusa, ed il metodo di Phong era mirato a trattare effetti anche speculari): ad un vertice interno alla superficie viene assegnata la media delle radiosità degli elementi che condividono quel vertice. Invece, se il vertice appartiene ad un lato, la radiosità di quel vertice viene determinate in due passi. Prima si trova quella del vertice Interno più vicino al vertice dato e quelle degli elementi che condividono il vertice interno più vicino e la radiosità del vertice dato, che è la nostra incognita. Così si arriva ad una equazione lineare, risolvendo la quale otteniamo la quantità cercata.

# **CALCOLO DEI FATTORI DI FORMA**

L'operazione più importante e computazionalmente più gravosa del metodo di radiosità è il calcolo dei fattori di forma, i quali misurano quale frazione dell'energia emessa da un elemento ne raggiunge un'altra. Sono quindi una misura geometrica della visibilità di una superficie da un'altra.

Nell'articolo [M.F.Cohen e D.P.Greenberg, The Hemi-Cube: A Radiosity Solution for Complex Environments, SIGGRAPH 85, in Computer Graphics 19(3), luglio 1985, pagg. 21-40] viene sviluppato un algoritmo a precisione di immagine per la determinazione delle superficie visibili per approssimare i fattori di forma. Date due elementi, la geometria che porta a determinare il fattore di forma da un elemento infinitesimo di area dA<sub>i</sub> della prima ad un elemento infinitesimo dA<sub>j</sub> della seconda (che chiamiamo  $F_{di-j}$ ) è illustrata nella figura seguente. Occorre moltiplicare le aree per i due rispettivi coseni per proiettarle perpendicolarmente al segmento che le congiunge (in generale sono disposte obliquamente rispetto ad esso, ed in tal modo coprono un angolo solido ridotto di un fattore pari a questo coseno).



Qui  $\theta_i$  è l'angolo che il raggio forma con la normale alla superficie A<sub>i</sub>,  $\theta_j$  è l'angolo che il raggio forma con la normale alla superficie A<sub>j</sub>, H<sub>ij</sub> assume i valori 0 o 1 a seconda che dA<sub>j</sub> sia visibile o meno da dA<sub>i</sub>, e r è la lunghezza del raggio. Partendo da questo risultato possiamo ottenere il fattore di forma da un elemento infinitesimo di area dA<sub>i</sub> all'intero elemento j-esimo di area A<sub>j</sub> integrando su tutta l'area dell'elemento j-esimo.

In tal modo otteniamo che F<sub>di-j</sub> è dato da

$$F_{di-j} = \int_{A_j} \frac{\cos \theta_i \cos \theta_j}{\pi r^2} H_{ij} dA_j$$

Ora, integrando sull'elemento i-esimo e prendendo la media integrale (cioè dividendo l'integrale per l'area A<sub>i</sub>), otteniamo il fattore di forma dall'elemento i all'elemento j:

$$F_{i-j} = \frac{1}{A_i} \int_{A_i} \int_{A_i} \frac{\cos \theta_i \cos \theta_j}{\pi r^2} H_{ij} dA_j dA_i$$

Si osservi che da questa formula segue la relazione di reciprocità accennata precedentemente:

$$A_i F_{i-j} = A_j F_{j-i}$$

Se assumiamo che il punto centrale di un elemento rappresenti con buona precisione la luminosità di tutti gli altri suoi punti, possiamo approssimare  $F_{i-j}$  con  $F_{di-j}$ , calcolato per dA<sub>i</sub> nel centro dall'elemento i-esimo.

Sono disponibili metodi numerici più semplici per calcolare  $F_{di-j}$ . Nel libro di *R.Siegel e J.Howell, Thermal Radiation Heat Transfer, 2<sup>nd</sup> edition, Hemisphere, Washington, DC, 1981*) si attribuisce a Nusselt l'aver notato che la quantità  $F_{ik}$  si possa calcolare mediante due proiezioni successive: innanzitutto bisogna proiettare quelle parti di  $A_k$  (elemento k-esimo) che sono visibili dal pezzo  $dA_i$ dell'elemento  $A_i$  (elemento i-esimo) su una semisfera unitaria, centrata in  $dA_i$ : si osservi che il fattore di forma di questa proiezione (ossia l'angolo solido da essa coperto) coincide con quello dell'elemento  $A_k$ . Successivamente quest'area sulla superficie della semisfera viene proiettata ortograficamente sul disco unitario che costituisce la base della semisfera. La percentuale dell'area di questo disco coperta dalla superficie proiettata è il fattore di forma (la proiezione sulla sfera unitaria è data dalla moltiplicazione per cos  $\theta_k/r^2$ , la proiezione sul disco unitario dalla moltiplicazione per cos  $\theta_i$ .



Nell'articolo [M.F.Cohen e D.P.Greenberg, The Hemi-Cube: A Radiosity Solution for Complex Environments, SIGGRAPH 85, in Computer Graphics 19(3), luglio 1985, pagg. 21-40] viene sviluppato un algoritmo a precisione di immagine (infatti basato su una decomposizione in una griglia di pixel, come lo z-buffer) nel quale la proiezione avviene sulla parte superiore di un semicubo, centrato in dAi. Ogni faccia viene suddivisa con una griglia regolare. Tutte gli altri elementi vengono proiettate verso il centro del cubo e quindi si calcola quali celle la proiezione copre sulle cinque facce superiori. Esattamente come nel caso dell'emisfero, il fattore di forma dell'elemento originale A<sub>k</sub> coincide con quello della sua proiezione sull'emicubo, e quindi tanto vale rimpiazzare la prima con la seconda (per la guale i vettori normali sono i vettori diretti perpendicolarmente alla faccia dell'emicubo su cui cade la proiezione). Per ciascuna cella di queste facce si determina quale sia l'elemento A<sub>k</sub> più vicina (nella direzione di visuale dal centro del cubo) mediante l'algoritmo di z-buffer. L'elemento piu vicina, per una data cella, è l'unico elemento che riflette energia luminosa su dAi attraverso quella cella, poiché le altre sono schermate dalla prima.



Ad ogni cella viene assegnato un fattore di forma precalcolato del tipo

$$\Delta F_{\rm p} = \frac{\cos\theta_{\rm i} \,\cos\theta_{\rm p}}{\pi r^2} \,\Delta A$$

dove  $\theta_p$  è l'angolo tra la normale alla cella p e il vettore r che congiunge il centro di dA<sub>i</sub> (cioè il centro del cubo) col centro di p, e  $\Delta A$  è l'area della cella. In un sistema di riferimento per il semicubo con l'origine nel suo centro, per le celle della faccia superiore otteniamo:

$$r = \sqrt{1 + x_p^2 + y_p^2}$$
$$\cos \theta_i = \cos \theta_p = \frac{1}{r}$$

dove  $(x_p, y_p)$  sono le coordinate del centro della cella p. La geometria è illustrata nella seguente figura:



Ora l'equazione precedente, per una cella appartenente alla faccia superiore, ci dà:

$$\Delta F_{p} = \frac{1}{\pi \left(1 + x_{p}^{2} + y_{p}^{2}\right)^{2}} \Delta A$$

Invece, la geometria delle celle in una faccia laterale del semicubo che è perpendicolare, diciamo, all'asse è illustrata nella figura seguente:



Analogamente, per queste celle si ha:

$$\Delta F_{p} = \frac{Z_{p}}{\pi \left(1 + x_{p}^{2} + y_{p}^{2}\right)^{2}} \Delta A$$

Qui stiamo considerando le facce laterali, ed quindi per due di esse  $x_p$  vale rispettivamente 1 o -1, e per le altre due  $y_p$  vale rispettivamente 1 o -1.

Ora il fattore di forma  $F_{di-k}$  tra l'elemento i-esimo e l'elemento k-esimo può essere approssimato con la somma di tutti i fattori di forma  $\Delta F_p$  associati ad ogni cella p del semicubo centrato sull'elemento i-esimo coperta dalle proiezioni dell'elemento k-esimo. Si noti che, poichè questi metodi usano la suddivisione in celle delle facce e lo z-buffer, essi sono algoritmi a precisione di immagine, e quindi soggetti al fenomeno dell'aliasing.

## ESERCIZI SUL CALCOLO ANALITICO DEI FATTORI DI FORMA

**Esercizio 1.** Una scena consiste di un cono retto di angolo al vertice pari a  $\pi/4$ , ed altezza h. Quindi la base del cono è un disco, che chiamiamo B. Supponiamo che B sia un elemento: quanto vale il suo fattore di forma visto dal vertice del cono? (Suggerimento: si usi il metodo di Nusselt).

Svolgimento. Per prima cosa proiettiamo il disco sulla superficie sferica di raggio 1, con proiezione radiale verso il centro di questa sfera, che è il punto da cui si vuol calcolare il fattore di forma. Per inciso, si osservi che il fattore di forma misura solo l'angolo solido sotteso, e quindi sarebbe invariato se la base del cono, invece di un disco piano, fosse una calotta sferica, ad esempio quella della sfera di raggio h, o anche di raggio 1: la prima fase del procedimento di Nusselt, la proiezione radiale, non è altro che questa semplice osservazione.

A questo punto abbiamo ottenuto una calotta C di ampiezza angolare  $\pi/4$ , disposta, diciamo, intorno al polo Nord (x=y=0, z=1) della superficie sferica di raggio 1 con centro l'origine. Il metodo di Nusselt consiste nell'osservare che il fattore di forma richiesto è esattamente la percentuale dell'area del disco di base della sfera (ossia il suo disco equatoriale), che vale  $\pi$ , data dall'area della proiezione verticale D della calotta C sul piano di base z=0. La deviazione angolare massima della calotta rispetto alla direzione verticale è  $\pi/8$ . Quindi la calotta si proietta sul disco D nel piano z=0 il cui raggio è il raggio della sfera per sin( $\pi/8$ ), ossia sin( $\pi/8$ ). L'area di D è  $\pi$  sin<sup>2</sup>( $\pi/8$ ), e quindi il fattore di forma vale  $\pi$  sin<sup>2</sup>( $\pi/8$ )/ $\pi$  = sin<sup>2</sup>( $\pi/8$ ).

*Esercizio 2.* Per il cono dell'esercizio precedente, ricalcolare il fattore di forma della sua base B visto dal vertice del cono tramite la formula integrale che lo definisce.

Svolgimento. La formula che dà il fattore di forma è  $F = \int_{C} \frac{\cos\theta_i \cos\theta_j}{\pi h^2} H_{ij} dC$ , dove

gli angoli  $\theta_i$  e  $\theta_j$  sono come nella figura già vista precedentemente, che riportiamo qui,



ed il fattore di occlusione H<sub>ij</sub> vale 1 se il punto j è visibile dal vertice da cui vogliamo calcolare il fattore di forma (nella figura questo vertice lo chiamiamo i). Osserviamo che il fattore di occlusione vale sempre 1 perché non ci sono altri elementi che occludono la base del cono quando vista dal vertice (il cono è convesso!). Abbiamo già osservato che il fattore di forma F rimane uguale se si rimpiazza il disco di base B con la calotta sferica C di raggio h incernierata sul cono. Inoltre sui punti di C l'angolo  $\theta_j$  vale sempre 0, perché la calotta sferica ha per versore normale esattamente il versore radiale. Quindi chiamiamo più semplicemente  $\theta$  l'altro angolo  $\theta_i$ , e notiamo che la deviazione angolare (azimuth)  $\theta$  verifica  $0 \le \theta \le \pi/8 \le /8$ . Allora l'integrale diventa

$$F = \int_{C} \frac{\cos\theta}{\pi h^2} dC = \int_{0}^{2\pi} \int_{0}^{\frac{\pi}{8}} \frac{\cos\theta}{\pi h^2} h^2 \sin\theta \, d\theta \, d\phi = \int_{0}^{\frac{\pi}{8}} \sin 2\theta \, d\theta = \frac{1}{2} \int_{0}^{\frac{\pi}{4}} \sin\theta \, d\theta$$
$$= \frac{1}{2} \left( 1 - \cos\frac{\pi}{4} \right) = \frac{1}{2} \left( 1 - \cos^2\frac{\pi}{8} + \sin^2\frac{\pi}{8} \right) = \sin^2\frac{\pi}{8}$$

Qui il fattore sin  $\theta$  dentro l'integrale di superficie è il fattore di dilatazione delle aree della trasformazione  $\sigma$  da coordinate sferiche  $\theta$ ,  $\phi$  (angoli di Eulero) a coordinate cartesiane x, y, z, a suo tempo calcolato:

$$\left\|\frac{\partial\sigma}{\partial\theta} \times \frac{\partial\sigma}{\partial\varphi}\right\| = r\sin\theta$$

Se invece si desidera svolgere il calcolo direttamente per il disco piano B, basta osservare che la normale al disco B è il versore verticale (diretto verso il basso, se il vertice del cono è all'origine ed il disco B ad altezza h>0). Perciò ora  $\theta_i = \theta_j$ : chiamiamo questo angolo  $\theta$ . Il raggio  $\rho_B$  di B è proporzionale al raggio  $\rho_E$  della
calotta sferica E che B sottende. A*ttenzione*: nella parte precedente di questo esercizio la calotta non era E ma C, ossia non era quella sottesa da B, bensì quella tangente a B al suo centro, che ha raggio  $h < \rho_E$ , ed infatti tale che

$$h/\rho_{\rm E} = \cos(\pi/8) \tag{1}$$

Quindi ora il denominatore nell'integrando del fattore di forma è  $\pi r^2$  invece di  $\pi h^2$ : questa volta r non è costante ma dipende da x e y. Nell'Esercizio 1 abbiamo visto che  $\rho_B = \rho_E \sin(\pi/8)$ . Eliminiamo  $\rho_E$  sostituendo (1) in questa identità: troviamo la relazione, peraltro geometricamente evidente,  $\rho_B = h \ tg(\pi/8)$ . Per la stessa ragione, un punto di B che dista  $\rho$  dal centro di B e r dall'origine ha una deviazione angolare  $\theta$  rispetto alla verticale tale che  $\rho = h \ tg \ \theta = r \ sin \ \theta$ , e h = r cos  $\theta$  (come in (1)). Pertanto, ponendo u =  $\rho/h$ , otteniamo u = tg  $\theta$ , cos<sup>2</sup>  $\theta$  =  $h^2/r^2$ , e, per il teorema di Pitagora,  $r^2 = \rho^2 + h^2$ . Ne segue:

$$F = \int_{B} \frac{\cos^{2}\theta}{\pi r^{2}} d\sigma = \int_{0}^{2\pi} \int_{0}^{htg\frac{\pi}{8}} \frac{h^{2}}{\pi r^{4}} \rho d\rho d\phi = 2 \int_{0}^{htg\frac{\pi}{8}} \frac{h^{2}}{\left(\rho^{2} + h^{2}\right)^{2}} \rho d\rho$$
$$= 2 \int_{0}^{tg\frac{\pi}{8}} \frac{h^{4}}{\left(h^{2}u^{2} + h^{2}\right)^{2}} u du = 2 \int_{0}^{tg\frac{\pi}{8}} \frac{1}{\left(1 + u^{2}\right)^{2}} u du = \int_{0}^{tg^{2}\frac{\pi}{8}} \frac{1}{\left(1 + v\right)^{2}} dv$$
$$= \int_{1}^{1 + tg^{2}\frac{\pi}{8}} \frac{1}{v^{2}} dv = -\frac{1}{v} \Big|_{1}^{1 + tg^{2}\frac{\pi}{8}} = 1 - \frac{1}{1 + tg^{2}\frac{\pi}{8}} = \frac{tg^{2}\frac{\pi}{8}}{1 + tg^{2}\frac{\pi}{8}}$$
$$= \cos^{2}\frac{\pi}{8} tg^{2}\frac{\pi}{8} = \sin^{2}\frac{\pi}{8}$$

#### Esercizio 3.

- Una scena consiste dell'intercapedine fra il piano z=0 considerato come un pavimento infinito, che chiamiamo A<sub>0</sub>, ed un soffitto infinito dato dal piano z=1, che chiamiamo A<sub>1</sub>. Siano x<sub>0</sub>, y<sub>0</sub> due numeri reali. Ovviamente il fattore di forma differenziale di A<sub>0</sub> (ossia il fattore di forma F<sub>dA1-A0</sub> del pavimento visto da un elemento infinitesimo di area dA<sub>1</sub> del soffitto) rispetto al punto V=(x<sub>0</sub>, y<sub>0</sub>,1) deve valere 1, perché il soffitto copre l'intero angolo solido nel semispazio frontale. Verificare questo fatto usando la formula integrale per il fattore di forma differenziale F<sub>dA1-A0</sub>.
- 2. Ora la scena consiste in una stanza cilindrica il cui pavimento è il disco di raggio k centrato nell'origine nel piano z=0, che chiamiamo A<sub>0</sub>(k), ed il soffitto, che chiamiamo A<sub>1</sub>(k), giace nel piano z=1. Calcolare il fattore di forma differenziale da un punto del soffitto a A<sub>0</sub>(k), usando la formula integrale per il fattore di forma differenziale F<sub>dA1-A0</sub>. Calcolare il limite di questo risultato per k che tende ad infinito e verificare che coincide con il caso limite della parte precedente (intercapedine infinita).

Svolgimento.

Parte 1. Il fattore di forma differenziale è

$$F = \int_{A_0} \frac{\cos\theta_0 \, \cos\theta_1}{\pi r^2} \, H_{01} \, d\sigma$$

dove il fattore di occlusione H<sub>01</sub> vale sempre 1 perché non ci sono elementi che si frappongono, r è la lunghezza del segmento da (x<sub>0</sub>, y<sub>0</sub>,1) al punto generico (x, y, 0) in A<sub>0</sub>, e gli angoli  $\theta_0$  e  $\theta_1$  sono quelli che questo segmento forma con il versore normale (0,0,1) al piano z=0 ed il versore normale (0,0,-1) al piano z=1, rispettivamente. È quindi chiaro che cos  $\theta_0 = \cos \theta_1 = 1/r$ , e quindi

$$F = \int_{\{z=0\}} \frac{1}{\pi r^4} dx dy = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{1}{\pi \left(1 + (x - x_0)^2 + (y - y_0)^2\right)^2} dx dy$$

Per l'invarianza per traslazione, l'ultimo integrale è costante rispetto a  $x_0$ ,  $y_0$ . Quindi

$$\mathbf{F} = \int_{0}^{2\pi} \int_{0}^{\infty} \frac{1}{\pi \left(1 + \rho^2\right)^2} \rho \, \mathrm{d}\rho \, \mathrm{d}\theta$$

Ponendo u =  $\rho^2$  si trova

$$F = \int_{0}^{\infty} \frac{1}{(1+u)^{2}} du = \int_{1}^{\infty} \frac{1}{u^{2}} du = 1$$

Parte 2. Procedendo come sopra, ora abbiamo

$$F = \int_{A_0(k)} \frac{1}{\pi r^4} \, dx \, dy = \int_0^{2\pi} \int_0^k \frac{1}{\pi (1 + \rho^2)^2} \, \rho \, d\rho \, d\theta$$

Ponendo u =  $\rho^2$  si trova

$$F = \int_{0}^{k^{2}} \frac{1}{\left(1+u\right)^{2}} du = \int_{1}^{1+k^{2}} \frac{1}{u^{2}} du = \left|-\frac{1}{u}\right|_{1}^{1+k^{2}} = 1 - \frac{1}{1+k^{2}}$$

Chiaramente il limite per k che tende a infinito è 1.

**Esercizio 4.** Si consideri una ciotola K di forma emisferica di raggio 1 appoggiata nel suo punto più basso all'origine, ossia tangente all'origine al piano {z=0} e contenuta nel semispazio superiore { $z \ge 0$ }: quindi il bordo della ciotola è un cerchio J sul piano {z=1}. Poiché K è convessa, il fattore di forma dalla ciotola a sé stessa è positivo.

- 1. Si mostri che l'equazione che definisce K è  $x^2 + y^2 + (z-1)^2 = 1$ , con  $z \le 1$ .
- Si calcoli l'integrale del fattore di forma differenziale F<sub>0K</sub> dall'origine a K e si mostri che vale ½ (suggerimento: si immagini di completare la scena in modo che sia uno spazio chiuso, ad esempio una sfera intera: se seguiamo questo suggerimento aggiungiamo una semisfera con bordo J, che chiamiamo C).
- Sia v=(x<sub>v</sub>, y<sub>v</sub>, z<sub>v</sub>) un altro punto di K. Si scriva esplicitamente l'integrale del fattore di forma differenziale da questo punto a K, in modo che l'integrando sia espresso come funzione solo delle coordinate x<sub>v</sub>, y<sub>v</sub>, z<sub>v</sub> di v e x<sub>p</sub>, y<sub>p</sub>, z<sub>p</sub> di un punto p che varia in K.
- Sia p un punto del bordo J della ciotola. Utilizzando opportuni ragionamenti basati su simmetrie, si dimostri che il fattore di forma differenziale F<sub>pK</sub> da questo punto alla ciotola è ½.

## Svolgimento.

*Parte 1.* Ovvio: la semisfera è la metà inferiore della sfera di raggio 1 e centro (0,0,1).

*Parte 2.* Si consideri la restrizione di K al piano xz. I punti terminali di K in questo piano sono (±1,1). Sia  $\theta$  l'angolo di deviazione dei punti di K dalla direzione verticale (misurata dall'origine): quindi per i punti di K la deviazione è  $\pi/4 \le \theta \le \pi/2$ . Completiamo K ad una sfera intera, e chiamiamo C la semisfera così aggiunta, il cui bordo è J. Chiaramente, per i punti di C, si ha  $0 \le \theta \le \pi/4$ . Per la conservazione dell'angolo solido totale, il fattore di forma differenziale è F<sub>0K</sub> = 1-F<sub>0C</sub> (la somma di tutti i fattori di forma differenziali da un punto a tutta la scena è il 100% dell'angolo solido totale, ossia 1).

Il calcolo di  $F_{0C}$  è analogo a quello svolto negli esercizi precedenti: si può ricavare con il metodo di Nusselt oppure con il calcolo diretto dell'integrale di superficie,

$$F_{0C} = \int_{C} \frac{\cos\theta}{\pi h^2} \, d\sigma = \int_{0}^{2\pi} \int_{0}^{\pi/4} \frac{\cos\theta}{\pi h^2} \, h^2 \sin\theta \, d\theta \, d\varphi = \int_{0}^{\pi/4} \sin 2\theta \, d\theta = \frac{1}{2} \int_{0}^{\pi/2} \sin\theta \, d\theta$$
$$= \frac{1}{2} \Big( 1 - \cos\frac{\pi/2}{2} \Big) = \frac{1}{2} \Big( 1 - \cos^2\frac{\pi/4}{4} + \sin^2\frac{\pi/4}{4} \Big) = \sin^2\frac{\pi/4}{4} = \frac{1}{2} \Big)$$

Pertanto,  $F_{0K} = 1 - F_{0C} = 1 - \sin^2(\pi/4) = 1 - \frac{1}{2} = \frac{1}{2}$ .

Parte 3. L'integrale è

$$F_{vK} = \int_{K} \frac{\cos \theta_{v} \cos \theta_{p}}{\pi r^{2}} H_{01} d\sigma(p)$$

dove il fattore di occlusione  $H_{01}$  vale costantemente 1, r è la distanza fra v e p, ossia la lunghezza del segmento **s** che li unisce, in altre parole

$$r^{2} = (x_{p}-x_{v})^{2} + (y_{p}-y_{v})^{2} + (z_{p}-z_{v})^{2}$$
(1)

e  $\theta_v e \theta_p$  sono gli angoli che le normali a K in v e p (ciascuna coincidente con il rispettivo versore radiale, diretto verso il centro (0,0,1)) formano con il segmento **s**. Chiamiamo o il centro (0,0,1) e consideriamo il triangolo T<sub>K</sub>={o, v, p}. I due lati che si incontrano in o hanno lunghezza 1 (pari al raggio!): ossia, il triangolo è isoscele. Pertanto ciascun coseno nell'integrando è uguale alla lunghezza della proiezione del corrispondente lato di lunghezza 1 sopra il terzo lato **s** che ha lunghezza r, ed evidentemente, visto che il triangolo è isoscele, le due proiezioni sono di lunghezza uguale. Quindi si ha

$$\cos \theta_{\rm v} = \cos \theta_{\rm p} = \frac{1}{2} \tag{2}$$

Ora, in base a (2), l'integrale diventa

$$F_{K} = \frac{1}{4} \int_{K} \frac{1}{\pi r^2} d\sigma(p)$$

Per scrivere esplicitamente l'integrale superficiale ormai resta solo da rammentare quanto vale la misura d $\sigma$ . In base alla parte 1, la superficie è parametrizzabile come

$$\sigma(x_{p}, y_{p}) = (x_{p}, y_{p}, 1 - \sqrt{1 - x_{p}^{2} - y_{p}^{2}})$$

(davanti al radicale abbiamo scelto il segno meno perché  $0 \le z_p \le 1$ ). Per questa parametrizzazione, se poniamo

$$f(x_p, y_p) = 1 - \sqrt{1 - x_p^2 - y_p^2}$$

il fattore di dilatazione dell'area (come già osservato nella Sottosezione *Fattore* locale di ingrandimento delle aree nella parametrizzazione di una superficie, e calcolo dell'area di una superficie parametrica della Sezione Mappa di rilievo del Capitolo **Rendering con mappe parametriche**) è

$$\left\|\frac{\partial \sigma}{\partial x_{p}} \times \frac{\partial \sigma}{\partial y_{p}}\right\| = \sqrt{1 + \left\|\nabla f\right\|^{2}} = \sqrt{1 + \frac{x_{p}^{2} + y_{p}^{2}}{1 - x_{p}^{2} - y_{p}^{2}}} = \sqrt{\frac{1}{1 - x_{p}^{2} - y_{p}^{2}}}$$

Pertanto,

$$F_{vK} = \frac{1}{4} \int_{K} \frac{1}{\pi r^{2}} \sqrt{\frac{1}{1 - x_{p}^{2} - y_{p}^{2}}} dx_{p} dy_{p}$$
$$= \frac{1}{4\pi} \int_{K} \frac{1}{\left(x_{p} - x_{v}\right)^{2} + \left(y_{p} - y_{v}\right)^{2} + \left(z_{p} - z_{v}\right)^{2}} \sqrt{\frac{1}{1 - x_{p}^{2} - y_{p}^{2}}} dx_{p} dy_{p}$$

dove abbiamo usato l'espressione di  $r^2$  data da (1).

*Variante per l'espressione dell'integrale*. Supponiamo di chiudere la ciotola K con un coperchio dato da un disco B nel piano {z=1}. Allora, come prima, si ha  $F_{vK} =$ 1 -  $F_{vB}$  perché gli angoli solidi che K e B sottendono quando visti dal punto v hanno per somma il 100% dell'angolo solido totale. Pertanto basta calcolare  $F_{vB}$ . In questo calcolo, l'integrale è sulla proiezione di B sul piano xy : infatti, le componenti del punto q=(xq, yq,1) che varia in B sono tutte e sole quelle che verificano xq<sup>2</sup> + yq<sup>2</sup> ≤ 1. L'angolo  $\theta_p$  è quello fra il segmento **s** e la normale a B diretta verso il basso, ossia (0,0, -1) : pertanto cos  $\theta_p$  è la proiezione di **s** sull'asse z, ossia

$$\cos \theta_p = 1 - z_v$$

Invece l'angolo  $\theta_v$  è, come prima, quello fra il segmento  $\mathbf{s} = \mathbf{vq} = \mathbf{q} - \mathbf{v}$  e la normale interna a K al punto v, che è orientata come il raggio di K, verso il centro (0,0,1). Si consideri allora il triangolo T<sub>B</sub>={o, v, q}. Ora questo triangolo non è più isoscele, ma di nuovo, come prima, abbiamo

$$\cos \theta_v = \rho$$

dove  $\rho$  è la lunghezza del segmento da v al piede w della proiezione di o su **s**. Calcoliamo questa lunghezza. Poiché il segmento da v a o ha lunghezza 1, è chiaro che la proiezione ortogonale è il prodotto scalare fra il vettore applicato **vo** = o-v ed il vettore applicato **vp** = p-v. Ora, o-v = (-x<sub>v</sub>, -y<sub>v</sub>, 1-z<sub>v</sub>), e quindi il prodotto scalare vale

$$\rho = \langle o-v, p-v \rangle = -x_v(x_p - x_v) - y_v(y_p - y_v) + (1 - z_v)(z_p - z_v)$$
(3)

Seconda variante per l'espressione dell'integrale. Possiamo immaginare di chiudere la ciotola K con un emisfero superiore C, in maniera da completare K ad una sfera. Il valore di cos  $\theta_p$  e di cos  $\theta_v$  si calcolano come nel caso in cui il punto finale q di **s=vq** giaceva in K (il significato di entrambi gli angoli ritorna ad essere quello della deviazione angolare di s rispetto ai versori normali interni alla sfera). Lasciamo i calcoli al lettore.

*Parte 4.* Basta completare la semisfera ad una sfera, aggiungendo una semisfera superiore. Un punto sul bordo J, per simmetria, ha lo stesso fattore di forma verso la semisfera bassa K e la semisfera alta.

### ESEMPIO: CALCOLO DEL FATTORE DI FORMA FRA DUE PARETI ADIACENTI DI UNA STANZA CUBICA

Dobbiamo calcolare il fattore di forma F fra due quadrati con lo stesso lato, incernierati su un lato e disposti perpendicolarmente. osserviamo che, fra i fattori di forma in una stanza i cui sei elementi sono le sei pareti, questo è quello numericamente più delicato, in quanto tale fattore di forma è un integrale improprio quadridimensionale con ordine di divergenza  $1/r^4$ , dove r è la distanza fra punti del primo e del secondo quadrato (si veda l'identità (1) nel seguito); ma il suo calcolo, che si concluse in (10), conferma che l'integrale improprio è convergente.

Introduciamo un opportuno sistema di coordinatre cartesiane. Disponiamo i due quadrati uno sul piano  $\{z = 0\}$  (lo chiamiamo base, o pavimento, e lo indichiamo con **b**) e l'altro sul piano  $\{x = 0\}$  (lo chiamiamo parete, od altezza, e lo indichiamo con **a**). Il fattore di forma rimane lo stesso se dilatiamo i due quadrati di uno stesso fattore, perché l'angolo solido differenziale che uno dei due copre quando visto dai punti dall'altro rimane invariato sotto la dilatazione. Quindi possiamo, per semplicità e senza perdita di generalità, assumere che il lato dei quadrati sia lungo 1. Per fissare le notazioni, diciamo che l'intersezione dei due elementi è il segmento [0, 1] sull'asse y. Quindi la base ha coordinate  $\{0 \leq x \leq 1, 0 \leq y \leq 1\}$  e la parete ha coordinate  $\{0 \leq y \leq 1, 0 \leq z \leq 1\}$ .

Ricordando che **b** è un quadrato di lato 1, e quindi di area 1, scriviamo l'integrale del fattore di forma dalla base alla parete nel modo seguente come integrale quadruplo rispetto alle variabili  $(y_a, z)$  dei punti  $p_a \in a$  ed alle variabili  $(x, y_b)$  dei punti  $p_b \in b$ ):

$$F := \frac{1}{\pi} \frac{1}{\text{Area}(\mathbf{b})} \int_{\mathbf{b}} \int_{\mathbf{a}} \frac{\cos \theta_{\mathbf{b}} \cos \theta_{\mathbf{a}}}{r^2} d\sigma_{\mathbf{a}} d\sigma_{\mathbf{b}}$$
$$= \frac{1}{\pi} \int_0^1 \int_0^1 \int_0^1 \int_0^1 \int_0^1 \frac{xz}{(x^2 + (y_a - y_b)^2 + z^2)^2} dx \, dy_a \, dy_b \, dz \quad (1)$$

Il cambiamento di variabili

$$s = \frac{y_a - y_b}{\sqrt{x^2 + z^2}}$$

trasforma l'integrale al lato di destra di (1) in

$$F = \frac{1}{\pi} \int_0^1 \int_0^1 \int_0^1 \int_{-y_b/\sqrt{x^2 + z^2}}^{(1-y_b)/\sqrt{x^2 + z^2}} \frac{xz\sqrt{x^2 + z^2}}{(x^2 + z^2)^2(1+s^2)^2} \, dx \, ds \, dy_b \, dz \tag{2}$$

Calcoliamo separatamente l'integrale rispetto alla variabile snel prossimo esercizio elementare di Calcolo:

Esercizio 1. La primitiva di  $1/(1+s^2)^2$  è  $\frac{1}{2} \left( \arctan s + \frac{s}{1+s^2} \right) + C.$ 

Svolgimento. Si osservi che

$$\frac{1}{(1+s^2)^2} + \frac{s^2}{(1+s^2)^2} = \frac{1}{1+s^2}$$

e quindi

$$\int \frac{ds}{(1+s^2)^2} = \int \frac{ds}{1+s^2} - \int \frac{s^2 \, ds}{(1+s^2)^2} = \arctan s - \int s \, \frac{s \, ds}{(1+s^2)^2} \tag{3}$$

Osserviamo che grazie alla sostituzione  $u = s^2$ , si calcola la primitiva del secondo fattore dell'ultimo integrando:

$$\int \frac{s \, ds}{(1+s^2)^2} = \frac{1}{2} \int \frac{du}{(1+u)^2} = -\frac{1}{2} \frac{1}{1+u} + C = -\frac{1}{2(1+s^2)} + C.$$

Quindi l'ultimo integrale in (3) si può integrare per parti come segue:

$$\int s \frac{s \, ds}{(1+s^2)^2} = -\frac{s}{2(1+s^2)} + \int \frac{ds}{2(1+s^2)} = \frac{1}{2} \left( -\frac{s}{1+s^2} + \arctan s \right) + C = \frac{1}{2} \left( -\frac{s}{1+s^2} + \arctan s \right) + C = \frac{1}{2} \left( -\frac{s}{1+s^2} + \arctan s \right) + C = \frac{1}{2} \left( -\frac{s}{1+s^2} + \arctan s \right) + C = \frac{1}{2} \left( -\frac{s}{1+s^2} + \arctan s \right) + C = \frac{1}{2} \left( -\frac{s}{1+s^2} + \arctan s \right) + C = \frac{1}{2} \left( -\frac{s}{1+s^2} + \arctan s \right) + C = \frac{1}{2} \left( -\frac{s}{1+s^2} + \arctan s \right) + C = \frac{1}{2} \left( -\frac{s}{1+s^2} + \arctan s \right) + C = \frac{1}{2} \left( -\frac{s}{1+s^2} + \arctan s \right) + C = \frac{1}{2} \left( -\frac{s}{1+s^2} + \arctan s \right) + C = \frac{1}{2} \left( -\frac{s}{1+s^2} + \arctan s \right) + C = \frac{1}{2} \left( -\frac{s}{1+s^2} + \arctan s \right) + C = \frac{1}{2} \left( -\frac{s}{1+s^2} + \arctan s \right) + C = \frac{1}{2} \left( -\frac{s}{1+s^2} + \ln s$$

Pertanto (3) diventa

$$\int \frac{ds}{(1+s^2)^2} = \arctan s - \frac{1}{2} \left( \arctan s - \frac{s}{1+s^2} \right) + C$$
$$= \frac{1}{2} \left( \arctan s + \frac{s}{1+s^2} \right) + C.$$

Grazie al risultato del precedente esercizio, l'integrale al lato di destra di (2) diventa:

$$F = \frac{1}{\pi} \int_{0}^{1} \int_{0}^{1} \int_{0}^{1} \frac{xz\sqrt{x^{2}+z^{2}}}{2(x^{2}+z^{2})^{2}} \left[ \arctan\left(\frac{1-y_{b}}{\sqrt{x^{2}+z^{2}}}\right) + \frac{1-y_{b}}{\sqrt{x^{2}+z^{2}}} \frac{1}{1+\left(\frac{1-y_{b}}{1+\sqrt{x^{2}+z^{2}}}\right)^{2}} + \arctan\left(\frac{y_{b}}{\sqrt{x^{2}+z^{2}}}\right) + \frac{y_{b}}{\sqrt{x^{2}+z^{2}}} \frac{1}{1+\left(\frac{y_{b}}{1+\sqrt{x^{2}+z^{2}}}\right)^{2}} \right] dx \, dy_{b} \, dz \,.$$
(4)

Spezziamo l'ultimo integrale come la somma di  $I_1 + I_2$ , dove

$$I_{1} := \frac{1}{\pi} \int_{0}^{1} \int_{0}^{1} \int_{0}^{1} \frac{xz\sqrt{x^{2}+z^{2}}}{2(x^{2}+z^{2})^{2}} \left[ \arctan\left(\frac{1-y_{b}}{\sqrt{x^{2}+z^{2}}}\right) + \arctan\left(\frac{y_{b}}{\sqrt{x^{2}+z^{2}}}\right) \right] dx \, dy_{b} \, dz \quad (5)$$

е

$$I_{2} := \frac{1}{\pi} \int_{0}^{1} \int_{0}^{1} \int_{0}^{1} \frac{xz\sqrt{x^{2}+z^{2}}}{2(x^{2}+z^{2})^{2}} \left[ \frac{1-y_{b}}{\sqrt{x^{2}+z^{2}}} \frac{1}{1+\left(\frac{1-y_{b}}{1+\sqrt{x^{2}+z^{2}}}\right)^{2}} + \frac{y_{b}}{\sqrt{x^{2}+z^{2}}} \frac{1}{1+\left(\frac{y_{b}}{1+\sqrt{x^{2}+z^{2}}}\right)^{2}} \right] dx \, dy_{b} \, dz \,.$$
(6)

Al fine di calcolare  $I_1$  premettiamo, come eserecizio, il ben noto calcolo (per parti) della primitiva dell'arcotangente:

Esercizio 2.

$$\int_0^x \arctan s \, ds = x \, \arctan x - \frac{1}{2} \, \ln(1+x^2) \, ds$$

Svolgimento. Immaginiamo l'integrando a primo membro come il prodotto dell'arcotangente e della funzione costante 1. Integrando per parti otteniamo

$$\int_0^x \arctan s \, ds = x \, \arctan x - \int_0^x \frac{s}{1+s^2} \, ds = x \, \arctan x - \frac{1}{2} \int_0^{x^2} \frac{u}{1+u} \, du$$
$$= x \, \arctan x - \frac{1}{2} \ln(1+x^2) \, .$$

Ora possiamo calcolare l'integrale  $I_1$  in (5). Applicando al primo integrale in (5) il cambiamento di variabili  $s = (y_b - 1)/(x^2 + z^2)$ , ed al secondo  $s = y_b/(x^2 + z^2)$ , otteniamo, grazie alla disparità della arcotangente ed al precedente esercizio:

$$I_{1} = \frac{1}{\pi} \int_{0}^{1} \int_{0}^{1} \frac{xz\sqrt{x^{2}+z^{2}}}{2(x^{2}+z^{2})^{2}} \left( \int_{1/\sqrt{x^{2}+z^{2}}}^{0} \arctan\left(s(-\sqrt{x^{2}+z^{2}})\right) + \int_{0}^{1/\sqrt{x^{2}+z^{2}}} \arctan\left(s(\sqrt{x^{2}+z^{2}})\right) ds \, dx \, dz$$

$$= \frac{1}{\pi} \int_{0}^{1} \int_{0}^{1} \frac{xz}{x^{2}+z^{2}} \int_{0}^{1/\sqrt{x^{2}+z^{2}}} \arctan s \, ds \, dx \, dz$$

$$= \frac{1}{\pi} \int_{0}^{1} \int_{0}^{1} \frac{xz}{x^{2}+z^{2}} \left[ \frac{1}{\sqrt{x^{2}+z^{2}}} \arctan \frac{1}{\sqrt{x^{2}+z^{2}}} - \frac{1}{2} \ln\left(1+\frac{1}{x^{2}+z^{2}}\right) \right] dx \, dz.$$
(7)

Ora, grazie agli stessi due cambiamenti di variabile che abbiamo applicato a

 $I_1$ , calcoliamo l'integrale  $I_2$  in (6):

$$I_{2} = \frac{1}{\pi} \int_{0}^{1} \int_{0}^{1} \frac{xz\sqrt{x^{2}+z^{2}}}{2(x^{2}+z^{2})^{2}} \left( \int_{1/\sqrt{x^{2}+z^{2}}}^{0} \frac{s}{1+s^{2}} \left( -\sqrt{x^{2}+z^{2}} \right) + \int_{0}^{1/\sqrt{x^{2}+z^{2}}} \frac{s}{1+s^{2}} \left( \sqrt{x^{2}+z^{2}} \right) ds dx dz \right)$$
$$= \frac{1}{\pi} \int_{0}^{1} \int_{0}^{1} \frac{xz}{x^{2}+z^{2}} \int_{0}^{1/\sqrt{x^{2}+z^{2}}} \frac{s}{1+s^{2}} ds dx dz$$
$$= \frac{1}{\pi} \int_{0}^{1} \int_{0}^{1} \frac{xz}{x^{2}+z^{2}} \left[ \frac{1}{2} \ln \left( 1 + \frac{1}{x^{2}+z^{2}} \right) \right] dx dz .$$
(9)

Nel riunire i risultati di (7) e (9) i termini con il logaritmo si cancellano e si ottene:

$$F = I_1 + I_2 = \frac{1}{\pi} \int_0^1 \int_0^1 \frac{xz}{x^2 + z^2} \left[ \frac{1}{\sqrt{x^2 + z^2}} \arctan \frac{1}{\sqrt{x^2 + z^2}} \right] dx \, dz.$$

Quest'ultimo integrale si calcola passando a coordinate cilindriche,  $x = \rho \cos \theta$ ,  $z = \rho \sin \theta$ , ossia  $\rho = \sqrt{x^2 + z^2}$ , come ora mostriamo. Prima, però, osserviamo che per  $\rho \sim 0$  l'integrando ha ordine di infinito  $1/\rho$ , e quindi, come sapevamo che doveva succedere, l'integrale doppio converge (dopo il passaggio a coordinate cilindriche, il fattore  $\rho$  nell'elemento di integrazione elimina la singolarità). Ecco il calcolo:

$$\begin{split} &\frac{1}{\pi} \int_{0}^{1} \int_{0}^{1} \frac{xz}{x^{2} + z^{2}} \left( \frac{1}{\sqrt{x^{2} + z^{2}}} \arctan \frac{1}{\sqrt{x^{2} + z^{2}}} \right) dx \, dz \\ &= \frac{2}{\pi} \int_{0}^{\frac{\pi}{4}} \int_{0}^{1/\cos\theta} \sin\theta \cos\theta \arctan \frac{1}{\rho} \, d\rho \, d\theta \\ &= \frac{2}{\pi} \int_{0}^{\frac{\pi}{4}} \sin\theta \cos\theta \left[ \rho \arctan \frac{1}{\rho} + \frac{1}{2} \ln(1 + \rho^{2}) \right]_{0}^{1/\cos\theta} \, d\theta \\ &= \frac{2}{\pi} \int_{0}^{\frac{\pi}{4}} \sin\theta \cos\theta \left[ \frac{1}{\cos\theta} \arctan \cos\theta + \frac{1}{2} \ln\left(1 + \frac{1}{\cos^{2}\theta}\right) \right] \, d\theta \\ &= \frac{2}{\pi} \int_{1}^{\sqrt{2}/2} t\sqrt{1 - t^{2}} \left[ \frac{1}{t} \arctan t + \frac{1}{2} \ln\left(1 + \frac{1}{t^{2}}\right) \right] \left( -\frac{1}{\sqrt{1 - t^{2}}} \right) \, dt \\ &= \frac{2}{\pi} \int_{\sqrt{2}/2}^{1} \arctan t + \frac{1}{2} t \ln\left(1 + \frac{1}{t^{2}}\right) \, dt \\ &= \frac{2}{\pi} \left[ t \arctan t - \frac{1}{2} \ln(1 + t^{2}) \right]_{\sqrt{2}/2}^{1} \\ &\quad + \frac{1}{2\pi} \left[ t^{2} \ln\left(1 + \frac{1}{t^{2}}\right) + \ln(1 + t^{2}) \right]_{\sqrt{2}/2}^{1} \\ &= \frac{2}{\pi} \left( \frac{\pi}{4} - \frac{1}{2} \ln 2 - \frac{\sqrt{2}}{2} \arctan \frac{\sqrt{2}}{2} + \frac{1}{2} \ln 3 \frac{3}{2} \right) \\ &\quad + \frac{1}{2\pi} \left( 2 \ln 2 - \frac{1}{2} \ln 3 - \ln \frac{3}{2} \right) \\ &= \frac{1}{2} - \frac{\sqrt{2}}{\pi} \arctan \frac{\sqrt{2}}{2} - \frac{2}{\pi} \ln 2 + \frac{3}{2\pi} \ln 2 + \frac{1}{\pi} \left( 1 - \frac{3}{4} \right) \ln 3 \\ &= \frac{1}{2} - \frac{\sqrt{2}}{\pi} \arctan \frac{\sqrt{2}}{2} - \frac{1}{2\pi} \ln 2 + \frac{1}{4\pi} \ln 3 \,. \end{split}$$

Sorprendentemente, l'ultima espressione vale esattamente  $\frac{1}{5}$ , come si può verificare con una calcolatrice (la Figura 1 è il listato del calcolo eseguito in Matlab). Quindi, in una scena cubica, tutte e cinque le facce viste da una data faccia hanno lo stesso fattore di forma.

Fattore\_di\_forma =
1/2-(sqrt(2)/pi)\*atan(sqrt(x)/x)-(1/(2\*pi))\*(log(2)-log(3)/2)
Valore =
0.2000

Figura 1: valore numerico calcolato in Matlab dell'espressione in (10) riguardo il fattore di forma fra quadrati perpendicolari adiacenti.

## SOTTOSTRUTTURAZIONE

Più fine è la suddivisione in elementi, migliore è il risultato ottenuto, ma il prezzo da pagare è un notevole incremento del tempo di calcolo perché il numero di fattori di forma è il quadrato del numero degli elementi, ed il calcolo di ciascun fattore di forma può essere laborioso. Per evitare questo incremento quadratico del numero dei fattori di forma, alcuni studiosi hanno proposto una suddivisione variabile degli elementi, basata sul ripartirli ulteriormente in sottoelementi laddove la radiosità varia più rapidamente. Per velocizzare il procedimento, i sottoelementi non vengono trattate come gli elementi originari. Infatti, per ciascun sottoelemento s si calcola solamente il fattore di forma Fs-i da s verso ciascuna ciascuno degli elementi i originari (usando la tecnica del semicubo), mentre i fattori di forma dagli elementi ai sottoelementi non vengono calcolati. In tal modo si miglioriamo il calcolo della distribuzione di energia ricevuta dagli elementi, considerando la geometria della ricezione di luce in modo più fine, spezzando le aree di ricezione laddove l'intensità è elevata, mentre non si raffiniamo la distribuzione dell'emissione. Dopo che un elemento è stata stato suddiviso, i suoi fattori di forma verso ogni altro elemento vengono sostituiti con stime più accurate, e precisamente con le medie dei fattori di forma dei suoi sottoelementi, pesate in proporzione alle aree:

$$F_{i-j} = \frac{1}{A_j} \sum_{s=1}^m F_{s-j} A_s$$

La radiosità di ogni sottoelemento s di un elemento i viene calcolata riscrivendo il sistema lineare della radiosità (presentato nella sezione *L'equazione della radiosità*) rispetto ai sottoelementi, ma mantenendo per ciascuna sottoelemento il coefficiente di riflessione dell'elemento da cui proviene (i coefficienti di riflessione per gli elementi originarie sono assegnati in fase di modellazione: non abbiamo a disposizione una stima più precisa per quelli dei sottoelementi). Di solito viene anche mantenuta per il sottoelemento la componente E<sub>i</sub> di energia luminosa creata internamente che vale per l'elemento da cui essa proviene, perché si è interessati ad una migliore distribuzione dell'energia ricevuta e non di quella emessa (e quindi normalmente non si sottostrutturano le sorgenti di luce), ma se si decide di spezzare in sottoelementi anche le sorgenti di luce si può usare, per ciascun elemento s, un suo valore specifico E<sub>s</sub> di energia creata.

Quindi abbiamo ora una stima più raffinata anche della radiosità dei sottoelementi, come media pesata in proporzione alle aree:

$$B_s = E_i + \rho_i \sum_{j=1}^n B_j F_{s-j}$$

Questa stima è stata ottenuta a partire dai valori già noti di B<sub>j</sub> e quindi non richiede la risoluzione di un sistema lineare della radiosità a dimensione più elevata; peraltro, essa tiene conto in modo più accurato di come la illuminazione si distribuisce sui sottoelementi, ma non di come questi la irradiano a propria volta in modo più appropriato: per il calcolo della radiosità degli altri elementi si usa sempre B<sub>i</sub>, non le B<sub>s</sub>. L'algoritmo procede per iterazione, suddividendo gli elementi nelle zone dove la radiosità è maggiore, fino a quando il miglioramento ottenuto dalla suddivisione non raggiunge una soglia sufficientemente piccola prefissata. Gli elementi ottenuti da questo processo vengono usate per determinare la radiosità nei vertici.

Il fatto di determinare le radiosità ai singoli vertici è particolarmente vantaggioso per come sono strutturati gli algoritmi di shading, tipicamente quelli di Gouraud e di Phong, che si basano su metodi di interpolazione a partire dai dati sui vertici.

## RICHIAMI SUL SISTEMA LINEARE DELLA RADIOSITÀ

Il sistema lineare della radiosità ricavato precedentemente (nella sezione sull'*Equazione della radiosità*) è del tipo Mb = e, dove e è il vettore dell'energia (o meglio potenza) luminosa creata dal singolo elemento (luce propria) e b è il vettore della radiosità, cioè dell'energia luminosa totale diffusa (quella creata o riflessa).

Se  $\rho_i$  indica il coefficiente di riflettività dell'elemento numero i e  $F_{ii}$  è il fattore di forma dall'elemento i a se stesso (che misura quale frazione dell'energia emessa dall'elemento i ritorna a quell'elemento), si ha:

$$0 \le \rho_i < 1$$
 e  $0 \le F_{ii} \le 1$ 

Abbiamo precedentemente visto che la matrice M del sistema lineare della radiosità verifica:

$$M_{ii} = 1 - \rho_i F_{ii}$$
  $i = 1, ..., n$ 

Abbiamo già osservato, nella sezione "<u>L'equazione della radiosità</u>", che  $\rho_i < 1 e \rho_i F_{ii} < 1$ . Quindi  $0 < M_{ii} \le 1$  per ogni i: i termini diagonali di M non si annullano. Perciò il sistema Mb=e, ossia

$$M_{ii}b_i + \Sigma_{j=1,...n, \ j\neq i} \ M_{ij}b_j = e_i \qquad \qquad i=1,..., \ n$$

diventa :

$$\mathbf{b}_{i} = -\sum_{j=1,...n, j \neq i} (\mathbf{M}_{ij}/\mathbf{M}_{ii})\mathbf{b}_{j} + \mathbf{e}_{i}/\mathbf{M}_{ii}$$
  $i = 1,..., n$  (1)

Questo sistema si può risolvere con metodi iterativi.

# SOLUZIONE CON METODI ITERATIVI: RILASSAMENTO DI JACOBI

Il più semplice metodo iterativo è quello di Jacobi: si parte con qualche vettore iniziale  $b^{(0)}$  al posto della soluzione b esatta da trovare, e si sostituisce nel lato destro di (<u>1</u>), per ottenere una prima approssimazione iterativa:

$$\mathbf{b}_{i}^{(1)} = -\sum_{j=1,...n, j\neq i} (\mathbf{M}_{ij}/\mathbf{M}_{ii})\mathbf{b}_{j}^{(0)} + \mathbf{e}_{i}/\mathbf{M}_{ii} \qquad i = 1,..., n$$
(2)

(rammentiamo che gli elementi diagonali  $M_{ii}$  sono non nulli). Le componenti di  $b^{(1)}$  vengono così determinate una dopo l'altra, in un primo ciclo (per i da 1 a N) che chiameremo la prima iterazione, ovvero il primo ciclo di iterazione. Poi si procede in modo analogo, ciclo dopo ciclo; alla m-sima iterazione determiniamo  $b^{(m)}$  da

$$b_i^{(m)} = - \Sigma_{j=1,...n, j \neq i} (M_{ij}/M_{ii}) b_j^{(m-1)} + e_i/M_{ii}$$
 (3)

Si dice che il metodo è convergente se i  $b^{(m)}$  convergono ad un vettore limite b, che in tal caso è necessariamente soluzione del sistema (1).

Si noti che tutte le componenti di  $b^{(m)}$  vengono aggiornate insieme nello stesso ciclo, utilizzando tutte le componenti di  $b^{(m-1)}$ : quindi il metodo di Jacobi richiede di conservare in memoria simultaneamente due diversi aggiornamenti del vettore della radiosità:  $b^{(m)}$  e  $b^{(m-1)}$ .

Metodi di questo tipo si chiamano anche *metodi di rilassamento*. Ora ne vedremo due varianti più efficienti.

## RILASSAMENTO DI GAUSS-SEIDEL APPLICATO AL SISTEMA LINEARE DELLA RADIOSITÀ

Il metodo iterativo di Gauss-Seidel è simile al metodo di Jacobi, tranne che, quando si svolge il ciclo di iterazione (3), le n equazioni vengono anche qui risolte una dopo l'altra in ordine progressivo (per i da 1 a n), ma, nella i-esima equazione, al membro di destra si utilizzano ove possibile le componenti già precedentemente aggiornate (nelle equazioni precedenti del ciclo, da 1 a i-1) del vettore b<sup>(m)</sup> invece che quelle di b<sup>(m-1)</sup>. Per le componenti non ancora aggiornate, cioè da i +1 a n, si continua a utilizzare b<sup>(m-1)</sup>. Quindi invece di (2) abbiamo:

$$b_i^{(m)} = -\sum_{j=1}^{i-1} \frac{M_{ij}}{M_{ii}} b_j^{(m)} - \sum_{j=i+1}^n \frac{M_{ij}}{M_{ii}} b_j^{(m-1)} + \frac{e_i}{M_{ii}}$$
(4)

In una prossima sezione dal titolo *Convergenza del metodo di Gauss-Seidel* mostreremo che il metodo di Gauss-Seidel è convergente se la matrice M ha la seguente proprietà di predominanza diagonale stretta:

**Definizione.** Una matrice  $n \times n$   $M \rightleftharpoons a$  predominanza diagonale stretta se, per ogni i = 1, ..., n si ha

$$|\mathsf{M}_{\mathrm{ii}}| > \Sigma_{\mathrm{j=1,..n, j\neq i}} |\mathsf{M}_{\mathrm{ij}}|$$

**Lemma.** La matrice M del sistema (<u>1</u>) della radiosità è a predominanza diagonale stretta.

*Dimostrazione.* Abbiamo visto nella sezione dal titolo *L'equazione della radiosità* che, per  $1 \le i \le N$ , si ha

$$M_{ii} = 1 - \rho_i F_{ii}$$

Inoltre, per  $1 \le j \le n$ , si ha  $\sum_{j=1,..n} F_{ij} = 1$  (qui supponiamo che anche lo sfondo sia un elemento, e quindi che l'area totale coperta da tutti gli elementi vista dall'elemento numero i sia l'intero angolo solido del semispazio anteriore, di  $\pi$  steradianti).

Sempre nella stessa sezione abbiamo anche visto che, per  $i \neq j$ , si ha  $M_{ij} = -\rho_i F_{ij}$ . Perciò  $M_{ii} = 1 - \rho_i F_{ii} = \Sigma_{j=1,..n} F_{ij} - \rho_i F_{ii}$ .

Ma  $\rho_i$ <1 e M<sub>ii</sub>>0 per ogni i. Quindi

$$|\mathsf{M}_{ii}| > \Sigma_{j=1,..n} \, \rho_i \; \mathsf{F}_{ij} - \rho_i \; \mathsf{F}_{ii} = \Sigma_{j=1,..N, \; j \neq \; i} \, \rho_i \; \mathsf{F}_{ij} = \Sigma_{j=1,..n, \; j \neq \; i} \; |\mathsf{M}_{ij}| \; ,$$

il che completa la dimostrazione.

## ANALISI DEL METODO DI GAUSS-SEIDEL

#### 1. Metodi di rilassamento

Dato un operatore lineare M su  $C^n$ , i metodi di rilassamento sono mirati a trasformare il problema lineare

Mb = e

in un problema equivalente di punto fisso che sia risolvibile in maniera rapida ed efficiente. Fissata una base in  $C^n$ , M si identifica con una matrice n x n. La trasformazione del problema si ottiene attraverso uno spezzamento della matrice M:

$$M = N - P$$

con N invertibile. Così si ottiene:

Nb = Pb + e $b = N^{-1}Pb + N^{-1}e$  (5)

Il problema Mb=e è quindi trasformato nel problema di punto fisso

$$b = Qb + a \tag{5'}$$

dove:

$$a = N^{-1} e e Q = N^{-1}P$$
 (5")

La matrice Q è chiamata la *matrice di iterazione*.

Ora, per ogni M c'è un unico spezzamento M = N - P con N triangolare inferiore e P triangolare strettamente superiore (cioè con zeri sulla diagonale). Sappiamo già che gli elementi diagonali di N sono non nulli e quindi N è invertibile. Il problema di punto fisso viene quindi affrontato con l'iterazione

$$b^{(k+1)} = Q b^{(k)} + a$$
 (6)

ossia

$$b^{(k+1)} = N^{-1}P b^{(k)} + N^{-1}e$$
 (6')

a partire da un vettore iniziale b<sup>(0)</sup> assegnato. Affinché questo approccio sia vantaggioso occorre che il sistema lineare

$$N y = P b^{(k)} + e$$

(dove il termine di destra è un vettore noto e y è un vettore incognito) sia risolvibile per y in maniera rapida, cioè con un tempo di calcolo trascurabile rispetto a quello richiesto per la risoluzione del problema originale Mb = e. Per questo si cerca di ridursi al caso in cui l'operatore invertibile N sia espresso da una matrice triangolare (inferiore o superiore), perché in questo caso la soluzione avviene per sostituzioni successive di ogni equazione del sistema lineare nella precedente. Qui abbiamo scelto N triangolare inferiore.

Ricordiamo che il raggio spettrale di un operatore lineare T su  $C^n$  è definito da

$$\rho(\mathsf{T}) = \limsup_{k \to \infty} ||\mathsf{T}^k||^{1/k}$$

E' ovvio che  $\rho(T) \le ||T||$ , perché  $||UV|| \le ||U|| ||V||$  per ogni coppia di operatori lineari su  $C^{N}$ .

Ora proveremo che, se la matrice M ha raggio spettrale minore di 1, allora l'errore tende a zero qualunque sia il vettore iniziale  $b^{(0)}$  (e vedremo dopo che questa condizione è anche necessaria per la convergenza), ma può accadere che, per specifiche scelte del vettore iniziale, l'errore tenda a zero senza che la matrice abbia raggio spettrale minore di 1. Ciò accade quando definitivamente il nucleo delle matrici  $M^k$  include  $b^{(0)}$  - b (come si vede dalla identità (6'') più sotto; di solito si sceglie  $b^{(0)} = 0$ , quindi la condizione è che il punto fisso b sia nel nucleo delle matrici  $M^k$  da un certo k in poi). D'altra parte, se l'errore tende a zero per ogni vettore iniziale  $b^{(0)}$ , allora M deve avere raggio spettrale minore di 1. Ecco la dimostrazione: definendo l'errore alla k-sima iterazione (k>0) come

$$\eta^{(k)} = b^{(k)} - b$$

da (6'), (5), (5') e (5'') si ottiene per questi errori la relazione di ricorrenza

$$\eta^{(k+1)} = b^{(k+1)} - b = N^{-1}P \ b^{(k)} + N^{-1}e - b = Q \ b^{(k)} - Qb = Q\eta^{(k)}$$

e quindi

$$\eta^{(k+1)} = Q\eta^{(k)} = Q^2 \eta^{(k-1)} = \dots = Q^{k+1} \eta^{(0)}$$
 (6")

(E' da qui che segue cio' che abbiamo accennato poco sopra: se  $\eta^{(0)} = b^{(0)} - b$  è nel nucleo di  $\bigcap_{i=k+1}^{\infty} Q^{i}$  allora  $0 = \eta^{(k+1)} = \eta^{(k+2)} = \eta^{(k+3)} \dots$ )

Dalla definizione di raggio spettrale,

$$\limsup_{k\to\infty} ||\mathbf{Q}^k||^{1/k} = \rho(\mathbf{Q})$$

segue che, se  $\rho(Q) < 1$ ,

$$\limsup_{k \to \infty} ||\mathbf{Q}^{k}|| \cong \limsup_{k \to \infty} \rho(\mathbf{Q})^{k} = 0,$$

ma poiché  $||Q^k|| \ge 0$ , questo implica

$$0 \le \lim_{k \to \infty} \left\| Q^k \right\| \le \lim_{k \to \infty} \sup \left\| Q^k \right\| = 0$$

e perciò  $\lim_{k\to\infty} ||Q^k|| = 0$ , e

$$\lim_{k\to\infty} \eta^{(k)} = \lim_{k\to\infty} Q^k \eta^{(0)} = 0.$$

Viceversa, se  $\rho(Q) \ge 1$ , in base alla stima  $||Q^k|| \cong \rho(Q)^k$  devono esistere vettori  $v_k$  di norma 1 tali che, per  $k \to \infty$ ,  $Q^k v_k$  ha norma  $\ge 1$  (esponenzialmente divergente nel caso  $\rho(Q) > 1$ ). Poiché la superficie sferica in C<sup>n</sup> di raggio 1 è compatta, la successione  $v_k$  ha una sottosuccessione convergente ad un vettore limite v (si riveda la compattezza nel corso di Analisi Matematica 1 o, ad esempio, nell'appendice di <u>Appunti di Algebra Lineare</u>), per il quale quindi si ha che Q<sup>k</sup>v ha norma  $\ge 1$  per k  $\to \infty$ , e quindi, se si sceglie b<sup>(0)</sup> =v, la successione Q<sup>k</sup>b<sup>(0)</sup> non converge ad un punto fisso.

Abbiamo dimostrato:

**Teorema 1.** Condizione necessaria e sufficiente affinché il metodo iterativo (6) converga per ogni scelta di vettore iniziale è che  $\rho(Q) < 1$ .

#### 2. Metodo di Gauss-Seidel (metodo delle sostituzioni successive)

La decomposizione adottata nel metodo di Gauss-Seidel è la seguente: M = L + D + U, dove D è una matrice diagonale (la diagonale di M) ed L e U sono la parte triangolare (strettamente) inferiore e superiore di M. Allora poniamo N = L+D e P=–U, ed osserviamo che la matrice N è triangolare (inferiore), e quindi ha gli stessi autovalori di D, cioè ha per autovalori gli elementi diagonali M<sub>ii</sub> di M. Ma abbiamo visto\_nella sezione dal titolo *Equazione della radiosità* che M<sub>ii</sub> > 0 per ogni i = 1,...,n: quindi N è invertibile.

Ora, nel calcolo di  $b_i^{(k+1)}$ , utilizziamo al secondo membro i valori di  $b_j^{(k+1)}$  invece di  $b_j^{(k+1)}$  per i<j (e per comodita' di scrittura portiamoli al primo membro). Allora l'iterazione

$$Nb^{(k+1)} = Pb^{(k)} + e$$

assume la forma



Come si può osservare, in questo metodo (delle sostituzione successive) il valore aggiornato di ogni singola componente  $b_j^{(k+1)}$  (*j*=1,....,n) viene utilizzato immediatamente per il calcolo delle componenti successive relative alla stessa iterazione, mentre in generale, nella risoluzione del sistema (6) con il metodo di Jacobi (introdotto nella sezione dal titolo *Metodi iterativi*), i valori aggiornati di tutte le componenti vengono sostituiti simultaneamente alla fine dell'iterazione.

#### 3. Convergenza del metodo di Gauss-Seidel

Consideriamo un sistema lineare Ax=b, e le sue iterazioni con le sostituzioni successive nel senso di (7) (dove invece di una matrice quadrata generica A abbiamo considerato, come in (4), la matrice M della radiosità). Se il procedimento iterativo converge, la soluzione del sistema (4) (qui scritto come in (7)) verifica quindi:

$$x_{i} = -\sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_{j} - \sum_{j=i+1}^{n} \frac{a_{ij}}{a_{ii}} x_{j} + \frac{b_{i}}{a_{ii}}$$
 (i=1,....,n)

Mentre

$$x_i^{(k+1)} = -\sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{(k+1)} - \sum_{j=i+1}^n \frac{a_{ij}}{a_{ii}} x_j^{(k)} + \frac{b_i}{a_{ii}} \ .$$

Gli errori

$$\eta_i^{(k)} = x_i^{(k)} - x_i$$

risolvono ad ogni iterazione il seguente sistema lineare:

$$\eta_i^{(k+1)} = -\sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} \eta_j^{(k+1)} - \sum_{j=i+1}^n \frac{a_{ij}}{a_{ii}} \eta_j^{(k)}$$

Perciò:

$$\left|\eta_{i}^{(k+1)}\right| \leq \sum_{j=1}^{i-1} \left|\frac{a_{ij}}{a_{ii}}\right| \left|\eta_{j}^{(k+1)}\right| + \sum_{j=i+1}^{n} \left|\frac{a_{ij}}{a_{ii}}\right| \left|\eta_{j}^{(k)}\right| \qquad (i=1,\ldots,n)$$
(7')

e quindi:

$$\left|\eta_{i}^{(k+1)}\right| \leq \sum_{j=1}^{i-1} \left|\frac{a_{ij}}{a_{ii}}\right| \left\|\eta^{(k+1)}\right\| + \sum_{j=i+1}^{n} \left|\frac{a_{ij}}{a_{ii}}\right| \left\|\eta^{(k)}\right\| \qquad (i=1,\ldots,n) \qquad (7'')$$

Detto  $M=M_k$  l'indice i per il quale il secondo termine della disuguaglianza è massimo, e scrivendo per brevità:

$$R_{M} = \sum_{j=1}^{M-1} \left| \frac{a_{M j}}{a_{MM}} \right| \quad \mathbf{e} \qquad S_{M} = \sum_{j=M+1}^{n} \left| \frac{a_{M j}}{a_{MM}} \right|$$

si ha

$$\left|\eta_{i}^{(k+1)}\right| \leq R_{M} \left\|\eta^{(k+1)}\right\| + S_{M} \left\|\eta^{(k)}\right\|$$
 (i=1,...,n)

e quindi, scegliendo come norma la norma  $\ell^{\infty}$ , ossia  $\|\eta^{(k+1)}\| = \max_{i=1...n} |\eta_i^{(k+1)}|$  otteniamo

$$\|\eta^{(k+1)}\| \le R_M \|\eta^{(k+1)}\| + S_M \|\eta^{(k)}\|$$
 (7''')

(a dimensione finita tutte le norme sono equivalenti, si veda l'Appendice degli <u>Appunti di Algebra Lineare</u>; ma anche senza questa scelta di norma, da (7') si deduce (7"), e quindi (7"), in qualsiasi scelta di norma, ad esempio nella consueta norma Euclidea). Pertanto

$$\left(1-R_{M}\right)\left\|\eta^{(k+1)}\right\|\leq S_{M}\left\|\eta^{(k)}\right\| \ .$$

Se supponiamo A a predominanza diagonale stretta, cioè se

$$\left|a_{ii}\right| > \sum_{i \neq j}^{n} \left|a_{ij}\right|$$

si ha:

$$R_M + S_M < 1$$

Ne segue che

$$R_M < 1 e S_M < 1$$

е

$$S_M / (1 - R_M) < 1$$

In questa disuguaglianza, l'indice M= M<sub>k</sub> dipende dalla scelta di k. Però M varia solo nell'insieme finito da 1 a n, e quindi la costante  $S_M/(1-R_M)$  varia in un insieme finito di numeri tutti minori di 1, in base all'ultima disuguaglianza. Con abuso di notazione, indichiamo il suo valore massimo ancora con  $S_M / (1-R_M) < 1$ . Allora

$$\left\|\eta_{i}^{(k+1)}\right\| \leq \frac{S_{M}}{1-R_{M}} \left\|\eta^{(k)}\right\| \leq \left(\frac{S_{M}}{1-R_{M}}\right)^{k+1} \left\|\eta^{(0)}\right\|$$
(8)

Ora, grazie alla disuguaglianza  $S_M/(1-R_M) < 1$ , segue da (8) che  $\lim_{k\to\infty} \eta_i^{(k+1)} = 0$  (ed osserviamo anche che da questo e dalla dimostrazione del precedente Teorema 1 segue che il raggio spettrale della matrice Q è minore di 1).

Abbiamo dimostrato dunque il seguente teorema.

**Teorema 2.** Se la matrice A è a predominanza diagonale stretta allora il metodo di Gauss-Seidel è convergente.

Poiché abbiamo dimostrato in un Lemma precedente che la matrice della radiosità è a predominanza diagonale stretta, questo dimostra il fatto seguente:

**Corollario.** Il metodo di Gauss-Seidel, applicato al problema della radiosità, è convergente.

# RILASSAMENTO DI SOUTHWELL APPLICATO AL SISTEMA LINEARE DELLA RADIOSITÀ

## 1. Rilassamento di Southwell

Abbiamo visto nella sezione precedente che in ogni ciclo di iterazione, da ognuna delle n equazioni nell'ordine j = 1, ..., n, il metodo di Gauss-Seidel calcola la componente j di una approssimazione del valore di radiosità di ogni singolo elemento.

Ora presentiamo il metodo di Southwell, che procede in ordine diverso: anch'esso calcola le varie componenti della nuova approssimazione risolvendo una equazione lineare alla volta, ma la scelta di j (cioè di quale equazione usare) è fatta in modo da rendere massima l'efficienza.

Quando si studiano scambi di radiosità questi paragoni sono più significativi se compiuti su valori di energia (emessi da un singolo elemento) piuttosto che su valori di radiosità (energia per unità di area).

Riscriviamo l'equazione della radiosità in termini di valore di energia per elemento e definiamo nuovi vettori  $\beta$  e  $\epsilon$  come

 $\begin{array}{ll} \beta_i = A_i \ b_i \\ \text{riflessa}) \\ \epsilon_i = A_i \ e_i \end{array} \qquad \begin{array}{ll} \text{energia totale che lascia l'elemento i (emessa internamente o (8'))} \\ \text{energia emessa internamente dall'elemento i} \end{array}$ 

Quindi riscriviamo l'equazione della radiosità, trasformando la matrice M del sistema lineare (introdotta nella sezione dal titolo *L'equazione della radiosità*) in una nuova matrice K, che rappresenta il sistema lineare dell'energia:

$$K\beta = \epsilon$$
, con  $K_{ij} = (A_i/A_j) M_{ij}$  (9)

Grazie alla relazione di reversibilità  $A_i F_{i-j} = A_j F_{j-i}$  (anch'essa introdotta nella sezione dal titolo *L'equazione della radiosità*), nel calcolo degli elementi della matrice K si possono far apparire i termini  $F_{j-i}$  invece che gli  $F_{i-j}$  che abbiamo usato nella matrice M. Quindi si ha:

$$K_{ij} = (A_i/A_j) M_{ij} = (A_i/A_j) (\delta_{ij} - \rho_i F_{i-j})$$

Si osservi ora che, grazie alla relazione di reversibilità, per i≠j, si ha

$$K_{ij} = -\rho_i F_{j-i}$$
(10)

mentre, nel caso i=j, si ha  $K_{ii}$ = 1 –  $\rho_i F_{i-i}$ . Pertanto, per ogni i, j, si ottiene

$$K_{ij} = \delta_{ij} - \rho_i F_{j-i} \tag{11}$$

ed inoltre

se 
$$F_{i-i} = 0$$
, allora  $K_{ii}=1$  (12)

L'algoritmo di Southwell funziona calcolando una serie di approssimazioni ( $\beta^{(k)}$ ) della soluzione  $\beta$ . Il vettore degli errori, o *vettore dei resti*, o anche *resto*, che ci permette di stimare la qualità dell'approssimazione, è:

$$\mathbf{r}^{(k)} = \mathbf{\epsilon} - \mathbf{K} \boldsymbol{\beta}^{(k)} \tag{13}$$

Il fatto che  $\beta^{(k)}$  converga alla soluzione esatta  $\beta$  equivale al fatto che il vettore resto r<sup>(k)</sup> tenda a 0. Notiamo che

$$K\beta_{i}^{(k)} = \sum_{j} K_{ij} \beta_{j}^{(k)} = \sum_{j} (A_{i} A_{j})M_{ij} \beta_{j}^{(k)} = A_{i} \sum_{j} M_{ij} b_{j}^{(k)} = A_{i} (Mb^{(k)})_{i}$$

Allora dalla definizione di  $\epsilon$  e  $\beta$  e da (9) segue che la i-sima componente del vettore resto alla k-sima iterazione è data da

$$r_i^{(k)} = A_i ((e_i - (Mb^{(k)})_i))$$
 (13')

L'algoritmo calcola ad ogni passo una nuova approssimazione  $\beta^{(k+1)}$  ottenuta modificando un singolo componente del vettore  $\beta^{(k)}$  in modo tale che una delle componenti di r<sup>(k)</sup> diventi nulla: resto 0 per quella componente. La componente risolta ad una data iterazione si dice *rilassata*. A tal fine, vediamo cosa comporta il fatto che ad ogni iterazione l'algoritmo si limiti a risolvere (ovvero a rilassare) solo una delle n equazioni del sistema. Supponiamo che alla iterazione k+1 venga risolta la equazione numero i: questo vuol dire, per (6'), che si trova un approssimante b<sup>(k)</sup> tale che

$$b_i^{(k+1)} = (N^{-1}P b^{(k)})_i + (N^{-1}e)_i.$$

Invece per le altre componenti b<sup>(k+1)</sup> coincide con b<sup>(k)</sup>. Nel metodo di Southwell, l'equazione da rilassare ad ogni iterazione viene scelta come quella che alla iterazione precedente dava luogo alla componente del vettore resto maggiore delle altre (ossia, diciamo così, viene rilassata la componente con l'errore più grande).

Quindi la differenza dal metodo di Gauss-Seidel è che questo aggiorna in maniera progressiva tutte le componenti del vettore della soluzione approssimata alla iterazione precedente, mentre il metodo di Southwell sceglie nel vettore resto r<sup>(k)</sup> la coordinata *i* alla quale corrisponde la componente più grande e rilassa la corrispondente componente  $\beta_i^{(k)}$  del vettore in modo da annullare r<sub>i</sub><sup>(k)</sup>.

Esaminiamo l'andamento dell'errore di approssimazione nel metodo di Southwell. Supponiamo che, al passo k, la componente più grande del vettore resto sia r<sub>i</sub><sup>(k)</sup> e che al passo successivo diventi zero:

$$0 = r_i^{(k+1)} = \varepsilon_i - \sum_{j=1,...n} K_{ij} \beta_j^{(k+1)}$$
(14)

Poiché modifichiamo soltanto la componente i si ha

$$\beta_j^{(k+1)} = \beta_j^{(k)}$$
 per j diverso da i (15)

mentre per la componente i segue da (14) che

$$\beta_{i}^{(k+1)} = \frac{1}{K_{ii}} \left( \varepsilon_{i} - \sum_{i \neq j} K_{ij} \beta_{j}^{(k)} \right)$$
(16)

Ossia, sempre da (14),

$$\beta_{i}^{(k+1)} = \frac{1}{K_{ii}} \left( \varepsilon_{i} - \sum_{j=1}^{n} K_{ij} \beta_{j}^{(k)} + K_{ii} \beta_{i}^{(k)} \right) = \beta_{i}^{(k)} + \frac{r_{i}^{(k)}}{K_{ii}}$$
(17)

Da (13) segue che

$$\mathbf{r}^{(k+1)} - \mathbf{r}^{(k)} = -K \left(\beta^{(k+1)} - \beta^{(k)}\right)$$
(18)

Da (18), grazie a (15) e (17), nella moltiplicazione riga per colonna all'ultimo membro entra in gioco solo la colonna i-sima della matrice K (perche solo la riga i-sima del vettore  $\beta^{(k+1)} - \beta^{(k)}$  è non nulla) e si ottiene

$$r_{j}^{(k+1)} = r_{j}^{(k)} - (K_{ji} / K_{ii}) r_{i}^{(k)}$$
(19)

Ripetiamo quanto sopra in un breve sommario. L'algoritmo di Southwell calcola per iterazione una soluzione approssimata  $\beta^{(k)}$  e stima l'errore di approssimazione, cioè il vettore resto r<sup>(k)</sup>.

A questo punto l'algoritmo trova quale è la componente del vettore resto più grande in valore assoluto ed aggiorna quella componente in maniera che il resto corrispondente diventi nullo. In questo modo cambiano anche le altre componenti; l'algoritmo ripete l'iterazione su quella di grandezza maggiore e così via.

Durante ogni iterazione viene usata una sola colonna della matrice K, quindi dobbiamo calcolare una sola riga della matrice dei fattori di forma (grazie a (10), la matrice K, al di fuori della diagonale, è la trasposta della matrice dei fattori di forma). La scelta del vettore  $\beta^{(0)}$  è arbitraria: infatti ora dimostriamo che l'algoritmo converge per qualunque scelta iniziale.

#### 2. Convergenza del metodo di Southwell

Consideriamo la norma del vettore resto. Dobbiamo dimostrare che essa converge a zero quando iteriamo il procedimento all'infinito. Al posto della norma Euclidea

$$|||\mathbf{r}^{(k)}||| = \left[\sum_{i=1,...n} (\mathbf{r}_{i}^{(k)})^{2}\right]^{1/2}$$

è equivalente usare la norma  $\ell^1$  data da

$$||\mathbf{r}^{(k)}|| = \sum_{i=1,...n} |\mathbf{r}_i^{(k)}|$$

perché

$$|||r^{(k)}||| \le ||r^{(k)}|| \le n^{\frac{1}{2}} |||r^{(k)}|||$$

grazie alla disuguaglianza di Cauchy-Schwarz (si vedano gli <u>Appunti di Algebra</u> <u>Lineare</u> (Appendice) o di <u>AnalisiArmonica</u>). (Rammentiamo comunque di nuovo che in dimensione finita tutte le norma sono equivalenti: si veda l'Appendice delle <u>Appunti di Algebra Lineare</u>.) Quindi vogliamo provare che

$$\lim_{k\to\infty} ||r^{(k)}|| = \lim_{k\to\infty} \sum_{i=1,\dots,n} |r_i^{(k)}| = 0$$

Da (19) abbiamo :

$$\left\| r^{(k+1)} \right\| = \sum_{\substack{j=1,\dots,n\\j\neq i}} \left| r_j^{(k)} - \frac{K_{ji}}{K_{ii}} r_i^{(k)} \right|$$
(20)

Poiché stiamo supponendo che alla iterazione k-sima la componente più grande del vettore  $r^{(k)}$  sia la i-esima, cioè che sia questa la componente che viene rilassata alla iterazione k+1, segue da (13) che  $r_i^{(k+1)} = 0$ .

Ora dalla disuguaglianza triangolare abbiamo

$$\left\| r^{(k+1)} \right\| \le \sum_{j \ne i} \left| r_{j}^{(k)} \right| + \sum_{j \ne i} \left| \frac{K_{ji}}{K_{ii}} r_{i}^{(k)} \right| \le \left\| r^{(k)} \right\| - \left| r_{i}^{(k)} \right| + \left| r_{i}^{(k)} \right| \sum_{j \ne i} \left| \frac{K_{ji}}{K_{ii}} \right|$$
(21)

Poiché la matrice K è a predominanza diagonale stretta, per ogni i,  $1 \le i \le n$ , si ha:

$$\left|K_{ii}\right| > \sum_{j \neq i} K_{ji} \tag{22}$$

Per i=1,...,n, poniamo

$$t = \max \sum_{j \neq i} \left| \frac{K_{ji}}{K_{ii}} \right|$$
(2). e

In altre parole, 0 < t < 1 per (22),

$$\sum_{i=1,\dots,n,\,j\neq i} \left| \frac{K_{ji}}{K_{ii}} \right| < t \quad \text{per } 1 \le i \le n$$
 (23)

Pertanto segue da (21) che:

$$\|r^{(k+1)}\| \le \|r^{(k)}\| - (1-t)|r_i^{(k)}|$$
 (24)

Per l'ipotesi del metodo di Southwell, all'iterazione k-sima la componente più grande del vettore r è la i-esima. Perciò:

$$\left\|\boldsymbol{r}^{(k)}\right\| \le n \left|\boldsymbol{r}_{i}^{(k)}\right| \tag{25}$$

Quindi

$$(1-t)\left|r_{i}^{(k)}\right| \ge \frac{1-t}{n} \left\|r^{(k)}\right\|$$
 (26)

e, sostituendo (26) in (24), otteniamo:

$$\left\|r^{(k+1)}\right\| \le \left(1 - \frac{1-t}{n}\right) \left\|r^{(k)}\right\|$$
 (27)

Osserviamo che, se si pone

$$T = 1 - \frac{1 - t}{n}$$

applicando iterativamente (27) si ottiene

$$\|r^{(k+1)}\| \le T^{k+1}\|r^{(0)}\|$$
 (28)

Poiché 0 < t < 1 si ha T<1, e quindi  $\lim_{k\to \infty} T^{k+1} = 0$ . Ne segue che  $\lim_{k\to \infty} ||r^{(k)}||=0$ , e perciò l'algoritmo converge per qualsiasi scelta delle condizioni iniziali.

# 3. Interpretazione fisica del procedimento di Southwell: luce emessa nell'ambiente

E' interessante interpretare i vari passi del metodo di Southwell quando esso viene applicato al problema della radiosità e vedere il significato fisico del rilassamento.

Per comodità, iniziamo il processo iterativo di Southwell assegnando al vettore approssimante valore zero (cioè tutte le componenti uguali a zero). In questo caso, in base a (13), il vettore iniziale dei resti è uguale al vettore dell'energia creata dall'elemento. La prima iterazione seleziona quindi la componente che corrisponde all'elemento che emette più luce propria. Questo primo passaggio porta ad una nuova stima dell'energia dell'elemento, calcolata in modo che il valore del resto diventi nullo per quell'elemento.

Assumendo per semplicità che tutti i fattori di forma da un elemento su sé stesso siano nulli, se indichiamo con i l'indice dell'elemento che viene rilassata durante l'iterazione k+1 (cioè quella con il resto più grande dopo l'iterazione k), la variazione del resto per ciascun elemento può essere espressa grazie a (19), (10) e (9) come

$$r_{j}^{(k+1)} = r_{j}^{(k)} - (K_{ji} / K_{ii}) r_{i}^{(k)} = r_{j}^{(k)} + \rho_{j} F_{i-j} r_{i}^{(k)}$$

dove j=1,...,n, e i è l'indice dell'elemento di massima energia residua  $r_i^{(k)}$  dopo l'iterazione k.

In altre parole, all'iterazione k+1, il resto  $r_i^{(k)}$  dopo l'iterazione *k* viene distribuito fra tutti gli altri elementi in funzione dell'appropriato fattore di forma e del coefficiente di riflettività. Poiché questi resti sono differenze di energia, ogni iterazione modella il trasferimento di energia da un elemento all'ambiente, dovuto ad un'ulteriore riflessione della luce dai vari elementi: all'iterazione k+1 l'elemento i di massimo resto, ossia di massima energia immagazzinata, si libera della propria energia distribuendola nell'ambiente in proporzione ai fattori di forma. Come conseguenza di queste iterazioni di irraggiamento di energia, ciascuna degli altri elementi riceve e accumula energia luminosa, fino a che non viene il suo turno di emetterla: il suo turno arriva quando l'energia accumulata diventa così elevata da far sì che quel particolare elemento sia quello fra tutti con la massima quantità di energia luminosa immagazzinata.

# VERSO UN RAFFINAMENTO PROGRESSIVO

La mole di calcoli dell'algoritmo di radiosità, se eseguito così come presentato fino a questo momento, è molto elevata. Sembra naturale domandarsi se è possibile procedere per approssimazioni successive, creando una prima immagine non accurata e migliorandola in seguito mediante un algoritmo incrementale. Con la definizione attuale di radiosità questo procedimento non è realizzabile nel metodo di Southwell. Infatti i processi di iterazione richiedono che la nuova luminosità dell'elemento sia calcolata sommando tutte le energie entranti provenienti dagli altri elementi: non basta tener conto dell'emissione da parte di un solo elemento. Quindi, prima di fornire una stima della radiosità, deve essere eseguito l'intero ciclo di iterazione. Se ci sono n elementi, ogni ciclo di iterazione richiede la moltiplicazione di una matrice n x n con un vettore colonna di dimensione n, un procedimento che richiede n<sup>2</sup> moltiplicazioni, e quindi un tempo di esecuzione dell'ordine di O(n<sup>2</sup>); inoltre i fattori di forma tra tutte gli elementi devono essere precalcolati all'inizio ed essere memorizzati prima dell'esecuzione, e questo richiede un tempo e uno spazio di memoria dell'ordine di  $O(n^2)$ , perché ci sono n elementi.

In [M.F.Cohen, S.E.Chen, J.R.Wallace e D.P.Greenberg, A Progressive Refinement Approach to Fast Radiosity Image Generation, SIGGRAPH '89, in Computer Graphics, 23(3), luglio 1989] viene sviluppato un algoritmo per il raffinamento progressivo del calcolo della radiosità, senza la necessità di precalcolare e memorizzare tutti i fattori di forma: il sistema lineare viene risolto in modo progressivo, una riga per volta. Nel modello classico il calcolo della radiosità dell'elemento i-esimo viene basato sulla stima dei contributi di energia ricevuti dagli altri elementi. Ogni termine della i-sima equazione del sistema della radiosità esprime l'effetto sulla radiosità dell'elemento i dovuta all'emissione dell'elemento j: per tutti i valori di j da 1 a n, l'energia (o meglio potenza) contribuita all'elemento i dalla radiosità b<sub>i</sub> è r<sub>i</sub> bj F<sub>i-j</sub>

Riassumendo, i metodi iterativi calcolano l'incremento di radiosità di un elemento grazie all'energia luminosa che esso raccoglie dagli altri. Abbiamo già osservato che tale procedimento è troppo oneroso.

# EMISSIONE DI ENERGIA NELL'AMBIENTE INVECE DI ASSORBIMENTO

Una revisione del procedimento, che porta ad un raffinamento progressivo molto meno oneroso dell'immagine generata iterazione dopo iterazione, consiste nel considerare la radiosità emanata da un elemento verso il resto dell'ambiente, piuttosto che quella ricevuta: scambiando i ruoli di i e j rispetto a prima (sezione *Equazione della radiosità*), si ottiene che il contributo alla radiosità b<sub>j</sub> dell'elemento j dovuto all'emissione di radiosità dell'elemento i è  $\rho_i$  b<sub>i</sub> F<sub>j-i</sub>.

Sfortunatamente questo procedimento richiede la stessa mole di calcoli e di occupazione di memoria di quello precedente: fissato l'elemento i, esso richiede la conoscenza del fattore di forma  $F_{j\cdot i}$  dall'elemento j all'elemento i, ed al variare di j ciascuno degli  $F_{j\cdot i}$  richiede un calcolo diverso, fatto su un diverso semicubo, quindi non con lo stesso z-buffer. Però qui possiamo utilizzare la relazione di reciprocità  $A_i F_{i\cdot j} = A_j F_{j\cdot i}$  fra i fattori di forma, e quindi scrivere il contributo alla radiosità  $b_j$  dovuto a  $b_i$  (cioè la frazione di energia emessa dall'elemento i verso l'elemento j e da questo riflessa indietro verso il resto dell'ambiente) come

$$\rho_j b_i F_{i-j} A_i / A_j$$

Quindi, nel corso del k-simo ciclo di iterazione, la i-sima riga del sistema lineare della radiosità rappresenta l'emissione di energia lanciata nell'ambiente nel corso di quella iterazione dall'elemento i verso ciascuno degli altri elementi (incluso sé stesso se  $F_{i-i}$  è non nullo).

# VANTAGGI NUMERICI DELL'APPROCCIO A EMISSIONE DI ENERGIA NELL'AMBIENTE

Il calcolo degli addendi a secondo membro di quest'ultima equazione al variare di j richiede solo la conoscenza dei fattori di forma F<sub>i-j</sub> calcolati usando un unico semicubo centrato al centro dell'elemento i-esimo. Se i fattori di forma dall'elemento i possono essere calcolati velocemente (ad esempio utilizzando uno z-buffer hardware), allora il tempo necessario per questo calcolo è breve; inoltre la memoria occupata da F<sub>i-j</sub> può essere rilasciata non appena la radiosità emanata dall'elemento i verso l'elemento j viene lanciata nell'ambiente (cioè appena calcolato l'addendo j-simo). In questo modo, ad ogni passo dobbiamo avere a disposizione memoria solo per un singolo semicubo e per i suoi fattori di forma. L'algoritmo, la cui convergenza è stata dimostrata nel Teorema 2 (dimostrato nella sottosezione *Convergenza del metodo di Gauss-Seidel*), viene iterato fino a quando non si raggiunge una tolleranza prefissata.

Inizialmente la maggior parte dell'energia luminosa è concentrata in pochi elementi (le sorgenti luminose). Se il vettore di partenza b<sup>(0)</sup> dell'iterazione si pone uquale a 0, al primo passo avremo il primo iterato b<sup>(1)</sup> uquale al vettore e dell'energia creata dalle sorgenti. Al passo successivo dobbiamo cominciare ad aggiungere gli addendi della somma, che rappresentano, per ciascun elemento i, l'energia addizionale emessa dall'elemento i verso ciascuno degli altri e da questi riflessa indietro. È chiaro che la convergenza diventa un pò più rapida se sviluppiamo il calcolo dapprima per la sorgente che ha più energia da emettere, e così via: questo significa che al primo passo ordiniamo gli elementi che sono sorgenti di luce in base alla luminosità (le altre, che non emettono luce propria, al primo passo non intervengono). Poi, via via, ad ogni ciclo di iterazione successivo scegliamo analogamente gli elementi da rilassare ordinandoli in base alla luminosità. In altre parole, non procediamo secondo l'ordine progressivo degli elementi, per i da 1 a n, bensì scegliamo volta per volta l'elemento di massima energia. Questo significa utilizzare, invece del metodo di Gauss-Seidel, il metodo di Southwell che è più efficiente.

# STIMA DELLA RADIOSITÀ RESIDUA DA EMETTERE

Nella sezione *Rilassamento di Southwell* abbiamo studiato il vettore  $\beta$  dell'energia irradiata e le sue iterazioni  $\beta^{(k)}$  nella soluzione iterativa del sistema lineare (9) dello scambio di energia fra gli elementi. Supponiamo che l'elemento con maggiore energia dopo la k-sima iterazione (cioè quella che nel corso dell'iterazione k+1 rilascia la sua energia accumulata) sia l'elemento i-esimo, e per semplicità facciamo l'ipotesi che non rifletta luce su sé stesso, cioè che il fattore di forma F<sub>i-i</sub> sia zero. Allora, per (12), il corrispondente termine diagonale della matrice K del sistema lineare dello scambio di energia vale K<sub>ii</sub> = M<sub>ii</sub> = 1, e quindi, nel ciclo di iterazione k+1, l'incremento del vettore  $\beta$  dell'energia totale irradiata è dato dall'equazione (17):

$$\beta_i^{(k+1)} = \beta_i^{(k)} + r_i^{(k)}$$
 (17 bis)

e, per j diverso da i, dall'equazione (15):

$$\beta_j^{(k+1)} = \beta_j^{(k)}.$$

Corrispondentemente si ha  $r_j^{(k+1)} = 0$  (grazie a (14): rilassamento dell'elemento i) e, per j diverso da i,

$$r_{j}^{(k+1)} = r_{j}^{(k)} - K_{ji} r_{i}^{(k)} = r_{j}^{(k)} + \rho_{j} F_{i \cdot j} r_{i}^{(k)}$$
 (29)

grazie all'equazione (19).

L'energia totale irradiata dall'elemento j al passo di iterazione k+1 è quindi la stessa che al passo precedente ( $\beta_j^{(k+1)} = \beta_j^{(k)}$ ), ma c'è una quantità  $r_j^{(k+1)} - r_j^{(k)}$  di energia addizionale pervenuta all'elemento in conseguenza del rilascio, in quella iterazione, dell'energia immagazzinata nell'elemento i fino alla iterazione precedente. Perciò è opportuno introdurre un nuovo vettore  $\beta'$  dell'energia totale già irradiata da ciascun elemento o in attesa di esserlo al successivo rilassamento:

$$\beta_{j}^{(k)} = \beta_{j}^{(k)} + r_{j}^{(k)}, \qquad j = 1,..., n$$
 (30).

Questo vettore, un passo di iterazione dopo l'altro, dà una stima progressivamente più precisa dell'energia totale che verrà irradiata dagli elementi al termine del processo iterativo.

Segue ora da (30), (15) e (29) che, per ogni j diverso da i,

$$\beta_{j}^{'(k+1)} = \beta_{j}^{(k)} + r_{j}^{(k+1)} = \beta_{j}^{'(k)} + r_{j}^{(k+1)} - r_{j}^{(k)} = \beta_{j}^{'(k)} + \rho_{j} F_{i-j} r_{i}^{(k)}$$
(31).

Ora ritorniamo alle radiosità, che sono le energie irradiate divise per le aree dei rispettivi elementi. Vogliamo una stima iterativa  $b_i^{(k)}$  della radiosità finale  $b_i$ , per ogni elemento i. Essa è la somma della radiosità effettivamente rilasciata da quell'elemento fino al ciclo di iterazione k, che in base ad (8') è data da

$$b_i^{(k)} = \beta_i^{(k)} / A_i$$

e della radiosità immagazzinata ed ancora in attesa di essere rilasciata,

$$\Delta b_i^{(k)} = r_i^{(k)} / A_i$$

Pertanto,

$$b_{i}^{'(k)} = b_{i}^{(k)} + \Delta b_{i}^{(k)} = (\beta_{i}^{(k)} + r_{i}^{(k)}) / A_{i} = \beta_{i}^{'(k)} / A_{i}$$
(32).

Ma poiché  $b_i^{(k)}$  è una stima via via più precisa della radiosità finale  $b_i$ , ora  $\Delta b_i^{(k)}$  è una stima della quantità residua di radiosità che all'elemento i resta da irradiare dopo il passo k. A causa di questa radiosità residua, le immagini ottenute durante i primi passi di iterazione sono scure, ed a poco a poco diventano più chiare avvicinandosi all'immagine finale. Purtroppo, però, questa stima  $b_i^{(k)}$  tiene sì conto dell'energia residua  $\Delta b_i^{(k)} = r_i^{(k)}/A_i$  immagazzinata nell'elemento i al passo k di iterazione, ma in modo impreciso: al termine del ciclo iterativo il residuo di energia immagazzinata non contribuisce solo alla radiosità dell'elemento i che la immagazzinava, ma anche a quella di tutti gli altri, in seguito alle ulteriori emissioni e ricezioni ed in proporzione alle loro riflettività ed ai fattori di forma. Occorre quindi una stima più accurata che tenga conto di questa distribuzione.
## CORREZIONE AMBIENTALE DELLA LUMINOSITÀ NEL CORSO DELLE ITERAZIONI

Se l'algoritmo di radiosità è formulato in termini dell'emissione di energia (o radiosità) dagli elementi, invece che della loro ricezione di energia dagli altri elementi, e l'approssimazione iniziale della soluzione è il vettore nullo, allora la sola illuminazione dopo il primo ciclo di iterazione è quella proveniente direttamente dalle sorgenti di luce. Nelle iterazioni successive si considerano una riflessione, due riflessioni e così via. Chiaramente, nelle immagini ottenute dalle prime fasi di iterazione manca l'effetto di molteplici riflessioni successive, e quindi queste immagini sono scure.

Vogliamo usare la stima (32) per determinare un termine di correzione ambientale da aggiungere alla radiosità di tutte gli elementi per compensare la bassa luminosità delle immagini iniziali: grazie ad esso si possono ottenere approssimazioni iterative dell'immagine finale già buone dopo non molte iterazioni, e quindi si velocizza considerevolmente il metodo della radiosità. Come accennato prima, questo algoritmo è stato introdotto in [M.F.Cohen, S.E.Chen, J.R.Wallace e D.P.Greenberg, <u>A Progressive Refinement Approach to Fast Radiosity Image Generation</u>, SIGGRAPH '89, in Computer Graphics, 23(3), luglio 1989].

Poiché questa correzione deve basarsi su una stima dell'effetto delle riflessioni successive, dobbiamo calcolarla stimando il coefficiente di riflettività media della scena, che si può definire come la media dei coefficienti di riflettività  $\rho_i$  degli elementi, pesata rispetto alla proporzione dell'area del singolo elemento rispetto all'area totale:

$$\rho_{\text{media}} = \sum_{i=1,\dots,n} \rho_i A_i / \sum_{i=1,\dots,N} A_i$$
(33)

Cosa si troverebbe se si cercasse una stima della luminosità ambientale residua già al passo iniziale di iterazione? A quel passo l'energia inviata all'ambiente per ogni elemento è data dal vettore e (la luce propria emessa dalle sorgenti). Allora, dopo la prima riflessione avremmo un contributo addizionale  $\rho_{media}$ e, dopo la seconda un ulteriore contributo ( $\rho_{media}$ )<sup>2</sup> e, e così via: l'incremento di luminosità è quindi Re, dove il numero R è il fattore di interriflessione globale

$$R = 1 + \rho_{media} + (\rho_{media})^2 + (\rho_{media})^3 + ... = 1/(1 - \rho_{media})$$
(34)

Osserviamo che 0 <  $\rho_{media}$  < 1, e quindi R > 1. L'incremento di luminosità dà quindi l'impressione falsa che l'energia non si conservi: ma in realtà qui stiamo misurando non l'energia ma la radiosità, ovvero la densità di potenza per unità di area: si parla quindi del flusso temporale di densità di energia. Questi effetti di maggiore luminosità sono dovuti a riflessioni successive della luce, che avvengono in istanti successivi del tempo: in ciascun istante l'energia totale rimane sempre quella iniziale, ma il fattore di correzione, che amplifica indebitamente questa energia, avrebbe l'effetto di ridistribuirla creando una illuminazione di fondo più precisa, se applicassimo il ragionamento non alla prima iterazione ma via via a quelle successive. Vediamo come.

Una stima dell'energia totale H ancora non irradiata nell'ambiente dopo la k-sima iterazione (quindi dopo k interriflessioni) è la somma delle energie residue immagazzinate dagli elementi:

$$H = \sum_{i=1,...,n} r_i^{(k)}$$
(35)

Ma a noi serve la correzione ambientale delle radiosità, non delle energie. Perciò dobbiamo utilizzare il flusso per unità di area di questa energia totale non irradiata. L'area totale dell'ambiente è la somma A delle aree degli elementi:

$$\mathsf{A} = \sum_{i=1,\dots,n} \mathsf{A}_i \, .$$

Quindi il flusso per unità di area dell'energia totale non irradiata è

$$\Phi = \mathsf{RH/A} \tag{36}$$

Questo flusso ambientale, effetto globale di tutte le interriflessioni successive, colpisce tutte gli elementi, e le illumina in proporzione al loro coefficiente di riflettività: pertanto al termine della k-sima iterazione la stima della radiosità con incremento ambientale è data da

$$b_{i}^{(\prime)} = b_{i}^{(k)} + \rho_{i} \Phi$$
 (37)

Al crescere di k,  $r_i^{(k)}$  tende a 0 per ogni i = 1,...n, e quindi H tende a 0. Pertanto  $\Phi$  tende a 0, e l'effetto di correzione si attenua progressivamente col procedere delle iterazioni. Esso è intenso solo all'inizio, e permette di simulare con buona approssimazione quanto devono essere schiarite le immagini approssimate iniziali perchè la loro luminosità totale (ma ovviamente non la loro distribuzione di luce) sia accettabile.

## ACCURATEZZA NEL CALCOLO DEI FATTORI DI FORMA, USO DEL RAY TRACING E MAPPA DI OCCLUSIONE

Nonostante l'utilizzo di uno z-buffer hardware renda il metodo di calcolo dei fattori di forma basato sulla proiezione sull semicubo un algoritmo estremamente efficiente per quanto riguarda il tempo di calcolo, la stessa cosa non si puo dire per quanto riguarda l'accuratezza.

Ecco alcuni problemi:

1 – A causa del modo in cui funziona lo z-buffer, ogni pixel del semicubo viene associato ad un solo elemento, anche nel caso in cui in quel pixel siano parzialmente visibili due o più elementi. Questo può portare a fenomeni di aliasing fra gli elementi che si sovrappongono quando proiettate su un lato del semicubo, ed alla scomparsa di parti di elementi o all'apparire di una alternanza a mosaico di due elementi lungo le linee sul buffer dove si intersecano due elementi diversi. Questi errori sono confinati ad aree grandi quanto un pixel, quindi piccole, ma queste aree sono piccole *solo per il calcolo dei fattori di forma*: la loro visibilità dal punto di vista dell'osservatore potrebbe essere elevata, ossia esse potrebbero coprire zone grandi nell'immagine resa.

2 - Inoltre, si assume che il punto centrale di un elemento sia rappresentativo della ubicazione e della tessitura e shading dell'intero elemento quando visto dagli altri. Nei casi in cui ciò non è vero l'elemento può essere suddiviso in sottoelementi, ma questo si fa una volta per tutte, e quindi una volta sola: se il punto centrale non è adeguato quando osservato da un altro elemento si procede a suddividere, ma a quel punto, se esso rimane inadeguato anche dopo tale suddivisione quando osservato da un terzo elemento, non si può risuddividere perché ormai i sottoelementi sono stati già utilizzati nel calcolo, ed occorrerebbe ripetere il calcolo da capo con una sottostrutturazione ancora più fine.

3 – Infine osserviamo che, affinché l'intero procedimento basato sul semicubo dia un risultato corretto, gli elementi devono essere lontani fra di loro. La stima dei fattori di forma è approssimata per difetto per elementi vicini o adiacenti: sulle zone di contatto il calcolo dei fattori di forma precalcolati per i pixel dell'emicubo è sottostimato, perché è relativo al centro del pixel (dove i due elementi sono un po' più lontani), e non al punto di contatto che corrisponde invece di solito al bordo del pixel.

Il calcolo progressivo della radiosità è stato anche effettuato mediante l'impiego, nella valutazione dei fattori di forma, del Ray Tracing (non ricorsivo) al posto della proiezione sul semicubo [J.R.Wallace, K.A.Elmquist e E.A.Haines, <u>A Ray Tracing Algorithm for Progressive Radiosity</u>, SIGGRAPH 89, in Computer Graphics 23(3), luglio 1989, pagg. 315-324]. Questo si fa nel modo seguente. Quando un elemento diffonde nell'ambiente la propria radiosità, per ogni vertice della scena vengono tracciati raggi verso l'elemento. Si suddivide l'elemento in un numero finito di sottoelementi, ognuno dei quali è il bersaglio di un raggio uscente dal vertice. Se il raggio non viene interrotto lungo il suo cammino, allora il sottoelemento dell'elemento è visibile e viene quindi calcolato, in modo analitico, il fattore di forma "infinitesimo" tra il vertice ed il sottoelemento, grazie

alla solita formula  $F_{di-j} = \int_{A_j} \frac{\cos \theta_i \cos \theta_j}{\pi r^2} H_{ij} dA_j$ , oppure lo stesso fattore di forma

viene calcolato in modo numerico o statistico. A questo scopo si tracciano raggi a partire dal vertice, attraverso un viewport (o meglio un emisfero o un emicubo) suddiviso in pixel. I raggi sono centrati al centro dei vari pixel oppure distribuiti aleatoriamente (quale sia la migliore scelta di distribuzione di probabilità lo vedremo nel terzo volume di quest'opera, dove presenteremo metodi probabilistici di illuminazione globale). Essi vengono tracciati e si conta quanti di essi colpiscono il sottoelemento: la frazione di quelli che lo colpiscono misura la proporzione dell'angolo solido da esso sotteso rispetto a quel vertice, e quindi il fattore di forma infinitesimo del sottoelemento rispetto al vertice.

A questo punto il fattore di forma tra il vertice e l'intero elemento viene calcolato come media pesata dei fattori di forma infinitesimi dei sottoelementi visti dai vertici. In tal modo si ottengono direttamente i fattori di forma degli elementi visti dai vertici, il che è un vantaggio perché evita la fase di postprocessing nella quale le radiosità delle facce vengono trasformate in radiosità dei vertici: qui si ottengono direttamente le radiosità dei vertici (si veda l'articolo [M.F.Cohen e D.P.Greenberg, <u>The Hemi-Cube: A Radiosity Solution for Complex Environments</u>, SIGGRAPH 85, in Computer Graphics 19(3), luglio 1985, pagg. 21-40] già citato prima, alla fine della sezione "L'equazione della radiosità").

Si noti che il fattore di forma di un elemento visto da un punto è una misura dell'occlusione frapposta dall'elemento all'illuminazione del punto dovuta al resto dell'ambiente, ossia del suo livello di ombra. In questo senso, il calcolo del fattore di forma tramite Ray Tracing dei vari elementi rispetto ad un punto equivale alla *mappa di occlusione* che ora definiamo e spieghiamo, a costo di una breve digressione. Consideriamo un punto, ad esempio appartenente ad una faccia piana di un elemento, ed il semispazio ad esso frontale. La frazione di angolo solido coperta dagli altri elementi, misurata attraverso il numero di hits di raggi

proiettori uscenti dal punto, misura l'intensità di ombra a quel punto come conseguenza dell'occlusione della luce dovuta agli altri elementi. Essa si può reinterpretare nel modo seguente. Supponiamo dapprima che il punto stia su un elemento da illuminare con luce puramente ambientale, non riflettente né diffusiva secondo il modello di Lambert. Immaginiamo che il semispazio frontale sia racchiuso in un emisfero uniformemente illuminato e non riflettente, che funge da sfondo luminoso, ossia da sorgente di luce diffusa. Emaniamo, dal punto in oggetto, un gran numero di raggi proiettori, approssimativamente angolarmente equidistribuiti nell'emisfero, e tracciamoli nella scena finche colpiscono un altro elemento o l'emisfero. Ciascun raggio proiettore che non viene bloccato dagli altri elementi colpisce l'emisfero, e, nel consueto meccanismo del Ray Tracing, ogni hit accumula un contributo di energia luminosa. La somma di guesti contributi è l'illuminazione del punto causata dallo sfondo luminoso: se si preferisce, è la luce ambientale che illumina il punto, detratta però di quella bloccata dagli altri elementi, ossia attenuata a causa dell'ombra che esse proiettano. In tal modo la mappa di occlusione definisce ombre, e quindi differisce dal modello banale della luce ambientale perché ha aspetti direzionali: tiene conto delle direzioni verso cui il punto vede gli altri elementi.

Se invece si intende simulare l'illuminazione, e quindi il livello di ombra, di un punto che giace in un elemento diffusivo secondo Lambert, è ancora possibile calcolare l'occlusione della scena mediante Ray Tracing. In questo caso dobbiamo immaginare che l'emisfero uniformemente luminoso si scomponga in tante piccole sorgenti di luce puntuali che emettono luce isotropicamente, tutte alla stessa intensità. In tal caso, ogni raggio proiettore, in base al modello di Lambert, colpisce l'emisfero in una delle piccole sorgenti di luce, e fornisce un contributo di illuminazione proporzionale al coseno dell'angolo di deviazione dal polo Nord: infatti questo è esattamente l'angolo fra la direzione della piccola sorgente di luce verso cui punta il raggio proiettore ed il versore normale al punto di emanazione, che è nient'altro che il versore verticale, ossia diretto verso il polo Nord. Pertanto basta, in questo caso, emanare raggi proiettori non più equidistribuiti, ma pesati secondo il coseno dell'angolo di deviazione dal polo Nord. Questo meccanismo pesa diversamente le sorgenti puntiformi di cui idealmente si compone l'emisfero. Vengono pesate di più le aree dell'emisfero frontali rispetto al punto di emanazione, cioè quelle in un intorno del polo Nord (dove la deviazione è zero ed il coseno vale 1), le quali inducono una illuminazione più intensa proprio perché frontali, in perfetto accordo con la legge empirica di Lambert. Queste aree vengono campionate maggiormente perché un maggior numero di raggi vengono emessi in tali direzioni, e quindi contribuiscono di più all'illuminazione del punto di emissione, naturalmente se esse non sono occluse.

Infine, se il punto giace in un elemento che oltre ad essere diffusivo è anche parzialmente speculare, e la scena è illuminata non isotropicamente, l'illuminazione e l'ombra si possono ancora calcolare come sopra. Supponiamo che la distribuzione di luci della scena sia tale che, ad esempio, ci sia una direzione dalla quale proviene principalmente la luce, mentre nelle altre direzioni l'emisfero è relativamente buio. Allora i raggi devono essere distribuiti in maniera

da privilegiare la direzione di provenienza della luce: la distribuzione angolare dei raggi deve essere più fitta in tale direzione, e deve decrescere come il coseno dell'angolo di deviazione.

Appofittiamo di queste osservazioni per riformularle in un senso che diventerà cruciale quando, nel terzo volume di quest'opera, lo riconsidereremo nell'ambito dell'illuminazione globale. Abbiamo già osservato che il calcolo dei fattori di forma, ovvero della mappa di occlusione, tramite Ray Tracing richiedono, come tipico del Ray Tracing, che vengano emessi molti raggi. C'è però un modo di evitare questo proliferare di raggi da emettere con una distribuzione che può essere anisotropa: invece di emettere molti raggi deterministicamente, ne emettiamo pochi ma stocasticamente, ovvero con una distribuzione di probabilità corrispondente alla distribuzione angolare voluta. Il senso di questa corrispondenza ed il modo in cui viene realizzata la distribuzione di probabilità vengono studiati nel terzo volume, nel capitolo sui metodi probabilistici per il calcolo di integrali e la loro applicazione all'illuminazione globale.

## **RIFLETTIVITA' BIDIREZIONALE**

Il metodo di radiosità sviluppato fino a questo momento tratta solo la riflessione diffusa. In tal modo, la radiosità uscente da un elemento è influenzata dalla radiosità totale e non dalle direzioni attraverso cui viene ricevuta energia. L'articolo [D.S.Immel, M.F.Cohen e D.P.Greenberg, A Radiosity Method for Non-Diffuse Environments, SIGGRAPH 86, in Computer Graphics 20(4), agosto 1986, pagg. 133-142] estende il meccanismo di radiosità per modellare anche la riflessione speculare. Invece di calcolare un singolo valore di radiosità per ogni elemento, viene calcolato un valore per ogni direzione. Più precisamente, si partiziona l'emisfero sopra l'elemento in un insieme finito di angoli solidi, ognuno dei quali individua una direzione per l'energia entrante o uscente (ovviamente, quindi, si limita l'attenzione ad un numero finito di direzioni: il problema viene discretizzato). Tramite il coefficiente di riflettività bidirezionale (si vedano le sezioni sui Modelli fisici di illuminazione e sulla Equazione integrale del rendering) ora si può calcolare la radiosità uscente in ogni direzione come somma dei contributi di radiosità entranti dalle varie direzioni (discretizzate, quindi in numero finito) più la radiosità creata in quel dato elemento ed emessa verso quella data direzione. L'immagine infine viene resa interpolando le intensità determinate per ogni vertice tramite interpolazione della radiosità direzionali uscenti dall'elemento a cui il vertice appartiene, interpolate a partire da quelle delle direzioni discretizzate più vicine alla direzione dal vertice all'osservatore. Questo tipo di approccio però porta ad una mole di calcoli estremamente onerosa.

## METODI DI RENDERING A DUE PASSI CHE COMBINANO RADIOSITÀ E RAY TRACING RICORSIVO

L'algoritmo di radiosità si adatta bene alla riflessione diffusa, ma non a quella speculare, poiché l'intensità della riflessione speculare dipende dall'angolo con il quale l'osservatore guarda la superficie, ma questo angolo non viene considerato nell'algoritmo di radiosità, che non è direzionale. D'altra parte il procedimento di ray tracing è ideale per il calcolo della riflessione speculare, ma non per la diffusione globale dell'ambiente. (Per maggiori dettagli, si veda la sezione sulla *Equazione integrale del rendering*)

È quindi opportune trovare il modo di combinare i due metodi in modo da sfruttare i vantaggi di entrambi. Purtroppo però non è sufficiente applicare entrambi i metodi e poi sommare i risultati, perché le componenti della luminosità dovute alla riflessione speculare contribuiscono anche alla illuminazione diffusa e viceversa.

Nell'articolo [J.R.Wallace, M.F.Cohen e D.P.Greenberg, A Two-pass Solution to the Rendering Equation: a Synthesis of Ray Tracing and Radiosity Methods, SIGGRAPH '87, in Computer Graphics, 21(4), luglio 1987, pagg. 311-320], viene sviluppato un approccio a due passi che combina il metodo di radiosità (indipendente dall'angolo di visuale) con quello di ray tracing, che invece ne dipende. Il primo passo, che non dipende dalla direzione di osservazione, consiste di un metodo ampliato di radiosità che tiene conto anche della riflessione speculare, perché amplia in maniera virtuale la scena, come se gli elementi riflettenti fossero finestre su un "mondo speculare". I fattori di forma da un elemento generica ad uno di questi elementi virtuali riflessi dietro un elemento di "specchio" rappresentano il contributo alla riflessione globale da parte della riflessione speculare dello specchio. Purtroppo, se le superficie speculari sono ampie, questa fase aumenta quasi di un fattore due i fattori di forma da calcolare, e quindi rende quattro volte più lento il tempo di calcolo.

Nel secondo passo, dipendente dal punto di visuale, si considera ogni punto visibile di ogni elemento. Per ogni sorgente di luce si considera la direzione di riflessione speculare a questo punto, e si costruisce, anziché il raggio riflesso del ray tracing, una piramide *di riflessione*, cioè uscente in tale direzione, di apertura piccola, perché limitata all'angolo solido che include solo le deviazioni dalla direzione di riflessione speculare per le quali l'intensità riflessa è sufficientemente elevata (per determinare quanto elevata sia l'intensità in ciascuna direzione uscente si calcola il coefficiente di riflettività bidirezionale: ancora una volta, si vedano le sezioni sui *Modelli fisici di illuminazione* e sulla *Equazione integrale ricorsiva del rendering*, nonché la sezione precedente). Sul piano perpendicolare all'asse della piramide (cioè alla direzione di riflessione speculare) si considera

un piccolo rettangolo che viene usato per eseguire uno z-buffer di piccole dimensioni: così si determina quali altri elementi siano visibili attraverso ciascun pixel, e di ciascuna di esse si considera l'illuminazione diffusa calcolata nella prima fase tramite la radiosità estesa (la radiosità è stata riportata ai vertici della scena, ed ora la si riporta ai punti visti attraverso i centri dei pixel mediante interpolazione di Gouraud) Se c'è più di un elemento visibile attraverso la piramide, si calcola il contributo totale risultante tramite interpolazione (ossia combinazione pesata in termini alla proporzione delle aree visibili). Inoltre si tiene conto se la piramide attraverso il pixel vede altri elementi speculari: per ciascuno di essi ripetiamo il procedimento in maniera ricorsiva (per un numero prefissato di generazioni) emettendo un'altra piramide riflessa (esattamente come nel procedimento di generazione iterativa di raggi riflessi del ray tracing ricorsivo). Si totalizzano i contributi di illuminazione ottenuti, come spiegato prima, da ognuna delle piramidi così generate, ed in tal modo si ottiene il contributo della riflessione speculare al vertice della piramide iniziale. Anche il calcolo di questa fase, ovviamente, è piuttosto oneroso.



Nella figura qui sopra è mostrata una piramide di riflessione. Riassumiamo il procedimento. Quale sia per ogni pixel l'elemento speculare visibile si calcola col metodo di z-buffer; si utilizza l'interpolazione di Gouraud per calcolare per quel pixel le intensità di riflessione diffusa del primo passo. Se l'elemento visibile ad un dato pixel di questo z-buffer è un elemento che dà luogo a riflessione speculare, allora si traccia il raggio dal vertice della piramide attraverso il centro del pixel e si determina l'angolo di riflessione speculare al punto in cui colpisce l'elemento: continuando iterativamente così si disegnano nuove piramidi in corrispondenza di ogni intersezione. Attraverso una media dei valori calcolati per ogni pixel dello z-buffer di ciascuna piramide si trova l'intensità della luce riflessa dell'elemento di partenza: il suo contributo nella direzione dell'osservatore si calcola infine grazie all'equazione di illuminazione di Phong o a modelli fisici.

Questo metodo riesce a combinare la radiosità col ray tracing ricorsivo, ma porta ad una proliferazione dei fattori di forma, il cui calcolo è lento. Abbiamo già osservato che, se tutte gli elementi riflettono luce anche specularmente, contando anche gli elementi virtuali il loro numero si raddoppia ed il numero dei fattori di forma si quadruplica. Una variante più efficiente, introdotta in *[F.Sillion, C.Puech, <u>A General Two-Pass Method Integrating Specular and Diffuse</u> <i>Reflection, SIGGRAPH '89,* in *Computer Graphics, 23(3), luglio 1989]* calcola fattori di forma estesi, i quali sono in grado di tener conto di un numero arbitrario di riflessioni speculari o rifrazioni. Questi fattori di forma estesi si calcolano mediante Ray Tracing ricorsivo fin dal primo passo, invece che mediante semicubi e z-buffer come nel meccanismo usuale della radiosità, oppure mediante Ray Tracing non ricorsivo come nel caso della <u>mappa di occlusione</u> vista prima.

## ESEMPIO 1: DISTRIBUZIONE DELLA LUCE ENTRO UNA TENDA PIRAMIDALE

La scena consiste dell'interno di una tenda, cioè una stanza a forma di piramide equilatera, quindi con quattro facce identiche a forma di triangolo equilatero. La faccia di base emette luce con intensità 1 (possiamo pensare che ci sia un fuoco acceso al centro della base: in tal caso la luce viene emessa al centro, ma nella presente approssimazione, peraltro alquanto rudimentale, l'intero triangolo di base consiste di un unico elemento, che indichiamo col numero 1). Le altre facce non emettono luce propria, solo riflessa. La base, cioè il pavimento, è di terra scura, e l'approssimiamo come una superficie non riflettente, cioè  $\rho_1 = 0$ . Le pareti sono grigio chiaro, e quindi poniamo  $\rho_i = \frac{1}{2}$  per i=2, 3, 4. Per comodità scriviamo  $\rho$  invece di  $\frac{1}{2}$ . Quindi il vettore dell'energia propria emessa è e =(1,0,0,0), mentre il vettore della riflettività è  $\rho = (0, 1/2, 1/2, 1/2)$ .

Però in questo caso gli elementi sono piane, e quindi i fattori di forma diagonali, F<sub>i-i</sub>, sono tutti nulli. D'altra parte, la tenda è completamente simmetrica: ogni faccia copre, vista da ogni altra faccia, lo stesso angolo solido. Perciò, per *i* diverso da *j*, i fattori di forma F<sub>i-i</sub> sono tutti uguali, cioè c'è un unico fattore di forma non nullo che per semplicità chiamiamo F. (Si osservi che, poiché le aree di tutte gli elementies sono uguali, questo risultato conferma anche l'equazione di reciprocità data dalla reversibilità dei percorsi ottici, A<sub>i</sub> F<sub>i-j</sub> = A<sub>j</sub> F<sub>j-i</sub> : questa e le prossime equazioni sono state introdotte nella sezione *L'equazione della radiosità*. Ora, dall'equazione della conservazione dell'energia (o, se si preferisce, di conservazione dell'angolo solido totale)  $\Sigma_j$  F<sub>i-j</sub> =1, si ottiene F=1/3. Pertanto il sistema lineare della radiosità diventa:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ -\rho F & 1 & -\rho F & -\rho F \\ -\rho F & -\rho F & 1 & -\rho F \\ -\rho F & -\rho F & -\rho F & 1 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Procediamo a risolvere il sistema con metodi di rilassamento. Per poter applicare il metodo iterativo formulato in (6') della sezione *Analisi del metodo di Gauss-Seidel* dobbiamo spezzare M come M = N - P, con N matrice triangolare inferiore (o meglio non superiore: la diagonale di N coincide con quella di M) e P triangolare (strettamente) superiore. Quindi:

$$N = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\rho F & 1 & 0 & 0 \\ -\rho F & -\rho F & 1 & 0 \\ -\rho F & -\rho F & -\rho F & 1 \end{pmatrix}$$
$$P = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \rho F & -\rho F \\ 0 & 0 & 0 & \rho F \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

#### 1. Metodo di Jacobi

Per primo, consideriamo il metodo di Jacobi, introdotto nella sezione *Soluzione con metodi iterativi: rilassamento di Jacobi*. In questo metodo, il ciclo di iterazione (equazione (3) della succitata sezione) viene eseguito su tutte le componenti dei vettori, quindi su tutte le *n* equazioni del sistema. Il vettore e vale (1,0,0,0). Per semplicità di calcolo conviene partire con l'approssimante iniziale  $\mathbf{b}^{(0)} = (0,0,0,0)$ . Si ottiene, da (6'):

$$\mathbf{b}^{(k+1)} = \mathbf{N}^{-1} \mathbf{P} \mathbf{b}^{(k)} + \mathbf{N}^{-1} \mathbf{e}$$
(38)

Una matrice triangolare inferiore a dimensione 4, con 1 sulla diagonale, cioè del tipo

$$J = \begin{pmatrix} 1 & 0 & 0 & 0 \\ x & 1 & 0 & 0 \\ y & z & 1 & 0 \\ u & v & w & 1 \end{pmatrix}$$

ha per inversa (esercizio!):

$$J^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -x & 1 & 0 & 0 \\ xz - y & -z & 1 & 0 \\ (y - xz)w + xv - u & zw - v & -w & 1 \end{pmatrix}$$

Perciò

$$N^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \rho F & 1 & 0 & 0 \\ \rho F + (\rho F)^2 & \rho F & 1 & 0 \\ \rho F + 2(\rho F)^2 + (\rho F)^3 & \rho F + (\rho F)^2 & \rho F & 1 \end{pmatrix}$$

Ora possiamo trovare gli approssimanti dati dalle prime iterazioni del metodo di Jacobi, cioè del rilassamento eseguito ricalcolando ad ogni ciclo iterativo tutte le n equazioni del sistema, ciascuna applicata ai dati provenienti dal ciclo precedente. Si trova, da (<u>38</u>):

$$\mathbf{b}^{(1)} = \mathbf{N}^{-1}\mathbf{e}$$
$$\mathbf{b}^{(2)} = \mathbf{N}^{-1}\mathbf{P}\mathbf{b}^{(1)} + \mathbf{N}^{-1}\mathbf{e} = \mathbf{N}^{-1}\mathbf{P}\mathbf{b}^{(1)} + \mathbf{b}^{(1)} = (\mathbf{N}^{-1}(\mathbf{P}+\mathbf{I})\mathbf{b}^{(1)}$$
$$\mathbf{b}^{(3)} = \mathbf{N}^{-1}\mathbf{P}\mathbf{b}^{(2)} + \mathbf{N}^{-1}\mathbf{e} = \mathbf{N}^{-1}\mathbf{P}\mathbf{b}^{(2)} + \mathbf{b}^{(1)}$$

Poiché il vettore e è il primo vettore della base canonica, il risultato dell'applicare ad e una matrice è il vettore dato dalla prima colonna della matrice. Pertanto:

$$b^{(1)} = \begin{pmatrix} 1 \\ \rho F \\ \rho F + (\rho F)^{2} \\ \rho F + 2(\rho F)^{2} + (\rho F)^{3} \end{pmatrix}$$

$$b^{(2)} = \left(N^{-1}P + I\right) \begin{pmatrix} 1 \\ \rho F \\ \rho F + (\rho F)^{2} \\ \rho F + 2(\rho F)^{2} + (\rho F)^{3} \end{pmatrix} = \begin{pmatrix} 1 \\ \rho F + 2(\rho F)^{2} + 3(\rho F)^{3} + (\rho F)^{4} \\ \rho F + 2(\rho F)^{2} + 4(\rho F)^{3} + 4(\rho F)^{4} + (\rho F)^{5} \\ \rho F + 2(\rho F)^{2} + 4(\rho F)^{3} + 7(\rho F)^{4} + 5(\rho F)^{5} + (\rho F)^{6} \end{pmatrix}$$

Nel nostro caso,  $\rho=1/2$  e F=1/3, quindi  $\rho$ F=1/6. Se, per semplicità di calcolo, limitiamo la precisione alla prima cifra decimale, possiamo trascurare, in prima approssimazione, le potenze ( $\rho$ F)<sup>n</sup> con n maggiore di 2, perché ( $\rho$ F)<sup>2</sup> = 0.02778 e ( $\rho$ F)<sup>3</sup> = 0.00077 (si potrebbe trascurare già ( $\rho$ F)<sup>2</sup> se non fosse per il fatto che questo fattore compare moltiplicato per 2 nella terza componente e nella quarta componente del vettore b<sup>(2)</sup>). Quindi, l'approssimazione con la prima cifra decimale esatta di b<sup>(1)</sup> è b<sup>(1)</sup>= (1, 0.17, 0,19, 0.23), mentre b<sup>(2)</sup>= (1, 0.24, 0,24, 0.25). Per b<sup>(3)</sup> si ottiene, ponendo per semplicità t=  $\rho$ F:

$$b^{(3)} = N^{-1}P \begin{pmatrix} 1 \\ t+2t^{2}+3t^{3}+t^{4} \\ t+2t^{2}+4t^{3}+4t^{4}+t^{5} \\ t+2t^{2}+4t^{3}+7t^{4}+5t^{5}+t^{6} \end{pmatrix} + \begin{pmatrix} 1 \\ t \\ t+t^{2} \\ t+2t^{2} + t^{3} \end{pmatrix}$$
$$= \begin{pmatrix} 1 \\ t+2t^{2}+4t^{3}+8t^{4}+11t^{5}+6t^{6}+t^{7} \\ t+2t^{2}+4t^{3}+8t^{4}+15t^{5}+16t^{6}+7t^{7}+t^{8} \\ t+2t^{2}+4t^{3}+8t^{4}+16t^{5}+26t^{6}+22t^{7}+8t^{8}+t^{9} \end{pmatrix} \approx \begin{pmatrix} 1 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix}$$

Si osservi che le componenti dei vettori  $b^{(2)}$  e  $b^{(3)}$  sono termini polinomiali in *t* nei quali i monomi di grado più basso si stabilizzano, cioè mantengono gli stessi coefficienti (lasciamo al lettore, come esercizio, dimostrare che il coefficiente del monomio t<sup>n</sup> in tutte le componenti delle approssimazioni successive si stabilizza sul valore  $2^{n-1}$ ). Poiché alla precisione richiesta solo questi termini contano, per n maggiore di 2 le approssimazioni iterative  $b^{(n)}$  sono costanti: abbiamo raggiunto l'equilibrio, cioè la soluzione b = (1, 0.25, 0.25, 0.25). Si osservi che la prima componente, che corrisponde al pavimento della tenda piramidale, vale sempre 1 perché, pur non riflettendo luce, emette luce propria con potenza 1 per via della fiamma al suo centro. Osserviamo anche che all'equilibrio le tre pareti della tenda ricevono la stessa quantita' di illuminazione: anche questo era previsto, per via della simmetria della scena fra le tre pareti.

E' evidente da questi calcoli che le iterazioni del metodo di Jacobi, in ciascuna delle quali si debbono ricalcolare tutte le equazioni del sistema, sono laboriose.

Presentiamo qui un notebook di <u>Mathematica</u> (software di manipolazione simbolica e numerica) che esegue i calcoli precedenti in maniera automatica:

 $P = \{\{0, 0, 0, 0\}, \{0, 0, t, t\}, \{0, 0, 0, t\}, \{0, 0, 0, 0\}\}$   $NN = \{\{1, 0, 0, 0\}, \{-t, 1, 0, 0\}, \{-t, -t, 1, 0\}, \{-t, -t, -t, 1\}\}$   $e = \{1, 0, 0, 0\}$  Ninv = Inverse[NN]; TraditionalForm[Ninv]  $b = \{0, 0, 0, 0\}$   $For[i=1, i<5, i++, Expand[b = Ninv . P . b + Ninv . e]; N[b] /. t \rightarrow 1/6]$ 

Il lettore che non disponga di Mathematica può facilmente convertire al linguaggio C il precedente notebook e quelli che presenteremo in seguito; però poi, per gli integrali numerici necessari per i fattori di forma, deve usare le sofisticate librerie matematiche disponibili in C.

#### 2. Metodo di Gauss-Seidel

Trattiamo ora lo stesso esempio con il metodo di Gauss-Seidel, introdotto nella sezione *Rilassamento di Gauss-Seidel applicato al sistema lineare della radiosità*. Il ciclo di iterazione (equazione (3) della sezione *Soluzione con metodi iterativi: rilassamento di Jacobi*) viene eseguito su tutte le componenti dei vettori, quindi su tutte le n equazioni del sistema, però ora in ciascuna equazione, diciamo la j-sima (j=1,...n), vengono utilizzate le componenti già aggiornate del vettore b: in altre parole, nell'iterazione k+1-sima al secondo membro dell'equazione j-sima si utilizzano le componenti b<sub>i</sub><sup>(k+1)</sup> invece di b<sub>i</sub><sup>(k)</sup>, per i=1,...j-1. Partiamo di nuovo con l'approssimante iniziale b<sup>(0)</sup> = (0,0,0,0). Usiamo ancora (38) dell'*Esempio 1*, ma con le sostituzioni appena indicate. Ecco i risultati:

 $\mathbf{b}^{(1)} = (1, 0.167, 0.194, 0.233)$  $\mathbf{b}^{(2)} = (1, 0.238, 0.245, 0.249)$  $\mathbf{b}^{(3)} = (1, 0.249, 0.25, 0.25)$  $\mathbf{b}^{(4)} = (1, 0.25, 0.25, 0.25)$ 

Il numero di iterazioni necessarie per arrivare al punto di equilibrio è lo stesso del metodo di Jacobi illustrato nell'*Esempio 1* della tenda piramidale, ma ora ciascun approssimante è più vicino al risultato finale di quanto non accadesse prima.

Ecco il corrispondente notebook in Mathematica:

 $P = \{\{0, 0, 0, 0\}, \{0, 0, t, t\}, \{0, 0, 0, t\}, \{0, 0, 0, 0\}\}$   $NN = \{\{1, 0, 0, 0\}, \{-t, 1, 0, 0\}, \{-t, -t, 1, 0\}, \{-t, -t, -t, 1\}\}$   $e = \{1, 0, 0, 0\}$  TraditionalForm[Ninv = Inverse[NN]]  $b = \{0, 0, 0, 0\}$  For [k = 1, k < 5, k++, For [i = 1, i < 5, i++,  $Expand[b[[i]] = (Ninv .P)[[i]]. b + Ninv[[i]] .e]]; N[b] /. t \rightarrow 1/6] \}$ 

#### 3. Metodo di Southwell

Trattiamo infine lo stesso esempio con il metodo di Southwell, introdotto nella sezione *Rilassamento di Southwell applicato al sistema lineare della radiosità*. Il ciclo di iterazione dovrebbe ora riguardare la matrice dell'energia K invece di quella della radiosità M, ma poiché le aree degli elementi sono tutte uguali le due matrici coincidono. Quindi il ciclo iterativo rimane lo stesso (equazione (3) nella sezione *Soluzione con metodi iterativi: rilassamento di Jacobi*), ma viene

eseguito ora su una sola equazione del sistema, quella con il resto maggiore. Il vettore dei resti alla k-sima iterazione è proporzionale a  $r^{(k)} = e - M b^{(k)}$ , grazie a (13') della sezione Rilassamento di Southwell applicato al sistema lineare della radiosità. Quando il vettore dei resti diventa nullo abbiamo raggiunto l'equilibrio. Per ciascuna iterazione calcoliamo il vettore dei resti, determiniamo l'indice (diciamo j) della sua componente più grande (in valore assoluto) e calcoliamo l'iterazione successiva svolgendo solo la j-sima equazione del sistema lineare della radiosità. Con i metodi precedentemente illustrati ci volevano da tre a quattro iterazioni, risolvendo 4 equazioni per ogni iterazione. Quindi ora potremmo aspettarci di dover svolgere da 12 a 16 iterazioni. Ma poiché ogni volta trattiamo proprio la componente con l'errore maggiore, il metodo è più efficiente, e bastano solo sette iterazioni. Forniamo i risultati con un numero maggiore di cifre decimali rispetto a prima per evidenziare la progressione dell'iterazione. Si noti, per verifica di quanto appena detto, che in ogni iterazione cambia una sola componente (mai la prima, perché quella ha errore zero, in quanto la corrispondente elemento ha coefficiente di riflettività zero).

 $\mathbf{b}^{(1)} = (1, 0, 0, 0.226852)$ 

 $\mathbf{b}^{(2)} = (1, 0, 0.238555, 0.226852)$ 

 $\mathbf{b}^{(3)} = (1, 0.244234, 0.238555, 0.226852)$ 

 $\mathbf{b}^{(4)} = (1, 0.244234, 0.238555, 0.248236)$ 

 $\mathbf{b}^{(5)} = (1, 0.249596, 0.238555, 0.248236)$ 

 $\mathbf{b}^{(6)} = (1, 0.249596, 0.238555, 0.249872)$ 

 $\mathbf{b}^{(7)} = (1, 0.249596, 0.249957, 0.249872)$ 

cioè, a due cifre decimali esatte (in realtà tre!),  $b^{(7)} = (1, 0.25, 0.25, 0.25)$ . Ecco il corrispondente notebook di <u>Mathematica</u>. Si noti che tutte gli elementi hanno la stessa area A, quindi, a meno della moltiplicazione per A, il vettore resto alla k-sima iterazione coincide con  $\mathbf{e} - \mathbf{M}\mathbf{b}^{(k)}$ .

 $P = \{\{0, 0, 0, 0\}, \{0, 0, t, t\}, \{0, 0, 0, t\}, \{0, 0, 0, 0\}\}$   $NN = \{\{1, 0, 0, 0\}, \{-t, 1, 0, 0\}, \{-t, -t, 1, 0\}, \{-t, -t, -t, 1\}\}$   $e = \{1, 0, 0, 0\}$  MatrixForm[M = NN - P] TraditionalForm[Ninv = Inverse[NN]]  $b = \{0, 0, 0, 0\}$  maxr = 1.; While[ (maxr > 0.001), r = N[e - M.b] /. t -> 1/6;

```
maxr = Max[r[[1]], r[[2]], r[[3]], r[[4]]];
For [i = = 1, i < 5, i++, If [(r[[i]] == maxr), i0 = i,
Continue] ];
b[[i0]] = N[(Ninv .P)[[i0]]. b + Ninv[[i0]] .e] /. t ->
1/6;
Print[TraditionalForm[N[b]]] ]
```

# ESEMPIO 2: DISTRIBUZIONE DELLA LUCE ENTRO UNA STANZA CUBICA

La scena ora consiste dell'interno di una stanza a forma di cubo, quindi con sei facce identiche quadrate. La faccia alta ha una lampada che emette luce con intensità 1. Le altre facce non emettono luce propria, solo riflessa. Non ci sono finestre. La base, cioè il pavimento, è più scura degli altri muri: fissiamo  $\rho_1 = 1/4$ . Le pareti ed il soffitto sono grigio chiaro, e quindi poniamo  $\rho_i = 1/2$  per i=2, 3, 4, 5, 6. Anche in questo caso ci limitiamo all'approssimazione rudimentale nella quale ciascuna parete forma un unico elemento (quindi non noteremo gradazioni di luce sulle pareti). Per comodità scriviamo  $\rho_+$  invece di 1/2 e  $\rho_-$  invece di 1/4. Quindi il vettore dell'energia propria emessa è  $\mathbf{e} = (0,0,0,0,0,1)$ , mentre il vettore della riflettività è  $\rho = (1/4, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ . Nel seguito scriveremo e invece di  $\mathbf{e}$ .

Anche in questo caso gli elementies sono pianepiani, e quindi i fattori di forma diagonali,  $F_{i\cdot i}$ , sono tutti nulli. D'altra parte, la stanza è completamente simmetrica. Quindi ogni faccia copre, vista da ogni altra faccia ad essa laterale, lo stesso angolo solido. Inoltre ogni faccia copre lo stesso angolo solido (ma diverso da quello di prima) quando è vista dalla faccia diametralmente opposta. Perciò ci sono solo due fattori di forma non nulli: quello laterale, che per semplicità chiamiamo F<sub>i</sub>, e quello frontale, F<sub>f</sub>. In particolare, F<sub>i-j</sub> = F<sub>f</sub> per (i,j)=(1,6), (2,4), (3,5) ed i loro trasposti,  $F_{i-j} = F_f$  per (i,j)= (1,2), (1,3), (1,4), (1,5), (6,2), (6,3), (6,4), (6,5), (2,3), (2,5), (3,4), (4,5) e tutti i loro trasposti. Si noti che ciascuna faccia ha quattro facce laterali ed una frontale. Quindi, dall'equazione della conservazione dell'energia (o dell'angolo solido totale)  $\Sigma_j$   $F_{i-j} = 1$ , che abbiamo introdotto nella sezione *L'equazione della radiosità*, si ottiene  $F_f + 4$   $F_I = 1$ . Il valore di  $F_f$  si può calcolare grazie alla formula dimostrata nella sezione *Calcolo dei fattori di forma*:

$$F_{i,j} = \frac{1}{A_i} \int_{A_j} \int_{A_j} \frac{\cos \theta_i \cos \theta_j}{\pi r^2} H_{ij} dA_j dA_i$$

Qui non ci sono ostruzioni dentro la stanza, quindi H vale costantemente 1, e le facce antistanti sono parallele, quindi i due angoli  $\theta_i$  e  $\theta_j$  sono uguali. Chiameremo quest'angolo  $\theta$ . L'integrale si può approssimare numericamente con il metodo illustrato nella sezione *Calcolo dei fattori di forma*, legato al metodo di z-buffer. Ma in questo caso l'integrale si può anche calcolare analiticamente come segue. Per simmetria, possiamo supporre che i=1 e j=6, cioè che la faccia i sia quella inferiore e la faccia j quella superiore. Per l'invarianza sotto cambiamento di scala degli angoli solidi sottesi da facce antistanti, possiamo

supporre che il cubo abbia lato 1, quindi tutte le facce abbiano area 1. Possiamo anche supporre che il cubo abbia le facce parallele ai piani coordinati e giaccia nell'ottante positivo, con un vertice nell'origine. Allora la sua faccia inferiore ha coordinate (x,y,0) per x e y che variano fra 0 e 1. Analogamente, la faccia superiore ha coordinate (x,y',1) per x' e y' che variano fra 0 e 1.

Perciò, a meno del fattore  $\pi$  al denominatore, l'integrando ha la forma seguente. Chiamiamo p il vettore che congiunge un punto generico della faccia inferiore ad uno della faccia superiore: p = (x',y',1) - (x,y,0) = (x'-x, y'-y, 1). Osserviamo che la terza coordinata di p vale costantemente 1. Ora normalizziamo p, ottenendo q = p/||p||. Il valore di cos  $\theta$  è esattamente il prodotto scalare fra il versore q ed il versore perpendicolare alla due facce, cioè il versore dell'asse z, (0,0,1). Pertanto cos  $\theta$  è nient'altro che la terza componente del versore q: cioè cos  $\theta$ =1/||p||. D'altra parte, la distanza r fra (x,y,0) e (x',y',1) è proprio ||p||. Quindi l'integrando è 1/( $\pi$ ||p||<sup>4</sup>). (Questi risultati sono stati già osservati nella sezione *Calcolo dei fattori di forma*).

Questo integrando è il reciproco di un polinomio di secondo grado nelle quattro variabili x,y,x',y', e quindi si può calcolare analiticamente in termini di funzioni arcoseno e logaritmo. Il calcolo è laborioso, ed è stato svolto in un Esempio precedente (*Calcolo del fattore di forma fra due pareti adiacenti di una stanza cubica*). Il risultato è:

 $F_{f} = 0.2$ 

Chi vuole può riverificare il risultato analitico tramite approssimazione numerica. Ecco il listato di un notebook di Mathematica che svolge il calcolo:

pointdown[x, y] = {x, y, 0}; pointup[x', y'] = {x', y', 1}; displacement[x, y, x', y'] = pointup[x', y'] - pointdown[x, y]; NormSquare[x, y, x', y'] = 0; For[i = 1, i < 4, i++, NormSquare[x, y, x', y'] = NormSquare[x, y, x', y'] += displacement[x, y, x', y'][[i]]^2]; Norm[x, y, x', y'] = NormSquare[x, y, x', y']^(1/2); Integrand[x, y, x', y'] = 1/ NormSquare[x, y, x', y']^2; N[F = (1/Pi) Integrate[Integrand[x, y, x', y'], {x, 0, 1}, {y, 0, 1}, {x', 0, 1}, {y', 0, 1}] ]

Pertanto, dall'equazione della conservazione dell'energia  $F_f + 4 F_l = 1$  (introdotta nella sezione *L'equazione della radiosità*), segue  $F_l = F_f = 0.2$ . Il fatto che  $F_l = F_f \dot{e}$  insolito, ed è vero solo per le prime due cifre decimali e solo perché la stanza ha

forma cubica. In seguito (*Esempio 3*) vedremo esempi di stanze di forma rettangolare dove questo non accade. Nel seguito indicheremo con F il valore comune  $F_1 = F_f$ .

Quindi la matrice M del sistema lineare della radiosità è la seguente:

$$\begin{pmatrix} 1 & -\rho_{-}F & -\rho_{-}F & -\rho_{-}F & -\rho_{-}F & -\rho_{-}F \\ -\rho_{+}F & 1 & -\rho_{+}F & -\rho_{+}F & -\rho_{+}F & -\rho_{+}F \\ -\rho_{+}F & -\rho_{+}F & 1 & -\rho_{+}F & -\rho_{+}F & B_{3} \\ -\rho_{+}F & -\rho_{+}F & -\rho_{+}F & 1 & -\rho_{+}F & B_{3} \\ -\rho_{+}F & -\rho_{+}F & -\rho_{+}F & 1 & -\rho_{+}F & B_{5} \\ -\rho_{+}F & -\rho_{+}F & -\rho_{+}F & -\rho_{+}F & 1 & -\rho_{+}F \\ B_{5} & B_{6} \end{pmatrix} = \begin{pmatrix} E_{1} \\ E_{2} \\ E_{3} \\ E_{4} \\ E_{5} \\ E_{6} \end{pmatrix}$$

cioè

( 1	-0.05	-0.05	-0.05	-0.05	-0.05	$\left(B_{1}\right)$		0
-0.1	1	-0.1	-0.1	-0.1	-0.1	$B_2$		0
-0.1	-0.1	1	-0.1	-0.1	-0.1	$B_3$	_	0
-0.1	-0.1	-0.1	1	-0.1	-0.1	$B_4$	_	0
-0.1	-0.1	-0.1	-0.1	1	-0.1	$B_5$		0
-0.1	-0.1	-0.1	-0.1	-0.1	1)	$\left(B_{6}\right)$		(1)

Applichiamo anche questa volta il metodo iterativo formulato in (6') della sezione *Analisi del metodo di Gauss-Seidel*, spezzando M come M = N – P, con N matrice triangolare inferiore (o meglio non superiore) e P triangolare (strettamente) superiore. Abbiamo:

$$\mathbf{N} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -0.1 & 1 & 0 & 0 & 0 & 0 \\ -0.1 & -0.1 & 1 & 0 & 0 & 0 \\ -0.1 & -0.1 & -0.1 & 1 & 0 & 0 \\ -0.1 & -0.1 & -0.1 & -0.1 & 1 & 0 \\ -0.1 & -0.1 & -0.1 & -0.1 & 1 \end{pmatrix}$$

	(0	0.05	0.05	0.05	0.05	0.05
P =	0	0	0.1	0.1	0.1	0.1
	0	0	0	0.1	0.1	0.1
	0	0	0	0	0.1	0.1
	0	0	0	0	0	0.1
	0	0	0	0	0	0 )

Calcoliamo l'inversa di N (possiamo usare il metodo di Gauss, o anche il metodo diretto usato prima per la tenda piramidale (*Esempio 1*). Per semplicità ricaviamo il risultato dalla stesso notebook di Mathematica usato in quella circostanza per verifica. Si ottiene:

	( 1	0	0	0	0	0)
$N^{-1} =$	0.1	1	0	0	0	0
	0.11	0.1	1	0	0	0
	0.121	0.11	0.1	1	0	0
	0.1331	0.121	0.11	0.1	1	0
	0.14641	0.1331	0.121	0.11	0.1	1

#### 1. Metodo di Jacobi

Ora gli approssimanti alle varie iterazioni successive si trovano come nel caso della tenda piramidale dell'*Esempio 1*. Ecco i risultati col metodo di Jacobi:

$$\begin{split} & b^{(1)} = (0.05, \, 0.105, \, 0.116, \, 0.127, \, 0.140, \, 1.053) \\ & b^{(2)} = (0.077, \, 0.151, \, 0.155, \, 0.158, \, 0.159, \, 1.070) \\ & b^{(3)} = (0.085, \, 0.163, \, 0.164, \, 0.164, \, 0.164, \, 1.074) \\ & b^{(4)} = (0.086, \, 0.165, \, 0.165, \, 0.166, \, 0.166, \, 1.075) \\ & b^{(5)} = (0.087, \, 0.166, \, 0.166, \, 0.166, \, 0.166, \, 1.075) \\ & b^{(6)} = (0.087, \, 0.166, \, 0.166, \, 0.166, \, 0.166, \, 1.075) \end{split}$$

Alla quinta iterazione l'approssimante ha raggiunto il punto di equilibrio (limitatamente alle prime tre cifre decimali: se si richiede una precisione maggiore allora sono necessarie più iterazioni).

Per completezza, riportiamo qui il notebook di Mathematica che abbiamo utilizzato:

```
 t=0.1 \\ P = \{\{0, t/2, t/2, t/2, t/2, t/2\}, \{0, 0, t, t, t, t\}, \{0, 0, 0, t, t, t\}, \{0, 0, 0, 0, t, t\}, \{0, 0, 0, 0, 0, 0\} \} \\ NN = \{\{1, 0, 0, 0, 0, 0\}, \{-t, 1, 0, 0, 0, 0\}, \{-t, -t, 1, 0, 0, 0\}, \{-t, -t, -t, 1, 0\}, \{-t, -t, -t, -t, -t, 1\} \} \\ e = \{0, 0, 0, 0, 0, 0, 1\} \\ Ninv = Inverse[NN] \\ TraditionalForm[Ninv] \\ b = \{0, 0, 0, 0, 0, 0\} \\ For[i = 0, i < 7, i++, N[Expand[b = Ninv. P . b + Ninv . e]]; Print[b] ]
```

#### 2. Metodo di Gauss-Seidel

Ed ora i risultati col metodo di Gauss-Seidel (scriviamo in rosso le cifre che sono cambiate rispetto al metodo precedente):

 $b^{(1)} = (0.05, 0.105, 0.116, 0.130, 0.146, 1.068)$   $b^{(2)} = (0.078, 0.154, 0.158, 0.161, 0.164, 1.074)$   $b^{(3)} = (0.086, 0.164, 0.165, 0.165, 0.166, 1.075)$   $b^{(4)} = (0.087, 0.166, 0.166, 0.166, 0.166, 1.075)$  $b^{(5)} = (0.087, 0.166, 0.166, 0.166, 0.166, 1.075)$ 

In questo esempio, il metodo di Gauss-Seidel impiega una iterazione in meno rispetto al metodo di Jacobi per raggiungere l'equilibrio, ed inoltre ad ogni iterazione i suoi approssimanti sono più vicini al risultato finale: se si richiedono tre sole cifre decimali, questa differenza è piuttosto evidente quando si è ancora lontani dall'equilibrio, cioè nelle prime tre iterazioni, soprattutto la seconda. Soprattutto in questo senso il metodo di Gauss-Seidel è più veloce.

Di nuovo, riportiamo il notebook di Mathematica che abbiamo utilizzato:

```
e={0,0,0, 0,0,1}
Ninv=Inverse[NN]
b={0,0,0,0,0,0}
For[k=1,k<10,k++,
    For [i=1, i<7, i++,
        Expand[b[[i]]=(Ninv .P)[[i]]. b + Ninv[[i]] .e] ];
Print[N[b]] ]</pre>
```

#### 3. Metodo di Southwell

Infine, presentiamo i risultati ottenuti con il metodo di Southwell, secondo le linee esposte nella sezione *Metodo del rilassamento di Southwell applicato al sistema lineare della radiosità*. La matrice dell'energia K, introdotta nella sottosezione *Rilassamento di Southwell*, coincide con M perché le aree di tutte gli elementi sono uguali; inoltre valgono 1, quindi i vettori  $\beta$  della energia e b della radiosità coincidono. I risultati si ottengono rilassando una sol'elemento (invece che tutte e sei) per ogni ciclo di iterazione. (Si tratta dell'elemento col valore massimo del vettore resto r<sup>(k)</sup>, il quale, grazie a (13'), vale r<sup>(k)</sup> = e - M b<sup>(k)</sup>).

Quindi mediamente potremmo aspettarci sei volte più iterazioni, ma, poiché ogni volta rilassiamo l'elemento con l'errore maggiore, questo metodo è più veloce: bastano 17 iterazioni per raggiungere l'equilibrio a meno di tre cifre decimali esatte. Non riportiamo i risultati per brevità: essi si possono trovare grazie al seguente notebook di Mathematica.

```
t = 0.1
 P = \{ \{0, t/2, t/2, t/2, t/2, t/2\}, \{0, 0, t, t, t, t\}, \{0, 0, 0, t\}, \{0, 0
 ,0,0,t,t\},\{0,0,0,0,0,t\},\{0,0,0,0,0,0\}\}
t,1,0,0, \{-t,-t,-t,-t,1,0\}, \{-t,-t,-t,-t,-1\}
M=NN-P
 e = \{0, 0, 0, 0, 0, 1\}
Ninv=Inverse[NN]
b = \{0, 0, 0, 0, 0, 0\}
 maxr=1.
 While[(maxr>0.001),
             r=N[e-M.b];
           maxr=Max[r[[1]], r[[2]], r[[3]], r[[4]], r[[5]],r[[6]]];
           For [i=1, i<7, i++,
                        If [(r[[i]]==maxr),i0=i,Continue]
                        ];
            b[[i0]]=N[(Ninv.P)[[i0]].b + Ninv[[i0]].e];
 Print[N[b]] ]
```

Ecco le approssimazioni successive secondo Southwell ottenute tramite questo

#### notebook di Mathematica:

{0.,0.,0.,0.,0.,0.,1.} 0.,0.,0.,0.,0.139755,1.} {0.,0.,0.,0.144806,0.139755,1.} <sup>(</sup>0.,0.,0.148367,0.144806,0.139755,1.} {0.,0.150457,0.148367,0.144806,0.139755,1.} [0.0791693,0.150457,0.148367,0.144806,0.139755,1.} {0.0791693,0.150457,0.148367,0.144806,0.139755,1.06874} {0.0791693,0.150457,0.148367,0.144806,0.163011,1.06874} {0.0791693,0.150457,0.148367,0.163851,0.163011,1.06874} {0.0791693,0.150457,0.16432,0.163851,0.163011,1.06874 {0.0791693,0.164545,0.16432,0.163851,0.163011,1.06874 {0.0862235,0.164545,0.16432,0.163851,0.163011,1.06874} 0.0862235,0.164545,0.16432,0.163851,0.163011,1.07451 0.0862235,0.164545,0.16432,0.163851,0.165701,1.07451 0.0862235,0.164545,0.16432,0.165798,0.165701,1.07451 {0.0862235,0.164545,0.165846,0.165798,0.165701,1.07451} {0.0862235,0.165868,0.165846,0.165798,0.165701,1.07451} {0.0868861,0.165868,0.165846,0.165798,0.165701,1.07451}

## ESEMPIO 3: DISTRIBUZIONE DELLA LUCE ENTRO UNA STANZA RETTANGOLARE

Ora, invece di una stanza a forma di cubo, consideriamo una stanza rettangolare, di altezza 1, lunghezza L e profondità W. Le condizioni di illuminazione sono le stesse della stanza cubica vista precedentemente nell' *Esempio* 2: le trascriviamo qui per completezza. La faccia alta ha una lampada che emette luce con intensità 1. Le altre facce non emettono luce propria, solo riflessa. Non ci sono finestre. La base, cioè il pavimento, è più scura degli altri muri: fissiamo  $\rho_1 = 1/4$ . Le pareti ed il soffitto sono grigio chiaro, e quindi poniamo  $\rho_i = 1/2$  per i=2, 3, 4, 5, 6. Anche in questo caso ci limitiamo all'approssimazione rudimentale nella quale ciascuna parete forma un unico elemento (quindi non noteremo gradazioni di luce sulle pareti). Per comodità scriviamo  $\rho_+$  invece di 1/2 e  $\rho_-$  invece di 1/4. Quindi il vettore dell'energia propria emessa è **e** = (0,0,0,0,0,1), mentre il vettore della riflettività è  $\rho = (1/4, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}$ ). Nel seguito scriveremo e invece di **e**.

Anche in questo caso gli elementies sono pianie, e guindi i fattori di forma diagonali, F<sub>i-i</sub>, sono tutti nulli. Ora però la stanza non è completamente simmetrica. Ogni faccia copre lo stesso angolo solido solo quando è vista dalle due facce laterali opposte, cioè antistanti fra loro. Quindi ogni faccia ha un fattore di forma per la faccia antistante ed altri due per le due coppie di facce laterali. In tutto quindi ci sono 9 fattori di forma diversi. Ma la relazione di reciprocità della reversibilità dei percorsi ottici, introdotta nella sezione L'equazione della *radiosità,* ci permette di ricavare  $F_{i-i}$  da  $F_{i-i}$ :  $F_{i-j} = A_i/A_i$   $F_{i-i}$  Perciò dobbiamo solo calcolare, tramite integrazione analitica o numerica, tre fattori di forma per facce in disposizione antistante e tre per disposizione laterale. Numeriamo le facce cosi': 1 il pavimento, 6 il soffitto, 2 e 4 le parete di lunghezza L, 3 e 5 le pareti di lunghezza W. In realtà, l'equazione della conservazione dell'energia (anch'essa introdotta nella sezione *L'equazione della radiosità*)  $\Sigma_i$  F<sub>i-i</sub> =1 ci fornisce il valore dell'ultimo fattore di forma F<sub>i-m</sub> di ogni faccia *i*, una volta noto F<sub>i-i</sub> per tutti i *j* diversi da m. Quindi dobbiamo calcolare in tutto sei fattori di forma.

Calcoliamo i fattori dalla seguente formula, dimostrata nella sezione Calcolo dei fattori di forma:

$$\mathbf{F}_{i-j} = \frac{1}{\mathbf{A}_i} \int_{A_i} \int_{A_i} \frac{\cos \theta_i \cos \theta_j}{\pi r^2} \mathbf{H}_{ij} \, \mathrm{d}\mathbf{A}_j \, \mathrm{d}A_i$$

Cominciamo con le facce opposte. Di nuovo, non ci sono ostruzioni dentro la stanza, quindi H vale costantemente 1, e le facce antistanti sono parallele, quindi  $\theta_i=\theta_j$ . Il calcolo è analogo a quello per la stanza cubica dell'*Esempio 2*. Approssimiamo l'integrale numericamente, usando un notebook di <u>Mathematica</u> analogo a quello impiegato per la stanza cubica. Ad esempio, se L=2 e W=2.5, troviamo i seguenti fattori di forma per le facce antistanti, a meno di due cifre decimali:

 $F_{1-6} = F_{6-1} = 0.45$ 

 $F_{2-4} = F_{4-2} = 0.13$ 

 $F_{3-5} = F_{5-3} = 0.08$ 

Invece, se L=W=2 (stanza quadrata) si trova:

 $F_{1-6} = F_{6-1} = 0.42$ 

 $F_{2-4} = F_{4-2} = F_{3-5} = F_{5-3} = 0.12$ 

Ecco il notebook che abbiamo usato per il caso i=1, j=6 (gli altri si ottengono ruotando in ordine ciclico i valori di height, length e width):

```
height=1
length=2
width =2
pointdown[x,y]={x,y,0};
pointup[x',y']={x',y',height};
displacement[x,y,x',y']=pointup[x',y']-pointdown[x,y];
NormSquare[x,y,x',y']=0;
For[i=1,i<4,i++,
    NormSquare[x,y,x',y']+=displacement[x,y,x',y'][[i]]^2];
Integrand[x,y,x',y']=height^2/ NormSquare[x,y,x',y']^2;
F=(1/(Pi * length * width))
NIntegrate[Integrand[x,y,x',y'],{x,0,length},{y,0,width},{x
',0,length},{y',0,width}]; Print[F]
```

Ora consideriamo il problema del calcolo di fattori di forma sulle facce laterali. Possiamo ancora supporre che la stanza abbia le facce parallele ai piani coordinati e giaccia nell'ottante positivo, con un vertice nell'origine. Allora la sua faccia 0 (il pavimento) ha coordinate (x,y,0) per x e y che variano fra 0 e L e fra 0 e W, rispettivamente. La faccia laterale per x=0 ha coordinate (0, y',z') per y' che varia fra 0 e L e z' fra 0 e 1. Se p è il vettore da (x,y,0) a (0, y',z') e q il versore p/||p||, allora nell'integrale  $\cos\theta_1$  è il prodotto scalare fra q ed il versore (0,0,1)

normale al piano di base, mentre  $\cos\theta_2$  è il prodotto scalare fra q ed il versore (1,0,0) normale al piano x=0 che contiene l'elemento 2. Quindi  $\cos\theta_1 = z'/||p||$  e  $\cos\theta_2 = x/||p||$ . Inoltre r=||p||. (Questi risultati sono stati già osservati nella sezione *Calcolo dei fattori di forma*). L'integrale si può calcolare analiticamente, ma il calcolo è laborioso. Oppure lo si può approssimare numericamente con Mathematica; per L=2 e W=2.5 si trova, ad esempio:

 $F_{1-2} = 0.23 = A_2/A_1 F_{2-1}$ 

Però l'approssimazione numerica è affetta da un forte errore, perché l'integrando diverge quando r si avvicina a 0, cioè al segmento in cui il pavimento tocca la parete. Per evitare rischi di imprecisione, supponiamo che la stanza sia quadrata, cioè che sia L=W (ma il valore comune non sia 1, perché abbiamo già trattato la stanza cubica). Allora  $F_{1-2}=F_{1-3}=F_{1-4}=F_{1-5}=F_{6-2}=F_{6-3}=F_{6-4}=F_{6-5}$ , e dall'equazione della conservazione dell'energia (o dell'angolo solido totale)  $\Sigma_j F_{i-j}$  =1 (introdotta nella sezione *L'equazione della radiosità*) si ricava il loro valore comune, che chiameremo  $F_{pav-lato}$ :

 $F_{pav-lato} = (1 - F_{1-6})/4 = 0.145$ 

Analogamente, chiamiamo  $F_{lato-pav}$  il valore comune  $F_{2-1} = F_{2-6} = F_{4-1} = F_{4-6} = F_{3-1} = F_{3-6} = F_{5-1} = F_{5-6} = F_{lato-lato}$  il valore comune  $F_{3-2} = F_{5-2} = F_{3-4} = F_{5-4} = F_{2-3} = F_{2-5} = F_{4-3} = F_{4-5}$ . Sappiamo che  $F_{2-1} = A_1/A_2$   $F_{1-2} = 2$   $F_{1-2}$ , quindi  $F_{lato-pav} = 2$   $F_{pav-lato} = 0.29$ . D'altra parte, sappiamo che  $\Sigma_j$   $F_{2-j} = 1$ , ma  $F_{2-3} = F_{2-5} = F_{lato-lato}$ ,  $F_{2-4} = 0.12$  e  $F_{2-6} = F_{2-1} = F_{lato-pav} = 0.29$ . Da qui segue che  $F_{lato-lato} = (1 - 0.12 - 2x0.29)/2 = 0.15$ . Abbiamo determinato tutti i fattori di forma per la stanza quadrata. Riassumendo:

ci sono due 2 fattori di forma frontali:

da pavimento a soffitto (o viceversa	) 0.42

da parete a parete antistante 0.12

c' è	un fattore di forma	da parete a parete adiacente	0.15
------	---------------------	------------------------------	------

- c' è un fattore di forma da parete a pavimento (o soffitto) 0.29
- c' è un fattore di forma da pavimento (o soffitto) a parete 0.145

Con questi dati possiamo scrivere la matrice M del sistema della radiosità, e le matrici triangolare inferiore e superiore N e P della decomposizione M = N - P, come in (6') nella sezione Analisi del metodo di Gauss-Seidel.

Ci limitiamo a scrivere N e P, ricordando che la riflettività del pavimento vale ¼ e quella degli altri elementi ½.

	( 1	0	0	0	0	0)
	$-\frac{1}{2}0.29$	1	0	0	0	0
NI	$-\frac{1}{2}0.29$	$-\frac{1}{2}0.15$	1	0	0	0
IN =	$-\frac{1}{2}0.29$	$-\frac{1}{2}0.12$	$-\frac{1}{2}0.15$	1	0	0
	$-\frac{1}{2}0.29$	$-\frac{1}{2}0.15$	$-\frac{1}{2}0.12$	$-\frac{1}{2}0.15$	1	0
	$\left(-\frac{1}{2}0.42\right)$	$-\frac{1}{2}0.145$	$-\frac{1}{2}0.145$	$-\frac{1}{2}0.145$	$-\frac{1}{2}0.145$	1)

	( 1	0	0	0	0	0)
	-0.145	1	0	0	0	0
	-0.145	-0.075	1	0	0	0
_	-0.145	-0.06	-0.075	1	0	0
	-0.145	-0.075	-0.06	-0.075	1	0
	1	-0.0725	-0.0725	-0.0725	-0.0725	1

$\mathbf{P} = \begin{bmatrix} 0 & 0 & \frac{1}{2}0.15 & \frac{1}{2}0.15 & \frac{1}{2}0.15 & \frac{1}{4}0.29 \\ 0 & 0 & 0 & \frac{1}{2}0.15 & \frac{1}{2}0.15 & \frac{1}{4}0.29 \\ 0 & 0 & 0 & 0 & \frac{1}{2}0.15 & \frac{1}{4}0.29 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{4}0.29 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$		(0)	$\frac{1}{2}0.145$	$\frac{1}{2}$ 0.145	$\frac{1}{2}$ 0.145	$\frac{1}{2}$ 0.145	$\frac{1}{4}0.42$
$\mathbf{P} = \begin{vmatrix} 0 & 0 & 0 & \frac{1}{2}0.15 & \frac{1}{2}0.15 & \frac{1}{4}0.29 \\ 0 & 0 & 0 & 0 & \frac{1}{2}0.15 & \frac{1}{4}0.29 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{4}0.29 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{vmatrix}$		0	0	$\frac{1}{2}0.15$	$\frac{1}{2}0.15$	$\frac{1}{2}0.15$	$\frac{1}{4}0.29$
$ \begin{bmatrix} 1 & - \\ 0 & 0 & 0 & 0 & \frac{1}{2} 0.15 & \frac{1}{4} 0.29 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{4} 0.29 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} $	<b>D</b> _	0	0	0	$\frac{1}{2}0.15$	$\frac{1}{2}0.15$	$\frac{1}{4}0.29$
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	г –	0	0	0	0	$\frac{1}{2}0.15$	$\frac{1}{4}0.29$
		0	0	0	0	0	$\frac{1}{4}0.29$
		0	0	0	0	0	0

	(0	0.0725	0.0725	0.0725	0.0725	0.1005
	0	0	0.075	0.06	0.075	0.0725
_	0	0	0	0.075	0.06	0.0725
=	0	0	0	0	0.075	0.0725
	0	0	0	0	0	0.0725
	0	0	0	0	0	0 )

#### 1. Metodo di Jacobi

Ora possiamo eseguire le iterazioni come nei precedenti esempi della tenda piramidale e della stanza cubica. Per non diventare ripetitivi, ci limitiamo a riportare i risultati del metodo di Jacobi (questa volta alle prime quattro cifre decimali), lasciando al lettore i calcoli per questo metodo (per il quale però trascriviamo il notebook di Mathematica) e per il metodo di Gauss-Seidel. Il metodo di Southwell verrà implicitamente trattato fra qualche pagina quando illustreremo il raffinamento progressivo.

 $b^{(1)} = (0.1005, 0.0871, 0.0936, 0.0993, 0.1067, 1.0491)$   $b^{(2)} = (0.1335, 0.1164, 0.118, 0.1192, 0.1202, 1.0624)$   $b^{(3)} = (0.1411, 0.1225, 0.1228, 0.1231, 0.1233, 1.0653)$   $b^{(4)} = (0.1427, 0.1238, 0. 1238, 0. 1239, 0. 1239, 1.0659)$   $b^{(5)} = (0.1430, 0.124, 0. 124, 0. 1241, 0. 1241, 1.066)$   $b^{(6)} = (0.1431, 0.1241, 0. 1241, 0. 1241, 0. 1241, 1.066)$  $b^{(7)} = (0.1431, 0.1241, 0. 1241, 0. 1241, 0. 1241, 1.066)$ 

Ecco il notebook che abbiamo usato per il calcolo:

```
t1=0.0725; t2=0.075; t3=0.06; t4=0.145
P={{0,t1,t1,t1,t1,0.1005},{0,0,t2,t3,t2,t1},{0,0,0,t2,t3,t1},{0,0,0,0,c2,t1},{0,0,0,0,0,0}}
NN={{1,0,0,0,0,0},{-t4,1,0,0,0,0},{-t4,-t2,1,0,0,0},{-t4,-t3,-t2,1,0,0},{-t4,-t2,1,0,0,0},{-t4,-t2,-t3,-t2,1,0},
{-0.21,-t1,-t1,-t1,-t1,1}}; TraditionalForm[NN]
e={0,0,0,0,0,0,1}
Ninv=Inverse[NN]; TraditionalForm[Ninv]
b={0,0,0,0,0,0}
For[i=0, i<10, i++, N[Expand[b=Ninv .P. b + Ninv .e]];
Print[b] ]</pre>
```

Si noti quanto rapido sia l'aumento delle componenti del vettore di radiosità nelle prime iterazioni: infatti all'inizio l'iterazione cambia la prima cifra decimale. Poi la prima cifra si stabilizza e comincia a cambiare la seconda, e via via le altre. Quindi le differenze successive diventano via via ordini di grandezza più piccoli. In effetti, osserviamo che le prime due cifre decimali sono stabili già dopo tre iterazioni, le prime tre dopo quattro iterazioni, le prime quattro dopo sei iterazioni.

Proprio per questo rapido incremento iniziale e lento incremento successivo il metodo del raffinamento progressivo, introdotto nelle sezioni Verso un raffinamento progressivo e seguenti, permette di estrapolare in pochissime iterazioni il punto fisso finale. Tra qualche pagina svolgeremo i calcoli con questo metodo.

### 2. Metodo di Southwell

In questo metodo, utilizziamo il vettore dell'energia creata  $\underline{\epsilon}$ , quello dell'energia irradiata  $\underline{\beta}$ , e la matrice dell'energia <u>K</u>. Le pareti laterali hanno area 2, mentre pavimento (i=1) e soffitto (i=6) hanno area 4. Poiché K<sub>ij</sub> = (A<sub>i</sub>/A<sub>j</sub>)M<sub>ij</sub> la matrice K si decompone come K = N' – P', dove le matrici N' e P' differiscono da N e P precedentemente utilizzate in questo modo:

- N' differisce da N solo nella ultima riga, colonne 2,3,4,5 (dove i coefficienti vengono moltiplicati per ½), e nella prima colonna, righe 2,3,4,5 (i cui coefficienti vengono moltiplicati per 2)
- P' differisce da P solo nella prima riga, colonne 2,3,4,5 (dove i coefficienti vengono moltiplicati per 2), e nella ultima colonna, righe 2,3,4,5 (i cui coefficienti vengono moltiplicati per ½)

Ora scriviamo per esteso le nuove matrici. Con queste matrici ed il notebook di Mathematica usato prima si possono trovare gli approssimanti iterativi  $\beta^{(k)}$ , e da essi ricavare b<sup>(k)</sup>. Per il momento omettiamo i calcoli, perché li ripeteremo fra qualche pagina nell'illustrare il raffinamento progressivo.

$$\mathbf{N}' = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -0.29 & 1 & 0 & 0 & 0 & 0 \\ -0.29 & -(1/2)0.15 & 1 & 0 & 0 & 0 \\ -0.29 & -(1/2)0.12 & -(1/2)0.15 & 1 & 0 & 0 \\ -0.29 & -(1/2)0.15 & -(1/2)0.12 & -(1/2)0.15 & 1 & 0 \\ -(1/2)0.42 & -(1/4)0.145 & -(1/4)0.145 & -(1/4)0.145 & -(1/4)0.145 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -0.29 & 1 & 0 & 0 & 0 & 0 \\ -0.29 & -0.075 & 1 & 0 & 0 & 0 \\ -0.29 & -0.075 & 1 & 0 & 0 & 0 \\ -0.29 & -0.075 & -0.06 & -0.075 & 1 & 0 \\ -0.21 & -0.03625 & -0.03625 & -0.03625 & 1 \end{pmatrix}$$

	(	1		0		0		(	)	0	0)
		-0.2	29	1		0		(	)	0	0
N! _		-0.2	29	-(1/2)0	).15	1		(	)	0	0
IN =		-0.2	29	-(1/2)0	).12	-(1/2)0	.15	1		0	0
		-0.2	29	-(1/2)0	).15	-(1/2)0	.12	-(1/2	)0.15	1	0
	(–(	1/2	)0.42	-(1/4)0	.145 -	-(1/4)0.	145 -	-(1/4)	0.145	-(1/4)0.145	51)
(	1		0	0		0	(	)	0)		
-0	).29		1	0		0	(	)	0		
-0	).29	_	0.075	1		0	(	)	0		
= -0	).29	_	-0.06	-0.07	75	1	(	)	0		
-0	).29	_	0.075	-0.0	6 -	-0.075	1	l	0		
(-(	).21	-0	.03625	-0.036	525 —	0.03625	-0.0	3625	1)		
			$\begin{pmatrix} 0 & 0 \end{pmatrix}$	.145	0.145	0.14	15	0.14	5	(1/4)0.42	
			0	0 (1,	(2)0.15	5 (1/2)0	).12	(1/2)0	).15	(1/8)0.29	
	P		0	0	0	(1/2)0	).15	(1/2)	).12	(1/8)0.29	
	1	_	0	0	0	0		(1/2)0	).15	(1/8)0.29	
			0	0	0	0		0		(1/8)0.29	
			(0	0	0	0		0		0 )	
		(0)	0.145	0.145	0.145	0.145	0.10	05			
		0	0	0.075	0.06	0.075	0.030	525			
	_	0	0	0	0.075	0.06	0.030	525			
	_	0	0	0	0	0.075	0.03	525			
		0	0	0	0	0	0.030	525			
		0	0	0	0	0	0				

#### 3. Raffinamento progressivo

Come già spiegato nella sottosezione Interpretazione fisica del procedimento di Southwell: luce emessa nell'ambiente, un vantaggio del metodo di che stiamo presentando (Southwell con raffinamento) è che si presta a riconsiderare il sistema lineare della radiosità dal punto di vista nel quale si considera l'energia irradiata da ogni elemento nell'ambiente invece che ricevuta dall'ambiente. In questo modo si ottengono cospicui guadagni di tempo e di allocazione di memoria nel calcolo dei fattori di forma, come visto nella sottosezione Vantaggi numerici dell'approccio ad emissione di energia nell'ambiente. Poiché però i fattori di forma li abbiamo già calcolati, tralasciamo questo aspetto e ci rivolgiamo alla estrapolazione del punto di equilibrio tramite raffinamento progressivo degli approssimanti. Questo richiede una stima della energia residua da irradiare, pesata con le aree degli elementi, presentata nella sottosezione *Correzione ambientale della luminosità nel corso delle iterazioni, alla quale facciamo riferimento per le formule che stiamo per citare.* 

Rammentiamo gli ingredienti di questa stima. Gli approssimanti della radiosità vengono incrementati di un termine che compensa l'energia ancora da irradiare in seguito ad ulteriori riflessioni, dato da (37):

 $b_{i}^{(k)} = b_{i}^{(k)} + \rho_{i} \Phi$ 

dove  $\Phi$  è una stima del flusso per unità di area dell'energia totale non irradiata, dato da (<u>36</u>):  $\Phi = RH/A$ .

In quest'ultima equazione, A è la somma totale delle aree degli elementi (nel nostro caso A=16, perché pavimento e soffitto hanno area 4 e le pareti hanno area 2), R è la riflettività media ambientale che tiene conto di un numero arbitrario di interriflerssioni della luce, e che si ottiene a partire dalla media pesata della riflettività degli elementi,  $\rho_{media} = \Sigma_{i=1,...,n} \rho_i A_i / \Sigma_{i=1,...,N} A_i$ , grazie a (<u>34</u>):

R = 1/(1-  $\rho_{media})$  .

Infine,  $H = H^{(k)}$  è una stima dell'energia totale ancora non irradiata nell'ambiente dopo

 $\mathsf{H}^{(\mathsf{k})} = \Sigma_{\mathsf{i}=1,\ldots,\mathsf{n}} \, \mathsf{r}_{\mathsf{i}}^{(\mathsf{k})}$ 

con il termine residuo dell'energia dato da (30):

$$r_i{}^{(k)} = \beta_i{}^{(k+1)}$$
 -  $\beta_i{}^{(k)}$  .

che è equivalente a r<sup>(k)</sup> =  $\epsilon - K \beta^{(k)}$ 

Nel nostro esempio, il pavimento ha riflettività ¼ ed area 4, le altre pareti ed il soffitto hanno riflettività ½ ed area rispettivamente 2 e 4, quindi  $\rho_{media} = (4 \times \frac{1}{4} + (4 \times 2 + 4) \times \frac{1}{2})/16 = 7/16 = 0.4375$ . Pertanto R = 16/9 = 1.778, e diventa facile trovare b<sub>i</sub><sup>'(k)</sup> da (37): basta generare le iterazioni  $\beta_i^{(k)}$  col metodo di Southwell, calcolare r<sub>i</sub><sup>(k)</sup> da (30), quindi H<sup>(k)</sup> da (35) ed infine  $\Phi = \Phi^{(k)}$  da (36). Tutto questo si ottiene quindi con questa semplice modifica del notebook con cui abbiamo generato le iterazioni del metodo di Southwell:

```
A={4,2,2,2,2,4};
rho={1/4,1/2,1/2,1/2,1/2,1/2};t=0.03625;t1=0.075;t2=0.06;t3=0.21;t4=0.2
9;
```

```
t5=0.1005;
P=\{\{0,t4/2,t4/2,t4/2,t4/2,t5\},\{0,0,t1,t2,t1,t,t2,t1,t,t2,t1,t,t2,t1,t,t2,t1,t,t2,t1,t,t2,t1,t,t2,t1,t,t2,t1,t,t1,t2,t1,t,t2,t1,t,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t1,t2,t
},{0,0,0,t1,t2,t},{0,0,0,0,t1,t},{0,0,0,0,t},{0,0,0,0,0}};
t1,1,0,0, {-t4, -t1, -t2, -t1, 1, 0},
\{-t3, -t, -t, -t, -t, 1\}\};
K=NN-P;
e = \{0, 0, 0, 0, 0, 0, 1\};
Ninv=Inverse[NN]; TraditionalForm[NN];
b = \{0, 0, 0, 0, 0, 0\};
eps=e; beta=b;bprime=b;
AreaTot=0;
For [i=1,i<7,i++,AreaTot=AreaTot+A[[i]]];</pre>
rhoMedia=0;
For [i=1,i<7,i++,rhoMedia=rhoMedia+rho[[i]] A[[i]] ];</pre>
rhoMedia=rhoMedia/AreaTot;
R=1/(1-rhoMedia);
kk=0;maxr=1;
While[(maxr>0.001),
       r=N[eps-K.beta];
      maxr=Max[r[[1]], r[[2]], r[[3]], r[[4]], r[[5]],r[[6]]];
      For [i=1, i<7, i++,
             If [(r[[i]]==maxr),i0=i,Continue]
             ];
       beta[[i0]]=N[(Ninv .P)[[i0]]. beta+ Ninv[[i0]] .eps];
      For [i=1,i<7,i++,b[[i]]=beta[[i]]/A[[i]]];</pre>
      H=0;
      For [i=1,i<7,i++,H=H+r[[i]]];</pre>
      Phi=R H/AreaTot;
      For [i=1,i<7,i++,bprime[[i]]=b[[i]]+ rho[[i]] Phi];</pre>
      Print[kk]; Print[N[b]];Print[N[bprime]];
         kk++]
```

Alla trentatreesima iterazione i due metodi danno lo stesso risultato, e da quel momento in poi le prime quattro cifre decimali si stabilizzano. Il metodo numerico basato sul raffinamento è ad ogni iterazione considerevolmente più vicino al risultato finale di quanto non sia il metodo di Southwell. Nonostante questo, entrambi i metodi impiegano lo stesso numero di iterazioni a stabilizzarsi, ma già alla sesta iterazione il metodo del raffinamento fornisce una buona approssimazione del risultato finale. A differenza del metodo di Southwell, il raffinamento non produce una approssimazione monotona crescente, anzi, dopo una crescita notevolissima al primo passo, gli approssimanti del raffinamento convergono in maniera decrescente, ma ad ogni passo essi sono più grandi di quelli di Southwell, come deve essere a causa di (<u>37</u>).

Ecco le iterazioni successive del metodo di Southwell ricavate da Mathematica. Per ogni iterazione, il primo vettore è l'approssimante vero, il secondo l'approssimante aumentato nel senso del raffinamento progressivo:

```
1
{0.,0.,0.,0.,0.,0.25}
{0.0277778,0.0555556,0.0555556,0.0555556,0.305556}
2
```

 $\{0.025125, 0., 0., 0., 0., 0.25\}$ {0.0319444,0.0136389,0.0136389,0.0136389,0.0136389,0.263639} 3 {0.025125,0.,0.,0.,0.040056,0.25} {0.0329774,0.0157047,0.0157047,0.0157047,0.0557607,0.265705} 4 {0.025125,0.,0.,0.0425984,0.040056,0.25} {0.0316227,0.0129954,0.0129954,0.0555938,0.0530514,0.262995} 5 {0.025125,0.,0.0449013,0.0425984,0.040056,0.25} {0.030182,0.0101141,0.0550154,0.0527125,0.05017,0.260114} 6 {0.025125,0.0469889,0.0449013,0.0425984,0.040056,0.25} {0.0286635,0.0540659,0.0519783,0.0496754,0.047133,0.257077} {0.0377795,0.0469889,0.0449013,0.0425984,0.040056,0.25} {0.0397288,0.0508876,0.0488001,0.0464971,0.0439547,0.253899} 8 {0.0377795,0.0469889,0.0449013,0.0425984,0.040056,0.261529} {0.0402491,0.0519281,0.0498405,0.0475376,0.0449952,0.266468} 9 {0.0377795,0.0469889,0.0449013,0.0425984,0.0522409,0.261529} {0.0392826,0.0499951,0.0479076,0.0456046,0.0552471,0.264535} 10 {0.0377795,0.0469889,0.0449013,0.0530102,0.0522409,0.261529} {0.0388705,0.049171,0.0470834,0.0551923,0.0544229,0.263711} 11 {0.0377795,0.0469889,0.0535775,0.0530102,0.0522409,0.261529} {0.0385184,0.0484667,0.0550553,0.054488,0.0537187,0.263006} 12 {0.0412054,0.0469889,0.0535775,0.0530102,0.0522409,0.261529} {0.0416509,0.0478799,0.0544684,0.0539012,0.0531318,0.26242} 13 {0.0412054,0.053977,0.0535775,0.0530102,0.0522409,0.261529} {0.0417917,0.0551496,0.0547501,0.0541829,0.0534135,0.262701} 14 {0.0412054,0.053977,0.0535775,0.0530102,0.0522409,0.262703} {0.0415554,0.0546769,0.0542774,0.0537102,0.0529409,0.263403} 15 {0.0412054,0.053977,0.0535775,0.0530102,0.0547526,0.262703} {0.0414569,0.05448,0.0540805,0.0535133,0.0552557,0.263206} 16 {0.0420122,0.053977,0.0535775,0.0530102,0.0547526,0.262703} {0.0421788,0.0543101,0.0539107,0.0533434,0.0550858,0.263036} 17 {0.0420122,0.053977,0.0535775,0.0549109,0.0547526,0.262703} {0.0422119,0.0543765,0.053977,0.0553104,0.0551521,0.263103} 18 {0.0420122,0.053977,0.0550148,0.0549109,0.0547526,0.262703} {0.0421476,0.0542479,0.0552858,0.0551819,0.0550235,0.262974} 19 {0.0420122,0.0550806,0.0550148,0.0549109,0.0547526,0.262703} {0.042099,0.0552543,0.0551886,0.0550846,0.0549263,0.262877} 20 {0.0423342,0.0550806,0.0550148,0.0549109,0.0547526,0.262703} {0.0423837,0.0551797,0.0551139,0.05501,0.0548517,0.262802} 21

{0.0423342,0.0550806,0.0550148,0.0549109,0.0551788,0.262703} {0.042397,0.0552062,0.0551404,0.0550365,0.0553043,0.262829} 22 {0.0423342,0.0550806,0.0550148,0.0549109,0.0551788,0.262898} {0.0423825,0.0551773,0.0551116,0.0550076,0.0552755,0.262995} 23 {0.0423342,0.0550806,0.0550148,0.0552347,0.0551788,0.262898} {0.0423662,0.0551446,0.0550789,0.0552988,0.0552428,0.262962} 24 {0.0423342,0.0550806,0.0552528,0.0552347,0.0551788,0.262898} {0.0423552,0.0551227,0.055295,0.0552768,0.0552209,0.26294} 25 {0.0424254,0.0550806,0.0552528,0.0552347,0.0551788,0.262898} {0.0424384,0.0551066,0.0552789,0.0552607,0.0552048,0.262924} 26 {0.0424254,0.0552633,0.0552528,0.0552347,0.0551788,0.262898} {0.0424422,0.0552968,0.0552864,0.0552683,0.0552123,0.262932} 27 {0.0424254,0.0552633,0.0552528,0.0552347,0.0552819,0.262898} {0.042436,0.0552845,0.055274,0.0552559,0.055303,0.262919} 28 {0.0424461,0.0552633,0.0552528,0.0552347,0.0552819,0.262898} {0.0424532,0.0552775,0.055267,0.0552489,0.0552961,0.262912} 29 {0.0424461,0.0552633,0.0552528,0.0552884,0.0552819,0.262898} {0.0424541,0.0552792,0.0552687,0.0553043,0.0552978,0.262914} 30 {0.0424461,0.0552633,0.0552528,0.0552884,0.0552819,0.262923} {0.0424523,0.0552756,0.0552651,0.0553006,0.0552941,0.262935} 31 {0.0424461,0.0552633,0.0552948,0.0552884,0.0552819,0.262923} {0.0424502,0.0552714,0.0553029,0.0552965,0.05529,0.262931} 32 {0.0424461,0.0552967,0.0552948,0.0552884,0.0552819,0.262923} {0.0424488,0.055302,0.0553001,0.0552936,0.0552871,0.262928} 33 {0.042458,0.0552967,0.0552948,0.0552884,0.0552819,0.262923} {0.0424595,0.0552997,0.0552978,0.0552914,0.0552849,0.262926}

Queste approssimazioni nelle quali ciascuna faccia e' un elemento sono terribilmente rudimentali. Per una ragionevole gradazione delle luci, una faccia dovrebbe scomporsi in molte centinaia di elementi. Ma la mole di calcoli crescerebbe notevolmente. Supponiamo ad esempio di considerare la modellazione, ancora troppo rudimentale, in cui ciascuna faccia si scompone in 25 elementi (quindi l'elemento centrale del soffitto è la lampada: finalmente il soffitto non emette più luce uniformemente da tutta la propria superficie!). Ci sono in tutto 25x6 = 150 elementi. Quindi le matrici sono di dimensione 150. Quanti fattori di forma devono essere calcolati? A prima vista sembrerebbe di doverne calcolare  $150^2 = 22500$ , un numero già elevatissimo. Ma invece sono di meno, perché molti coincidono. Ad esempio, data un elemento, tutti i fattori di forma verso elementi di un'altra parete dipendono solo dallo spostamento relativo fra le due, quindi sono solo 25; inoltre, l'insieme dei fattori di forma da elementi di una parete a elementi di una parete adiacente rimane identico
indipendentemente da quale delle quattro pareti adiacenti si sceglie. Quindi basta calcolare 25 fattori di forma per elementi su pareti adiacenti ed altri 25 per elementi su pareti antistanti: in tutto 50. Pero' poi bisogna scrivere il codice in maniera intelligentemente appropriata per fare in modo che tenga conto in modo automatico di quali fattori di forma si ripetono.

Lasciamo al lettore il calcolo di quale mole di lavoro sarebbe richiesta se anziché suddividere ogni faccia in 25 elementi la considerassimo un unico elemento ma provvedessimo a sottostrutturarla in 25 <u>sottoelementi</u>.

# ESERCIZI FINALI SULLA RADIOSITÀ

**Esercizio 1.** Una stanza di forma cubica ha una luce di intensità 1 sul soffitto ed una finestra che lascia entrare luce con intensità 1 sulla parete sud. Il fattore di forma fra pavimento e soffitto vale 0.2. Le pareti senza finestra ed il pavimento sono neri (coefficiente di riflettività zero); il soffitto e la parete sud hanno coefficiente di riflettività  $\frac{1}{2}$ .

Calcolare le prime due iterazioni del metodo di Jacobi e del metodo di Gauss-Seidel per il calcolo delle radiosità, prendendo le pareti, il pavimento ed il soffitto come elementi.

**Esercizio 2.** Una scena consiste dell'interno di una semisfera di raggio 2, al centro della quale c'è una cabina semisferica di raggio 1. La scena è l'intercapedine fra le due semisfere. La luce viene da un'apertura sulla semisfera esterna, con potenza 1. La semisfera esterna ha fattore di riflettività 2/3. Quella interna ha fattore di riflettività 1/3. Il pavimento ha fattore di riflettività 0. Il fattore di forma dal pavimento alla semisfera esterna è 7/8.

• Trovare tutti gli altri fattori di forma.

• Trovare le prime tre iterazioni della radiosità  $(b^{(1)}, b^{(2)} e b^{(3)})$  con il metodo di Jacobi.

- Trovare b<sup>(1)</sup> con il metodo di Gauss-Seidel.
- Trovare la seconda iterazione del metodo di Southwell.

**Esercizio 3.** La scena consiste dell'interno dell'astronave Enterprise, fatta a ciambella, di metallo con fattore di riflettività 1/2. Tutte le luci sono spente a causa di un attacco dei Klingon, tranne che nel pannello di comando, che emette luce con potenza 1. I membri dell'equipaggio, impavidi nonostante l'attacco, rimasti senza computers funzionanti, alla fioca luce emessa dal pannello di comando sono impegnati a calcolare a mano l'illuminazione che si distribuisce dal pannello di comando all'astronave (un modo come un altro per scoprire se nella corazza si è formato qualche buco, che ovviamente avrebbe riflettività 0 invece che ½). Essi immaginano l'interno dell'astronave separato in quattro settori di 90 gradi tramite piani perpendicolari che si incontrano sull'asse centrale della ciambella (quello rispetto al quale la ciambella è invariante per rotazione). Così l'astronave si scompone in quattro elementi, al centro di una delle quali è il pannello di controllo. La ciambella è sufficientemente sottile perché da ciascun

elemento si riescano a vedere solo quelle adiacenti. Il fattore di forma di ogni elemento su sé stesso è ½. Purtroppo, quando comincia il calcolo l'intero elemento opposto al pannello di comando è stata demolito dai Klingon, e quindi è diventato nero (il colore dello spazio esterno). I membri dell'equipaggio, prima di essere uccisi dai Klingon, sono in grado di:

- Trovare tutti i fattori di forma.
- Trovare le prime tre iterazioni della radiosità  $(b^{(1)}, b^{(2)} e b^{(3)})$  con il metodo di Jacobi.
- Trovare b<sup>(1)</sup> con il metodo di Gauss-Seidel.
- Trovare la seconda iterazione del metodo di Southwell.

Lo siete anche voi, prima di essere uccisi da questo esercizio? Assumete di sapere che l'elemento opposto al pannello di comando sia ormai nero.

**Esercizio 4.** Una scena consiste dell'interno di una torre cilindrica di raggio r ed altezza m. Sia  $\alpha$  il fattore di forma dal pavimento alla parete cilindrica. È possibile determinare tutti i fattori di forma in funzione di  $\alpha$ ? Se non lo è, mostrare perché; se lo è, per quali r e m la scelta  $\alpha = \frac{1}{2}$  è compatibile con il problema? Se lo è per r = m (m come definito sopra), scrivere in tal caso il sistema lineare della radiosità assumendo che il pavimento sia nero, la parete ed il soffitto bianchi (ossia con fattore di riflettività 1) ed il soffitto emetta luce a potenza 1, e calcolare le prime tre iterazioni per la radiosità con il metodo di Jacobi.

Svolgimento. Supponiamo, per maggiore generalità, che il raggio sia r (in seguito porremo r=m). Siano  $\beta$  il fattore di forma da pavimento a soffitto (e viceversa),  $\gamma$  quello da parete a pavimento (o soffitto), e  $\sigma$  quello dalla parete e sè stessa: non ci sono altri fattori di forma. L'area della parete è  $2\pi$ rm, quella del pavimento è  $\pi$ r<sup>2</sup>, quindi, per la regola di reciprocità,

$$\gamma = \frac{r}{2m}\alpha$$
 (1)

Le identità relative alla conservazione dell'angolo solido totale sono  $\alpha + \beta = 1$  e  $2\gamma + \sigma = 1$ . Grazie all'identità (1), queste ora possono essere riscritte come

$$\sigma = 1 - \frac{r}{m} \alpha$$
(2)  
$$\beta = 1 - \alpha$$
(3)

e quindi tutti i fattori di forma sono funzioni di  $\alpha$ . Affinché il risultato sia accettabile occorre che tutti i fattori di forma siano positivi e non superiori a 1. Quindi, per (2), la scelta  $\alpha = \frac{1}{2}$  è compatibile se e solo se r/m  $\leq$  2. In particolare lo è per r = m. In tal caso, numerando con 1 il pavimento, 2 la parete e 3 il soffitto, e ponendola matrice M della radiosità diventa

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 \\ -\rho\gamma & 1-\rho\sigma & -\rho\gamma \\ -\rho\beta & -\rho\alpha & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{\rho\alpha}{2} & 1-\rho(1-\alpha) & -\frac{\rho\alpha}{2} \\ -\rho(1-\alpha) & -\rho\alpha & 1 \end{pmatrix}$$

Ora spezziamo M = N - P, con

$$\mathbf{N} = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{\rho\alpha}{2} & 1 - \rho(1 - \alpha) & 0 \\ -\rho(1 - \alpha) & -\rho\alpha & 1 \end{pmatrix} \qquad \mathbf{e} \qquad \mathbf{P} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & \frac{\rho\alpha}{2} \\ 0 & 0 & 0 \end{pmatrix}$$

Qui N è del tipo N =  $\begin{pmatrix} 1 & 0 & 0 \\ x & a & 0 \\ z & y & 1 \end{pmatrix}$ , e si verifica subito che la sua inversa è

$$\mathbf{N}^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{\mathbf{x}}{\mathbf{a}} & \frac{1}{\mathbf{a}} & 0 \\ \frac{\mathbf{xy} \cdot \mathbf{az}}{\mathbf{a}} & -\frac{\mathbf{y}}{\mathbf{a}} & 1 \end{pmatrix}$$

quindi nel nostro caso

$$\mathbf{N}^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2}\rho^{2}\alpha^{2} + a\rho(1-\alpha) & & \\ \frac{1}{1-\rho(1-\alpha)} & \frac{1}{1-\rho(1-\alpha)} & 0 \\ \frac{\rho\alpha}{1-\rho(1-\alpha)} & \frac{\rho\alpha}{1-\rho(1-\alpha)} & 1 \end{pmatrix} N^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2}\frac{\rho\alpha}{1-\rho(1-\alpha)} & \frac{1}{1-\rho(1-\alpha)} & 0 \\ \frac{\rho(1-\alpha) + \frac{1}{2}\rho^{2}(1-2\alpha)}{1-\rho(1-\alpha)} & \frac{\rho\alpha}{1-\rho(1-\alpha)} & 1 \end{pmatrix}$$

Pertanto si ha

$$\mathbf{N}^{-1}\mathbf{P} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} \frac{\rho\alpha}{1-\rho(1-\alpha)} \\ 0 & 0 & \frac{1}{2} \frac{\rho^2 \alpha^2}{1-\rho(1-\alpha)} \end{pmatrix}$$

La potenza creata per unità di area è il vettore  $\mathbf{e} = (0,0,1)$ . Partendo dall'approssimazione iniziale  $b^0 = (0,0,0)$  si ottiene  $b^{(1)} = N^{-1}\mathbf{e} = (0,0,1)$  (la terza colonna della matrice  $N^{-1}$ !). Quindi  $N^{-1}P$   $b^{(1)}$  è la terza colonna della matrice  $N^{-1}P$ , e pertanto

$$b^{(2)} = N^{-1} P b^{(1)} + N^{-1} e = (N^{-1} P + Id) e = \frac{1}{2} \begin{pmatrix} 0 \\ \frac{\rho \alpha}{1 - \rho(1 - \alpha)} \\ \frac{\rho^2 \alpha^2}{1 - \rho(1 - \alpha)} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{1}{2} \frac{\rho \alpha}{1 - \rho(1 - \alpha)} \\ 1 + \frac{1}{2} \frac{\rho^2 \alpha^2}{1 - \rho(1 - \alpha)} \end{pmatrix}$$

Allora

$$b^{(3)} = N^{-1} P b^{(2)} + N^{-1} e = N^{-1} P \begin{pmatrix} 0 \\ \frac{1}{2} \frac{\rho \alpha}{1 - \rho(1 - \alpha)} \\ 1 + \frac{1}{2} \frac{\rho^2 \alpha^2}{1 - \rho(1 - \alpha)} \end{pmatrix} + e$$

$$= \begin{pmatrix} 0 \\ \frac{1}{2} \frac{\rho \alpha}{1 - \rho(1 - \alpha)} + \frac{1}{4} \frac{\rho^3 \alpha^3}{(1 - \rho(1 - \alpha))^2} \\ \frac{1}{2} \frac{\rho^2 \alpha^2}{1 - \rho(1 - \alpha)} + \frac{1}{4} \frac{\rho^4 \alpha^4}{(1 - \rho(1 - \alpha))^2} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{1}{2} \frac{\rho \alpha}{1 - \rho(1 - \alpha)} + \frac{1}{4} \frac{\rho^3 \alpha^3}{(1 - \rho(1 - \alpha))^2} \\ 1 + \frac{1}{2} \frac{\rho^2 \alpha^2}{1 - \rho(1 - \alpha)} + \frac{1}{4} \frac{\rho^4 \alpha^4}{(1 - \rho(1 - \alpha))^2} \end{pmatrix}$$

Nel nostro caso,  $\rho = \alpha = \frac{1}{2}$ , e quindi

**Esercizio 5.** Consideriamo una stanza cubica di lato 10 metri nella quale, per semplicità, assumiamo uguali tutti i fattori di forma non nulli. Nella stanza stendiamo un telo verticale, dal lato che congiunge due pareti a quello opposto (ossia quello che congiunge le due pareti opposte). Il telo divide la stanza in due parti uguali: consideriamo solo una delle due. Il telo e le pareti di questa metà della stanza sono elementi. Numeriamo il soffitto (triangolare) con 1, le due pareti (quadrate) con 2 e 3, il pavimento (triangolare) con 4, il telo con 5. Il telo è nero, le pareti ed il soffitto sono bianchi.

a. Supponiamo che il fattore di forma da soffitto a pavimento sia 1/4, e lo stesso valga per il fattore di forma da parete a parete e da soffitto a parete.

Questi dati sono sufficienti per calcolare tutti i fattori di forma? Sono compatibili fra loro? Se sì calcolarli, se no spiegare perché.

b. Come cambia la risposta se si chiede solo che il fattore di forma da soffitto a pavimento e quello da parete a parete siano ¼?

*Svolgimento*. Poiché gli elementies sono piane, i fattori di forma da un elemento a sé stessa sono nulli. Per simmetria, ci sono solo cinque fattori di forma indipendenti: da triangolo (soffitto) a quadrato (parete intera), da triangolo a triangolo (soffitto a pavimento), da rettangolo (telo) a triangolo, da quadrato a rettangolo e da quadrato a quadrato: i fattori di forma in senso inverso si ricavano da questi tramite l'equazione di reciprocità A<sub>i</sub>  $F_{i-j} = A_j F_{j-i}$ . Indichiamo questi cinque fattori di forma con  $F_{TQ}$ ,  $F_{TT}$ ,  $F_{RT}$ ,  $F_{QR}$  e  $F_{QQ}$ , rispettivamente. Osservando che le aree di questi elementi verificano

$$A_{\rm Q} = 2 A_{\rm T}$$
$$A_{\rm R} = 2^{1/2} A_{\rm Q}$$

dalla relazione di reciprocità otteniamo

$$F_{QT} = \frac{1}{2} F_{TQ}$$
$$F_{TR} = 2^{1/2} F_{RT}$$
$$F_{QR} = 2 F_{RQ}$$

Ora applichiamo la relazione di conservazione dell'energia (ovvero dell'angolo solido),  $\Sigma_i F_{i-j} = 1$ . Si ottiene il sistema lineare:

$$F_{QQ} + F_{QR} + 2 F_{QT} = 1$$
  
 $F_{TT} + F_{TR} + 2 F_{TQ} = 1$   
 $2 F_{RT} + 2 F_{RQ} = 1$ 

Sostituendo in queste equazioni le precedenti identità ora abbiamo

$$F_{QQ} + 2 F_{RQ} + F_{TQ} = 1$$
  
 $F_{TT} + 2^{1/2} F_{RT} + 2 F_{TQ} = 1$   
 $2 F_{RT} + 2 F_{RQ} = 1$ 

Per semplicità scriviamo a =  $F_{QQ}$ , b =  $F_{TT}$ , c =  $F_{RQ}$ , d =  $F_{TQ}$ , e =  $F_{RT}$ . Abbiamo tre equazioni in cinque incognite a, b, c, d, e i cui valori però sono vincolati ad appartenere all'intervallo [0,1]. Dall'ultima equazione ricaviamo c =  $\frac{1}{2}$  - e. Sostituendo nella prima otteniamo a+(1-2e)+d = 1, ossia

a+d=2e.

La seconda equazione invece diventa  $b+2^{1/2}(\frac{1}{2} - e)+2d=1$ , ossia

 $b+2d-2^{1/2}e=1-1/2^{1/2}$ .

In tal modo restano due equazioni in quattro incognite. L'ipotesi della parte (a) del problema è a=b=d=1/4. Pertanto tre delle quattro incognite vengono fissate, e rimangono due equazioni in una sola incognita, che sono le seguenti:

$$\frac{1}{2} = a + d = 2e$$
, ossia  $e = 1/4$ ,

е

$$b+2d-2^{1/2}e = 1-1/2^{1/2}$$
, ossia  $\frac{3}{4}-2^{1/2}/4=1-1/2^{1/2}$ . ossia  $3-2^{1/2}=4-2^{3/2}$ .

Quest'ultima identità è falsa, quindi i dati sono sufficienti ma incompatibili fra loro. Nella parte (b) si richiede solo a=d=1/4. Pertanto restano due equazioni in due incognite, le seguenti: la prima è e=1/4, come prima, mentre la seconda diventa b+1/2 -  $2^{1/2}/4 = 1-1/2^{1/2}$ , ossia b= $2^{1/2}/4 + 1/2 - 1/2^{1/2}$ , Quindi si ottiene un valore di b dell'ordine di 0.15, ossia fra 0 e 1. I dati sono compatibili.

**Esercizio 6.** La scena consiste di una stanza cubica di lato 1 il cui pavimento è stato sostituito da una piramide il cui vertice alto tocca il centro del soffitto. Le quattro facce della piramide sono numerate da 1 a 4, le pareti da 5 a 8 (5 antistante a 1, 6 antistante a 2 e così via), ed il soffitto ha il numero 9. La piramide non riflette la luce, però la emette con potenza per unità di area uguale a 1. Le pareti 5 e 7 sono nere. Tutti gli altri coefficienti di riflettività sono ½ tranne quello del soffitto che è 1/4. I fattori di forma da ciascuna faccia della piramide alla parete antistante ed al soffitto valgono entrambi a, ed il fattore di forma da ciascuna parete al soffitto vale b, dove a e b sono due numeri fra 0 e 1.

1. Qual'è l'area di ciascuna faccia della piramide?

2. Se a = b, i dati sono sufficienti a trovare tutti i fattori di forma? Se sì, determinare i valori di a per cui i dati sono compatibili (ossia tali che tutti i fattori di forma siano compresi fra 0 e 1).

3. Se la risposta al punto precedente è negativa, assumiamo a diverso da b. Che condizioni dobbiamo imporre su a e b affinché i dati siano compatibili?

Se si sono trovate ai punti precedenti valori di a e b per i quali i dati sono compatibili, si scelga a piacere una tale coppia a e b (uguali fra loro, se la domanda al punto (2) aveva risposta positiva), e si risponda alle seguenti domande:

4. Trovare le prime due iterazioni della radiosità (b<sup>(1)</sup>, b<sup>(2)</sup>) con il metodo di Jacobi.

5. Trovare  $b^{(1)}$  con il metodo di Gauss-Seidel.

6. Trovare la seconda iterazione del metodo di Southwell.

**Esercizio 7.** Una stanza si compone di un cubo la cui faccia superiore è sostituita da un tetto fatto a forma di calotta a volta, con area doppia di quella delle facce del cubo. Il soffitto emette luce con potenza 1, il pavimento è nero, le pareti ed il soffitto hanno fattore di riflettività  $\frac{1}{2}$ . Il fattore di forma da una parete a quella opposta frontalmente vale  $\alpha$ .

i) La scelta di  $\alpha$  = 1/5 è compatibile con i dati del problema? Se sì prendere  $\alpha$  = 1/5, se no scegliere un altro valore di  $\alpha$  compatibile.

ii) Calcolare la prima iterazione della radiosità con il metodo di rilassamento di Jacobi.

iii) Calcolare le prime due iterazioni del metodo di Jacobi.

## Svolgimento.

Chiamiamo  $F_{lat}$  il fattore di forma da una faccia quadrata ad una faccia quadrata adiacente, e  $F_{front}$  quello da una faccia quadrata a quella frontale. La volta, vista dalle altre facce, copre lo stesso angolo solido che coprirebbe la faccia quadrata superiore del cubo, e quindi il fattore di forma da faccia laterale a volta vale ancora  $F_{lat}$  e quello da pavimento a volta vale  $F_{front}$ . Poiché la volta ha area doppia delle altre facce, dalla relazione di reciprocità  $A_i F_{i-j} = A_j F_{ji-i}$  otteniamo che il fattore di forma da volta a faccia laterale vale 1/2  $F_{lat}$  e quello da volta a faccia laterale vale 1/2  $F_{lat}$  e quello da volta a pavimento vale 1/2  $F_{front}$ . Infine, la volta è convessa invece che piana, e quindi ha un fattore di forma non nullo  $F_v$  verso sè stessa. Allora la conservazione dell'energia, ossia il fatto che l'angolo solido totale sia il 100% (ovvero 1), porta alle seguenti identità:

$4 F_{lat} + F_{front} = 1$	(angolo	solido	visto	dal	pavimento	0	da	ciascuna
delle pareti)								
$F_v + 2F_{lat} + \frac{1}{2}F_{front} = 1$	(angolo	solido v	visto da	alla v	/olta)			

Questo sistema lineare ha due equazioni e tre incognite, di cui una,  $F_{front}$ , è posta uguale ad un parametro fisso  $\alpha$  compreso fra 0 e 1. Allora si può ricavare  $F_{lat}$  dalla prima equazione e  $F_{v}$  dalla seconda, ma occorre verificare per quali valori di questi due risultati rimangono compresi fra 0 e 1. Si trova:

$$F_{lat} = (1 - \alpha)/4$$

sempre compreso fra 0 e 1 se lo è, e

$$F_v = 1 - (1 - \alpha)/2 - \alpha/2 = \alpha$$

anch'esso compreso fra 0 e 1. Quindi per ogni valore ammissibile del parametro  $\alpha$  si trova una soluzione.

Numeriamo con 1 il soffitto, con 2,3,4,5 le quattro pareti laterali e con 6 il pavimento. Poiché tutti i coefficienti di riflettività valgono ½ tranne quello del pavimento che vale 0, la matrice M della radiosità è la seguente:

$\left(1-\frac{1}{2}\alpha\right)$	$(\alpha - 1)/16$	$(\alpha - 1)/16$	$(\alpha - 1)/16$	$(\alpha - 1)/16$	$-\frac{1}{4}\alpha$
$(\alpha - 1)/8$	1	$(\alpha - 1)/8$	$-\frac{1}{2}\alpha$	$(\alpha - 1)/8$	$(\alpha - 1)/8$
$(\alpha - 1)/8$	$(\alpha - 1) / 8$	1	$(\alpha - 1)/8$	$-\frac{1}{2}\alpha$	$(\alpha - 1)/8$
$(\alpha - 1)/8$	$-\frac{1}{2}\alpha$	$(\alpha - 1) / 8$	1	$(\alpha - 1)/8$	$(\alpha - 1)/8$
$(\alpha - 1)/8$	$(\alpha - 1)/8$	$-\frac{1}{2}\alpha$	$(\alpha - 1)/8$	1	$(\alpha - 1)/8$
0	0	0	0	0	1

Il vettore della potenza luminosa creata è E = (1,0,0,0,0,0), perché solo il soffitto emette luce. Il sistema lineare della radiosità, M**b**=**e**, ci dà il vettore delle radiosità dei sei elementi, ma osserviamo che la radiosità del pavimento è 0 perché il pavimento ha coefficiente di riflettività 0 (essendo nero) e non emette luce propria. Quindi possiamo scartare l'ultima incognita e ridurci al seguente problema lineare a dimensione cinque:

1	$(1-\frac{1}{2}\alpha)$	$(\alpha - 1)/16$	$(\alpha - 1)/16$	$(\alpha - 1)/16$	$(\alpha - 1)/16$	$(b_1)$	) (	(1)	
	$(\alpha - 1)/8$	1	$(\alpha - 1)/8$	$-\frac{1}{2}\alpha$	$(\alpha - 1)/8$	$b_2$		0	
	$(\alpha - 1)/8$	$(\alpha - 1)/8$	1	$(\alpha - 1)/8$	$-\frac{1}{2}\alpha$	$b_3$	=	0	
	$(\alpha - 1)/8$	$-\frac{1}{2}\alpha$	$(\alpha - 1)/8$	1	$(\alpha - 1)/8$	$b_4$		0	
	$(\alpha - 1)/8$	$(\alpha - 1)/8$	$-\frac{1}{2}\alpha$	$(\alpha - 1)/8$	1	$b_5$		0)	

Con abuso di notazione, continuiamo a denotare questo sistema lineare con M**b**=**e**. Scriviamo M=N-P, con

$$\mathbf{N} = \begin{pmatrix} 1 - \frac{1}{2}\alpha & 0 & 0 & 0 & 0\\ (\alpha - 1)/8 & 1 & 0 & 0 & 0\\ (\alpha - 1)/8 & (\alpha - 1)/8 & 1 & 0 & 0\\ (\alpha - 1)/8 & -\frac{1}{2}\alpha & (\alpha - 1)/8 & 1 & 0\\ (\alpha - 1)/8 & (\alpha - 1)/8 & -\frac{1}{2}\alpha & (\alpha - 1)/8 & 1 \end{pmatrix}$$

е

$$\mathbf{P} = \begin{pmatrix} 0 & (1-\alpha)/16 & (1-\alpha)/16 & (1-\alpha)/16 & (1-\alpha)/16 \\ 0 & 0 & (1-\alpha)/8 & \frac{1}{2}\alpha & (1-\alpha)/8 \\ 0 & 0 & 0 & (1-\alpha)/8 & \frac{1}{2}\alpha \\ 0 & 0 & 0 & 0 & (1-\alpha)/8 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Il rilassamento di Jacobi approssima la soluzione **b** con le seguenti iterazioni:

$$\mathbf{b}^{(k+1)} = \mathbf{N}^{-1}\mathbf{P}\mathbf{b}^{(k)} + \mathbf{N}^{-1}\mathbf{e}$$

Calcoliamo N<sup>-1</sup>, ad esempio con il metodo di Gauss. Se si lascia arbitrario il valore di  $\alpha$ , l'espressione di N<sup>-1</sup> è assai complicata e laboriosa: la riportiamo qui di seguito per comodità del lettore. La matrice N<sup>-1</sup> è triangolare inferiore con sulla diagonale i valori reciproci di quelli della matrice N: quindi

$$N_{11}^{-1} = \frac{1}{1 - \alpha/2} = \frac{2}{\alpha - 2}$$

e gli altri valori diagonali valgono 1. Per quanto riguarda i valori sotto la diagonale, si trova, riga per riga:

$$N_{21}^{-1} = \frac{\alpha - 1}{4(\alpha - 2)}$$

$$N_{31}^{-1} = \frac{9 - 10\alpha + \alpha^2}{64 - 32\alpha}; \quad N_{32}^{-1} = \frac{1 - \alpha}{8}$$

$$N_{41}^{-1} = \frac{-73 + 51\alpha + 21\alpha^2 + \alpha^3}{256(\alpha - 2)}; \quad N_{42}^{-1} = \frac{1 + 30\alpha + \alpha^2}{64}; \quad N_{43}^{-1} = \frac{1 - \alpha}{8}$$

$$N_{51}^{-1} = \frac{649 - 476\alpha - 226\alpha^2 + 52\alpha^3 + \alpha^4}{2048(\alpha - 2)}; \quad N_{52}^{-1} = \frac{65 - 3\alpha - 61\alpha^2 - \alpha^3}{512}; \quad N_{53}^{-1} = \frac{1 + 30\alpha + \alpha^2}{64}; \quad N_{54}^{-1} = \frac{1 - \alpha}{8}$$

Vediamo i risultati nel caso particolare  $\alpha$ =1/5, che è molto più agevole. Si trova

$$\mathbf{N}^{-1} = \begin{pmatrix} 1.111111 & 0 & 0 & 0 & 0 \\ 0.111111 & 1 & 0 & 0 & 0 \\ 0.122222 & 0.1 & 1 & 0 & 0 \\ 0.134444 & 0.11 & 0.1 & 1 & 0 \\ 0.147889 & 0.121 & 0.11 & 0.1 & 1 \end{pmatrix}$$

е

	0	0.05	0.05	0.05	0.05
	0	0	0.1	0.1	0.1
P =	0	0	0	0.1	0.1
	0	0	0	0	0.1
	0	0	0	0	0 )

Cominciamo con l'approssimazione iniziale  $\mathbf{b}^{(0)} = 0$ . Si ottiene:

$$\mathbf{b}^{(1)} = \mathbf{N}^{-1}\mathbf{e} = \begin{pmatrix} 1.111111\\ 0.111111\\ 0.122222\\ 0.134444\\ 0.147889 \end{pmatrix}$$

Allora, i risultati sono:

$$b^{(2)} = N^{-1}Pb^{(1)} + N^{-1}e = N^{-1}Pb^{(1)} + b^{(1)} = N^{-1} \begin{pmatrix} 0.0286481 \\ 0.0433204 \\ 0.0354302 \\ 0.0255288 \\ 0.0132927 \end{pmatrix} + \begin{pmatrix} 1.111111 \\ 0.111111 \\ 0.122222 \\ 0.134444 \\ 0.147889 \end{pmatrix} = \begin{pmatrix} 1.13976 \\ 0.154431 \\ 0.157652 \\ 0.159973 \\ 0.161182 \end{pmatrix}$$

Ovviamente, per simmetria, a parte il primo valore gli altri dovrebbero essere uguali, ma due sole iterazioni sono poche. Con tre iterazioni si trova

$$b^{(3)} = \begin{pmatrix} 1.14629\\ 0.16251\\ 0.162996\\ 0.163298\\ 0.163509 \end{pmatrix}$$

e con quattro

$$b^{(4)} = \begin{pmatrix} 1.14735\\ 0.163715\\ 0.163787\\ 0.163836\\ 0.163869 \end{pmatrix}$$

e con cinque

$$\mathbf{b}^{(5)} = \begin{pmatrix} 1.14751\\ 0.1639\\ 0.16392\\ 0.163919\\ 0.163924 \end{pmatrix}$$

Le iterazioni convergono a

$$\mathbf{b}^{(\infty)} = \begin{pmatrix} 1.14754\\ 0.163934\\ 0.163934\\ 0.163934\\ 0.163934 \end{pmatrix}$$

*Problema:* quale è la differenza se si calcola la seconda iterazione del metodo di Gauss-Seidel? Spiegare le differenze passo per passo.

**Esercizio 8.** Una scena consiste dell'interno di un corridoio ad angolo, dato dall'unione di tre cubi di lato 1, il secondo ed il terzo dei quali si congiungono al primo su due pareti adiacenti di esso (ovviamente non ci sono pareti interne). La parete del terzo cubo opposta a quella dove esso si congiunge al primo cubo e' una finestra dalla quale entra luce a potenza per unita' di area pari a 1: ovviamente il suo coefficiente di riflettivita' e' zero. Le altre pareti non emettono luce; le pareti laterali ed il soffitto hanno coefficiente di riflettivita' ½, il pavimento 0. Il pavimento, il soffitto e le sei pareti laterali, una delle quali e' la finestra, formano altrettanti elementi. Numeriamo il pavimento con 0, poi procedendo in verso orario (visto dall'alto) la finestra con 1, le due pareti corte che formano un angolo ottuso con 3 e 4, l'ultima parete corta con 5, le due pareti lunghe con 6 e 7, il soffitto con 8.

1. Quanti sono i fattori di forma indipendenti?

2. Supponiamo di assegnare valore  $\alpha$  al fattore di forma dal pavimento al soffitto,  $\beta$  dalla parete lunga alla parete corta adiacente,  $\gamma$  dalla parete lunga alla parete lunga alla parete lunga alla parete lunga alla parete lunga all'altra parete lunga,  $\theta$  fra due pareti corte adiacenti che si vedono. Questi dati sono sufficienti a trovare tutti i fattori di forma? Spiegare.

3. Se lo sono, determinare gli altri fattori di forma, se no fissare il numero minimo indispensabile di altri fattori di forma necessario per poterli trovare tutti, e quindi trovare i restanti.

4. Attribuire ai fattori di forma valori numerici a piacere ma compatibili, e trovare le prime due iterazioni del metodo di Jacobi.

**Esercizio 9.** Come nell'<u>Esercizio 4 della sezione *Esercizi sul calcolo dei fattori di forma*, si consideri una ciotola K di forma emisferica di raggio 1 appoggiata nel suo punto più basso all'origine, ossia tangente all'origine al piano {z=0} e contenuta nel semispazio superiore { $z \ge 0$ }: quindi il bordo della ciotola è un cerchio J sul piano {z=1}. Poiché K è convessa, il fattore di forma dalla ciotola a sé stessa è positivo.</u>

- Si assuma che il fattore di forma F<sub>KK</sub> sia 0.75, e che la ciotola abbia coefficiente di riflettività ½. L'esterno della ciotola non riflette luce e si può approssimare quindi con uno sfondo semisferico nero (quindi la scena diventa l'interno di S). Al polo nord di S poniamo una luce puntiforme con potenza 1 (densità di energia per unità di tempo e area). Calcolare le prime quattro iterazioni del metodo di Jacobi per la radiosità.
- 2. Calcolare le prime tre iterazioni del metodo di Gauss-Seidel.
- 3. Calcolare le prime tre iterazioni del metodo di Southwell.

## Svolgimento.

*Parte 1.* Sia C l'emisfero superiore nero. Ovviamente  $F_{CC} = F_{KK} = 0.75$ , e per la conservazione del'angolo solido totale  $F_{KC} = F_{CK} = 0.25$ . Numeriamo l'elemento K con il numero 1 e l'elemento C con 2. Il vettore **e** della potenza creata è **e**=(0,1). La matrice della radiosità è

$$\mathbf{M} = \begin{pmatrix} 1 - \frac{1}{2} 0.75 & -\frac{1}{2} 0.25 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{5}{8} & -\frac{1}{8} \\ 0 & 1 \end{pmatrix}$$

Decomponiamo M = N - P, con

$$\mathbf{N} = \begin{pmatrix} \frac{5}{8} & 0\\ 0 & 1 \end{pmatrix}$$

$$\mathbf{P} = \begin{pmatrix} 0 & \frac{1}{8} \\ 0 & 0 \end{pmatrix}$$

Chiaramente,

$$\mathbf{N}^{-1} = \begin{pmatrix} \frac{8}{5} & 0\\ 0 & 1 \end{pmatrix}$$

e quindi

$$\mathbf{N}^{-1}\mathbf{P} = \begin{pmatrix} \frac{8}{5} & 0\\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & \frac{1}{8}\\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{5}\\ 0 & 0 \end{pmatrix}$$

Troviamo le prime iterazioni del *metodo di Jacobi*. Partiamo, come al solito, con l'approssimazione iniziale  $\mathbf{b}^{(0)} = (0,0)$ . Allora

$$\mathbf{b}^{(1)} = \mathbf{N}^{-1}\mathbf{e} = (0,1)$$

(la seconda colonna di  $N^{-1}$ ). Osserviamo che, per ogni t, si ha  $N^{-1}P(t,1) = (1/5,0)$ , e quindi

$$\mathbf{b}^{(2)} = \mathbf{N}^{-1}\mathbf{P} \ \mathbf{b}^{(1)} + \mathbf{N}^{-1}\mathbf{e} = (1/5,0) + (0,1) = (1/5,1), \\ \mathbf{b}^{(3)} = \mathbf{N}^{-1}\mathbf{P} \ \mathbf{b}^{(2)} + \mathbf{N}^{-1}\mathbf{e} = (1/5,0) + (0,1) = (1/5,1).$$

Pertanto, dopo il terzo passo, le iterazioni si stabilizzano: la radiosità della ciotola K è 1/5 della potenza emessa dalla lampada, il resto della luce si perde nello sfondo (ma naturalmente lo sfondo continua a creare luce con densità di potenza pari a 1).

*Parte 2.* Ora rieseguiamo il calcolo delle prime tre iterazioni con il *metodo di Gauss-Seidel*. Nel calcolo di  $\mathbf{b}^{(1)} = N^{-1}\mathbf{e}$ , la prima componente del risultato è 0, ossia non cambia rispetto a  $\mathbf{b}^{(0)}$ , e pertanto anche nel calcolo cella seconda componente la prima iterazione di Gauss-Seidel coincide con quella di Jacobi:

$$\mathbf{b}^{(1)} = (0,1).$$

Nella seconda iterazione, applichiamo la prima riga della matrice N<sup>-1</sup>P a  $\mathbf{b}^{(1)}$ , ottenendo così la prima componente  $b_{21}$  di  $\mathbf{b}^{(1)}$ , che, come prima, vale 1/5. Invece la seconda riga, per il calcolo della seconda componente  $b_{22}$  di  $\mathbf{b}^{(2)}$ , viene applicata al vettore che ha come prima componente quella appena ottenuta. Ma la seconda riga di N<sup>-1</sup>P è nulla, e quindi non cambia niente: i due metodi coincidono alla seconda iterazione. Per lo stesso motivo, si vede che al passo successivi metodi continuano a coincidere, ed a quel punto le iterazioni si sono stabilizzate.

Parte 3. Ora rieseguiamo il calcolo col metodo di Southwell. Alla n-sima iterazione, dobbiamo trovare quale è la componente più grande del vettore resto  $\mathbf{r}^{(n)} = \mathbf{e} - \mathbf{M} \mathbf{b}^{(n)}$  e rilassare solo quella. Per n=0 ovviamente si ha  $\mathbf{r} = \mathbf{e}$ , e quindi rilassiamo la seconda riga (la prima componente di **e** è 0). Il risultato è  $\mathbf{b}^{(1)} = (0,1)$ (ossia la prima componente non cambia): questo era anche il risultato degli altri metodi, i quali però richiedevano di risolvere entrambe le righe del sistema lineare e non una sola (è vero che l'altra è banale, ma il software di esecuzione questo non può saperlo e quindi non può tenerne conto). Ora abbiamo  $\mathbf{r}^{(1)} = \mathbf{e}$  -M  $\mathbf{b}^{(1)} = (0,1) - (-1/8,1) = (1/8,0)$ . La componente più grande è la prima, e quindi rilassiamo la prima riga. Come già visto, si ottiene come prima componente di  $\mathbf{b}^{(2)} = \mathbf{N}^{-1}\mathbf{P} \mathbf{b}^{(1)} + \mathbf{N}^{-1}\mathbf{e}$  il valore 1/5. La seconda componente non cambia, resta quindi 1, ed il risultato è  $\mathbf{b}^{(2)} = (1/5,1)$ : lo stesso risultato degli altri due metodi. Allo stesso modo, si trova  $\mathbf{b}^{(3)} = (1/5,1)$  ed a questo punto l'iterazione si è stabilizzata: ma abbiamo eseguito solo la metà delle operazioni aritmetiche (in effetti, abbiamo evitato di risolvere quelle equazioni lineari banali i cui coefficienti sono tutti nulli).

**Esercizio 10.** Una scena si compone del volume delimitato esternamente dalle seguenti due superficie: l'emisfero di raggio 1 e centro l'origine nel semispazio  $z \ge 0$ , ed un cono diretto verso il basso (ossia tutto al di sotto del livello del suo vertice) nel semispazio  $z \ge 0$ , con asse centrale l'asse z e tangente all'emisfero. L'intersezione di questo cono con il piano z = 0 è un disco il cui raggio è pari all'altezza del cono. Le due superficie (sfera e cono) delimitano un volume V che forma la nostra scena (un altro volume è delimitato da sfera, cono ed il piano di base z = 0, ma in questo problema questo secondo volume non ci interessa). D'ora in poi chiamiamo C la porzione del cono dal cerchio di tangenza al vertice del cono, e S la calotta sferica consistente della parte dell'emisfero compresa fra il parallelo di tangenza ed il polo nord. Suddividiamo la scena nei due elementi C (primo elemento, che indicheremo come soffitto) e S (secondo elemento, pavimento).

Si osservi che il cono è generato, per rotazione intorno all'asse z, da una semiretta con pendenza 1 (ossia angolo  $\frac{\pi}{4}$  rispetto alla verticale).

- (i) A quale altezza  $z_0$  il cono è tangente alla sfera? A quale angolo di latitudine sull'emisfero è situato il parallelo su cui le due superficie sono tangenti?
- (ii) Calcolare l'area di C.
- (iii) Calcolare l'area di S.
- (*iv*) Calcolare i fattori di forma  $F_{SS}$ ,  $F_{CC}$ ,  $F_{SC}$ ,  $F_{CS}$ , usando la reversibilità dei percorsi ottici e la conservazione dell'angolo solido. Scrivere la matrice M della radiosità assumendo che i fattori di riflettività siano  $\rho_S = 1/2, \ \rho_C = 1/4.$
- (v) Calcolare  $M^{-1}$ . Assumendo che una lampada di potenza 1 stia al vertice del cono, calcolare le radiosità.
- (vi) Si consideri il fattore di forma differenziale dal vertice del cono alla calotta S, già calcolato in un esercizio precedente. Si noti che il pavimento della scena (la calotta sferica S) è convesso. Rimpiazziamolo con un pavimento piatto (il disco di base B del cono, considerato nella domanda (iii)), o con uno concavo (ad esempio un'altra calotta sferica S', della stessa area di S e della stessa curvatura, ma concava, ossia che si protende verso l'esterno del volume del cono invece che verso

l'interno: si tratta della riflessione speculare di S rispetto al disco di base B). Come cambia il fattore di forma differenziale?

- (vii) Supponiamo di rimpiazzare S con un nuovo pavimento S' dato dalla calotta sferica riflessa speculare concava introdotta nella domanda (vi). In tal modo si ottengono altri fattori di forma, ad esempio  $F_{CS'}$  invece di  $F_{CS}$ . Qual è più grande fra  $F_{CS'}$  e  $F_{CS}$ ?
- (viii) Ripetere il calcolo dei fattori di forma e della radiosità per il solido delimitato dal cono C e dalla superficie sferica concava S' considerata nella parte precedente.

Svolgimento. Parte (i). L'angolo formato dalla superficie del cono con l'asse  $z \ e \ \frac{\pi}{4}$ . Quindi il parallelo a cui il cono è tangente alla sfera deve avere la stessa pendenza. La inclinazione della sfera rispetto alla verticale  $e \ \frac{\pi}{4}$  al parallelo di latitudine  $\frac{\pi}{4}$ . Questo parallelo è la circonferenza  $z = \cos \frac{\pi}{4} = \frac{1}{\sqrt{2}}$ ,  $x^2 + y^2 = \sin \frac{\pi}{4} = \frac{1}{\sqrt{2}}$ .

Parti (ii) e (iii). Abbiamo calcolato l'area di coni e calotte sferiche nella Sottosezione Fattore locale di ingrandimento delle aree nella parametrizzazione di una superficie, e calcolo dell'area di una superficie parametrica della Sezione Mappa di rilievo del Capitolo Rendering con mappe parametriche).

Rammentiamo il calcolo dell'area della calotta sferica: si usa la formula del fattore di ingrandimento delle aree per una superficie parametrica del tipo p(x,y) = (x, y, f(x, y)), dove f è una funzione di due variabili. Nel nostro caso, sia B il disco di base del cono C, ovvero  $B = \left\{z = \frac{1}{\sqrt{2}}, x^2 + y^2 \leqslant \frac{1}{\sqrt{2}}\right\}$ , e K la sua proiezione sul piano z = 0, ossia  $K = \left\{x^2 + y^2 \leqslant \frac{1}{\sqrt{2}}\right\}$ . Allora f è la funzione  $\sqrt{1 - x^2 - y^2}$ , il cui grafico sul dominio K è appunto la calotta sferica S. Il fattore di ingrandimento delle aree in tal caso risulta essere

$$\sqrt{1 + \left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial x}\right)^2} = \frac{1}{\sqrt{1 - x^2 - y^2}}$$

Quindi l'area è data dal seguente integrale

Area(S) = 
$$\int_{K} \frac{1}{\sqrt{1 - x^2 - y^2}} dx \, dy = \int_{0}^{2\pi} \int_{0}^{\frac{1}{\sqrt{2}}} \frac{1}{\sqrt{1 - r^2}} r \, dr \, d\theta$$
  
=  $\pi \int_{0}^{\frac{1}{2}} \frac{1}{\sqrt{1 - u}} du = 2\pi \sqrt{1 - v} \Big|_{\frac{1}{2}}^{0} = 2\pi (1 - \frac{1}{\sqrt{2}}) = \pi (2 - \sqrt{2}) .$ 

Si rammenti anche che, in base alla stessa formula per il fattore di ingrandimento, l'area del cono è  $\pi AR$ , dove R è il raggio della sua circonferenza di base e A è la distanza da un generico punto della circonferenza di base al vertice del cono. Il cono C ha altezza h uguale al raggio di base, che è  $\frac{1}{\sqrt{2}}$ , quindi  $R = \frac{1}{\sqrt{2}}$  e, per il teorema di Pitagora,  $A = \sqrt{R^2 + h^2} = 1$ . Pertanto Area $(C) = \frac{\pi}{\sqrt{2}}$ .

Parte (iv). Poiché la calotta sferica S è convessa, il fattore di forma  $F_{SS}$  è nullo, e per la conservazione dell'angolo solido totale abbiamo  $F_{SC} =$  1. Allora, per la relazione di reciprocità ed il calcolo delle aree nelle parti precedenti, si ha

$$F_{CS} = \frac{\operatorname{Area}(S)}{\operatorname{Area}(C)} F_{SC} = 2(\sqrt{2} - 1) .$$

Invece il cono C è concavo e quindi  $F_{CC} > 0$ ; ancora per la conservazione dell'angolo solido totale si ha  $F_{CC} + F_{CS} = 1$ , da cui

$$F_{CC} = 1 - F_{CS} = 3 - 2\sqrt{2}$$
.

Pertanto, con i valori scelti  $\rho_C = 1/4$  <br/>e $\rho_S = 1/2$ , la matrice M della radiosità risulta essere

$$M = \begin{pmatrix} 1 - \rho_C F_{CC} & -\rho_C F_{CS} \\ -\rho_S F_{SC} & 1 - \rho_S F_{SS} \end{pmatrix} = \begin{pmatrix} 1 - \frac{1}{4} F_{CC} & -\frac{1}{4} F_{CS} \\ -\frac{1}{2} F_{SC} & 1 \end{pmatrix}$$
$$= \begin{pmatrix} \frac{3}{4} + \frac{1}{4} F_{CS} & -\frac{1}{4} F_{CS} \\ -\frac{1}{2} & 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{4} + \frac{1}{\sqrt{2}} & \frac{1}{2} - \frac{1}{\sqrt{2}} \\ -\frac{1}{2} & 1 \end{pmatrix}$$
$$= \begin{pmatrix} \frac{\sqrt{2}+4}{4\sqrt{2}} & \frac{\sqrt{2}-2}{2\sqrt{2}} \\ -\frac{1}{2} & 1 \end{pmatrix}.$$

Parte (v). È ben noto ed immediatamente verificato che l'inversa di una matrice

$$A = \left(\begin{array}{cc} a & b \\ c & d \end{array}\right)$$

è

$$A^{-1} = \frac{1}{\det A} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

dove det A = ad - bc. Nel caso presente,

det 
$$M = 1 - \rho_C F_{CC} - \rho_C \rho_S F_{CS} = 1 - \rho_C (1 - F_{CS}) - \rho_C \rho_S F_{CS}$$
  
=  $1 - \rho_C + \rho_C (1 - \rho_S) F_{CS}$ 

ossia, con i valori $\rho_C=1/4$ e $\rho_S=1/2,$ 

det 
$$M = \frac{3}{4} + \frac{1}{8}F_{CS} = \frac{1}{2} + \frac{1}{4}\sqrt{2} = \frac{1}{2}(1 + \frac{1}{\sqrt{2}}) = \frac{1 + \sqrt{2}}{2\sqrt{2}}.$$

Quindi

$$M^{-1} = \frac{1}{1 - \rho_C + \rho_C (1 - \rho_S) F_{CS}} \begin{pmatrix} 1 - \rho_S F_{SS} & \rho_C F_{CS} \\ \rho_S F_{SC} & 1 - \rho_C F_{CC} \end{pmatrix}$$
$$= \frac{1}{1 - \rho_C + \rho_C (1 - \rho_S) F_{CS}} \begin{pmatrix} 1 & \rho_C F_{CS} \\ \rho_S & 1 - \rho_C (1 - F_{CS}) \end{pmatrix}$$
$$= \frac{2\sqrt{2}}{1 + \sqrt{2}} \begin{pmatrix} 1 & \frac{2 - \sqrt{2}}{2\sqrt{2}} \\ \frac{1}{2} & \frac{\sqrt{2} + 4}{4\sqrt{2}} \end{pmatrix} = \begin{pmatrix} \frac{2\sqrt{2}}{1 + \sqrt{2}} & \frac{2 - \sqrt{2}}{1 + \sqrt{2}} \\ \frac{\sqrt{2}}{1 + \sqrt{2}} & \frac{4 + \sqrt{2}}{2 + 2\sqrt{2}} \end{pmatrix}.$$

La sorgente di luce a potenza 1 sta nel primo elemento, quindi il vettore della potenza creata è il primo vettore della base canonica

$$\mathbf{e} = \left(\begin{array}{c} 1\\ 0 \end{array}\right)$$

e pertanto il vettore delle radiosità è la prima colonna di ${\cal M}^{-1}$  :

$$\mathbf{b} = M^{-1}\mathbf{e} = \begin{pmatrix} \frac{2\sqrt{2}}{1+\sqrt{2}} \\ \frac{\sqrt{2}}{1+\sqrt{2}} \end{pmatrix} \approx \begin{pmatrix} 1.1716 \\ 0.5858 \end{pmatrix}.$$

Parte (vi). Come già osservato precedentemente negli esercizi della Sezione Esercizi sula calcolo analitico dei fattori di forma più sopra in questo Capitolo, il fattore di forma differenziale dal vertice del cono non cambia affatto, perché qualunque sia la forma del pavimento esso copre sempre, visto dal vertice, lo stesso angolo solido, che è l'angolo solido di apertura del cono. Questo è conseguenza del fatto che ogni semiretta interna al cono uscente dal vertice incontra un punto del pavimento ma non incontra mai un altro punto del cono: non ci sono quindi fattori di ostruzione. Si osservi però che la stessa cosa non è vera per le semirette uscenti da altri punti del cono: esse possono incontrare punti del cono invece che del pavimento. Se il pavimento è piatto o concavo (ossia incurvato verso l'esterno del volume) allora il volume risultante è convesso, e quindi le semirette uscenti da un vertice del cono ed indirizzate ad altri punti del cono non possono essere ostruite da punti del pavimento: ne risulta che, in questo caso, il fattore di forma differenziale daun punto fissato del cono a sé stesso rimane costante al variare della forma del pavimento, purché essa rimanga concava (naturalmente, questo fattore di forma differenziale ad un punto del cono può cambiare con la scelta del punto. Invece se, come nella situazione originale, il pavimento è convesso, allora alcune semirette da un punto del cono diverso dal vertice al cono stesso possono essere intercettate dal pavimento, e la percentuale di tali semiretta cambia al cambiare della superficie (ad esempio, se il pavimento è una superficie sferica rientrante dentro il cono, allora sempre più semirette uscenti da punti vicini alla circonferenza di base del cono vengono intercettate da scelte via via più curve del pavimento. Pertanto, in questo caso, per punti sufficientemente vicini alla base del cono il fattore di forma differenziale decresce al crescere della curvatura.

Parte (vii). I fattori di forma globali  $F_{CS}$  e  $F_{CS'}$  sono definiti come le medie (di area) dei corrispondenti fattori di forma differenziali al variare del punto origine nell'area del cono. Da quanto visto alla parte precedente, i fattori di forma differenziali se il pavimento è concavo S' sono maggiori (se il punto origine è abbastanza vicino alla base del cono) dei fattori di forma differenziali nel caso del pavimento convesso S. Questo è vero anche se i pavimenti S e S' non sono sferici e riflessi speculari l'uno dell'altro. ma nel caso in esame, in cui essi, essendo riflessi speculari, hanno la stessa area, allora è facile raggiungere la stessa conclusione anche senza ricorrere ai fattori di forma differenziali. Infatti, adesso si ha  $F_{SS} = 0$  per la convessità di S, ma  $F_{S'S'} > 0$  per la concavità di S'. Dalla conservazione dell'angolo solido totale ora segue  $F_{SC} = 1$  ma  $F_{S'C} < 1$ . Dalla reciprocità ora abbiamo

$$F_{CS} = \frac{\operatorname{Area}(S)}{\operatorname{Area}(C)} F_{SC} = \frac{\operatorname{Area}(S)}{\operatorname{Area}(C)}$$

ma

$$F_{CS'} = \frac{\operatorname{Area}(S')}{\operatorname{Area}(C)} F_{S'C} < \frac{\operatorname{Area}(S')}{\operatorname{Area}(C)} = \frac{\operatorname{Area}(S')}{\operatorname{Area}(C)} = F_{CS}.$$

Quindi  $F_{CS'} < F_{CS}$ .

Parte (viii). Questi calcoli vengono lasciati per esercizio al lettore.

### Esercizio 11.

- (i) Una scena consiste di una stanza cilindrica avente per base un disco intorno all'origine di raggio 1 ed altezza 2; il pavimento ed il soffitto hanno riflettività  $\frac{1}{2}$  e la parete riflettività  $\frac{1}{4}$ . Sia  $\alpha$  il fattore di forma dal soffitto al pavimento. Si possono determinare tutti gli altri fattori di forma in termini di  $\alpha$ ? Se no, spiegare perchè; se sì, scrivere la matrice della radiosità e cercare di specificare quale sia il range dei valori ammissibili per  $\alpha$ .
- (ii) Nella stessa stanza il pavimento viene sostituito da un cono retto che estrude dalla stanza (la quale quindi è convessa), di altezza pari al raggio. Rispondere alle domande precedenti. (Primo suggerimento: se non si ricorda l'area del cono, la si può calcolare grazie alla formula del fattore di ingrandimento delle aree per una superficie parametrica del tipo p(x, y) = (x, y, f(x, y)), dove f è una funzione di due variabili. Il fattore di ingrandimento delle aree in tal caso risulta essere

$$\sqrt{1 + \left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Questo permette di calcolare l'area come integrale doppio. Secondo suggerimento: visto che il cono estrude, la stanza rimane convessa. Quindi il pavimento non crea occlusioni nella visibilità relativa di ciascunao degli altri elementi, e pertanto gli angoli solidi sottesi ed i fattori di forma non cambiano. Neppure il fattore di forma da soffitto al pavimento conico e dalla parete laterale al pavimento conico cambiano: rimangono entrambi uguali a quelli che erano quando il pavimento era piatto. Le uniche cose che cambiano rospetto a quando il pavimento era piatto sono:

- cambia il fattore di forma dal pavimento alla parete laterale ed al soffitto (perché cambiano i fattori di forma differenziali dai punti del pavimento conico alle altre due pareti, in quanto ora varia la distanza da ciascuno di questi punti alle pareti: in effetti a caus dell'aumento di questa distanza si vede che dal pavimento che estrude a parete laterale e soffitto i fattori di forma sono più piccoli di quando il pavimento era piatto);
- diventa non nullo il fattore di forma dal pavimento a sé stesso: quest'ultimo fatto ci dà un fattore di forma in più. Ma d'altra parte i fattori di forma da pavimento a parete e da pavimento a soffitto si calcolano per reciprocità una volta calcolata l'area del pavimento: quindi ciò che viene a diminuire in questi fattori di forma va a finire nel nuovo fattore di forma da pavimento a pavimento

Segue da quanto appena detto che tutti i fattori di forma si calcolano a partire da uno di essi, diciamo ad esempio il fattore di forma a parete a pavimento, o da soffitto a pavimento, che rimangono esattamente come erano quando il pavimento era piatto.)

(iii) Ora nella stessa stanza il pavimento viene sostituito da un cono retto che intrude nella stanza (la quale quindi non è convessa), di altezza pari al raggio. Rispondere alle domande precedenti. (Suggerimento: ora la stanza non è convessa. Il pavimento occlude la visibilità dalla parete laterale a sé stessa, e quindi cambia il fattore di forma dalla parete a sé stessa; però il pavimento non occlude la visibilità da parete a soffitto, ed il fattore di forma da soffitto a pavimento resta lo stesso delle domande precedenti, come quando il pavimento era piatto, perché anche in questo caso la visibilità non viene occlusa.)

*Esercizio 12.* Nella stessa stanza cilindrica di raggio m, ora il pavimento P è parabolico, di equazione  $z = f(x, y) = x^2 + y^2 - m^2$ .

- (i) Calcolare l'area di P.
- (*ii*) Quanto vale il fattore di forma differenziale dal centro del soffitto al pavimento?
- (*iii*) Quanto vale il fattore di forma differenziale dal centro del pavimento al soffitto?

(Suggerimento: la stanza è convessa, e quindi ogni raggio dal centro del soffitto (o anche ogni altro punto del soffitto) al pavimento interseca il disco di base del cilindro. Quindi il fattore di forma differenziale è lo stesso che se il pavimento fosse piatto.)

# METODI NUMERICI ALTERNATIVI (MA LENTI) PER FACILITARE IL CALCOLO DELLE PRIME ITERAZIONI

Riprendiamo in esame il modo di trasformare il problema della radiosità,  $M\mathbf{b} = \mathbf{e}$ , in un metodo iterativo: si decompone M = A - B ed il problema diventa  $A\mathbf{b} = B\mathbf{b} + \mathbf{e}$ , da cui la procedura iterativa

$$\mathbf{b}^{(\mathbf{k}+1)} = A^{-1}B\mathbf{b}^{\mathbf{k}} + A^{-1}\mathbf{e}.$$

La parte più onerosa del calcolo della prima iterazione consiste nel trovare la matrice inversa  $A^{-1}$ : osserviamo però che il calcolo dell'inversa è necessario solo una volta, alla prima iterazione: basta memorizzare l'inversa ed alle iterazioni successive la si può usare senza rideterminarla.

# 1 Procedura semplificata per il metodo di Jacobi

Nel metodo di Jacobi abbiamo decomposto M = N - P, con P strettamente triangolare superiore (zeri sulla diagonale) e N triangolare inferiore ma non nulla sulla diagonale (come osservato nella sezione "L'equazione della radiosità"). In altre parole, abbiamo posto A = N e B = P. Per tradizione, nei metodi numerici la matrice triangolare superiore si indica con U (da up-per triangular) invece che con P: lo faremo spesso anche qui di seguito.

Ora invece effettuiamo una decomposizione più fine, spezzando N come N = D - L, dove L è la parte strettamente triangolare inferiore (con segno cambiato) e D è la parte diagonale (con elementi diagonali non nulli, come appena osservato: in particolare, D è invertibile). Da questa decomposizione otteniamo  $D\mathbf{b} = (L + U)\mathbf{b} + \mathbf{e}$ , ossia

$$\mathbf{b}^{(\mathbf{k}+1)} = D^{-1}(L+U)\mathbf{b}^{\mathbf{k}} + D^{-1}\mathbf{e}.$$

Ora il calcolo della matrice inversa  $D^{-1}$  è banale:  $D^{-1}$  è la matrice diagonale che sulla diagonale ha i reciproci degli elementi corrispondenti di D. Questo velocizza la prima iterazione, ma vedremo che a lungo andare il procedimento diventa più lentoperché è più lenta la velocità di convergenza degli approssimanti. Quindi questo approccio è sconsigliabile nella pratica numerica, ma è comodo per risolvere problemi in cui sono richieste solo le prime iterazioni. Però ci sono due osservazioni importanti.

La prima è che occorre dimostrare che questo metodo di rilassamento converge ad un punto fisso **b**. La condizione necessaria e sufficiente, data dal Teorema 1, è che il raggio spettrale della matrice  $D^{-1}(L+U)$  sia minore di 1. Questa condizione sembrerebbe difficile da verificare. Certamente, se la matrice diagonale  $D^{-1}$  ha autovalori grandi, allora è grande il suo raggio spettrale, che coincide con il massimo modulo dei suoi autovalori. I coefficienti diagonali di  $D^{-1}$  sono i reciproci dei coefficienti diagonali di D: saremmo quindi tentati a limitare l'attenzione al caso (frequente nella pratica) in cui i coefficienti diagonali di D, ossia i numeri  $M_{ii} = 1 - \rho_i F_{ii}$ , sono uguali a 1. Spesso gli elementi della scena sono piani o convessi e quindi, in tali casi,  $F_{ii} = 0$ : quindi i coefficienti diagonali di D sono sempre 1 se la riflettività di ogni elemento della scena che non sia piano o convesso è 0. Sotto questa ipotesi, quindi, il raggio spettrale di  $D^{-1}$  vale 1: ma a noi serve che il raggio spettrale di  $D^{-1}(L+U)$  sia minore di 1 (condizione equivalente alla convergenza del metodo iterativo), e questo non è ovvio ed occorre dimostrarlo. Su questo punto ci viene in aiuto un risultato di analisi numerica, il teorema di Stein e Rosenberg (Teorema 8.2.14 in J. Stoer, R. Bulirsch, Introduzione all'analisi numerica, vol. 2, Zanichelli, Bologna, 1979), che vale in generale, anche senza l'ipotesi di comodo che i coefficienti diagonali di D siano tutti 1. In base a questo teorema, se il raggio spettrale  $\rho_{L^{-1}U}$  di  $N^{-1}P = L^{-1}U$ è minore di 1, allora lo è anche quello di  $D^{-1}(L+U)$ , che indichiamo con  $\rho_{D^{-1}(L+U)}$ , e questo è quanto quanto volevamo; inoltre, se  $\rho_{L^{-1}U} < 1$ , questo teorema dice di più: vale anche la disuguaglianza  $0 < \rho_{L^{-1}U} < \rho_{D^{-1}(L+U)} < 1$ . Si badi al fatto che le matrici  $L \in U$  in questo riferimento bibliografico sono le matrici  $D^{-1}L \in D^{-1}U$  nella nostra notazione). Che il raggio spettrale di  $N^{-1}P$  sia minore di 1 è la condizione equivalente alla convergenza del metodo usuale di Jacobi, e questa convergenza è verificata perché sappiamo che la matrice M = N - P è a predominanza diagonale stretta (si veda il Lemma nella Sezione Rilassamento di Gauss-Seidel applicato al sistema lineare della radiosità, più sopra in questo Capitolo.

Per inciso, osserviamo che la matrice L + U ha elementi diagonali tutti nulli e quindi certamente non è a predominanza diagonale stretta; la matrice  $D^{-1}(L + U)$  si ottiene dalla precedente moltiplicando ciascuna riga per il corrispondente numero sulla diagonale di  $D^{-1}$ , e quindi neanche essa è a predominanza diagonale stretta. Pertanto non si applica la condizione sufficiente di convergenza per il metodo di rilassamento di Gauss–Seidel, ma per fortuna il teorema di Stein e Rosenberg assicura la convergenza del metodo.

La seconda osservazione è che le iterazioni così ottenute *non* sono le stesse del metodo di Jacobi introdotto precedentemente. Però, visto che il metodo converge, ossia il raggio spettrale  $\rho_{L^{-1}U}$  è minore di 1, allora il ragionamento che porta all'identità (6") prova che la differenza fra le iterate k-sime del metodo di Jacobi ed il punto fisso **b** tende a zero come  $\rho_{L^{-1}U}^{k}$ . Lo stesso ragionamento prova che la differenza fra le iterate k-sime del metodo semplificato ed il punto fisso **b** tende a zero come  $\rho_{D^{-1}(L+U)}^{k}$ , che, come visto sopra, è più grande. Quindi il metodo se3mplificato converge allo stesso punto fisso, ma più lentamente. Negli esempi comunque noi non utilizziamo mai in questo libro il metodo di iterazione semplificato.

# 2 Cosa succede se si applica la stessa idea al metodo di Gauss–Seidel

Reinterpretiamo il metodo di Gauss–Seidel come segue. Scriviamo, come abbiamo fatto all'inizio, M = N - P = D - L - U, ossia A = D - L e B = U. Pertanto,

$$(D-L)\mathbf{b}^{(\mathbf{k}+1)} = U\mathbf{b}^{\mathbf{k}} + \mathbf{e}\,,$$

da cui

$$D\mathbf{b}^{(\mathbf{k}+1)} = L\mathbf{b}^{(\mathbf{k}+1)} + U\mathbf{b}^{\mathbf{k}} + \mathbf{e}.$$

Riconosciamo a destra l'aspetto tipico del metodo iterativo di Gauss–Seidel, dove per ogni riga *i* si impiegano, per i termini della somma con j < i, i valori delle componenti del vettore approssimante  $\mathbf{b}^{(\mathbf{k}+1)}$  invece che quelli di  $\mathbf{b}^{(\mathbf{k})}$ . Pertanto l'iterazione diventa

$$\mathbf{b}^{(\mathbf{k}+\mathbf{1})} = D^{-1} \left( L \mathbf{b}^{(\mathbf{k}+\mathbf{1})} + U \mathbf{b}^{\mathbf{k}} + \mathbf{e} \right).$$

Si riottiene così, parola per parola, esattamente il metodo di Gauss-Seidel come lo abbiamo già presentato nelle equazioni (7): quindi questa nuova decomposizione non porta alcun vantaggio rispetto a prima (infatti, è proprio la stessa di prima). Il prodotto a sinistra per la matrice  $D^{-1}$  altro non fa che dividere la componente *i*-sima per il termine diagonale *i*-simo della matrice della radiosità,  $M_{ii}$ .

# VOLUME 3

# **ILLUMINAZIONE GLOBALE**

# OTTAVA PARTE ILLUMINAZIONE GLOBALE

## CAPITOLO 1

# Il trasporto della luce: flusso, radianza, riflettività bidirezionale ed equazione del rendering

L'obiettivo degli algoritmi di rendering è quello di creare immagini fotorealistiche. Per ogni pixel dell'immagine, questi algoritmi devono trovare l'oggetto della visibile in quel determinato pixel, e poi visualizzarlo con il colore appropriato. Si tratta quindi di algoritmi a precisione di immagi.

### 1. Quantità radiometriche

1.1. Flusso. La potenza radiante, spesso rappresenta con  $\Phi$ , viene espressa in Watt (W=Joule/sec) e si chiama *flusso*. Il flusso esprime l'energia totale che passa da, a, o attraverso una superficie per unità di tempo. Il flusso non dipende dalla dimensione della sorgente di luce, o da quella dell'oggetto che la riceve, e neanche dalla distanza tra la sorgente e l'oggetto. Questa è considerata la grandezza radiometrica fondamentale, sulla base della quale sono definite tutte le altre.

**1.2. Irradianza.** L'irradianza, solitamente indicata con E, è il flusso radiante incidente su una superficie per unità di area, e viene espressa in Watt/m<sup>2</sup>:

$$E = \frac{d\Phi}{d\sigma}$$

**1.3. Emissione radiante o Radiosità.** L'emissione radiante (M), detta anche *radiosità* (B), è il flusso radiante emesso da una sorgente estesa per unità di area, ed è espressa in W/m<sup>2</sup>:

$$M = B = \frac{d\Phi}{d\sigma}$$

**1.4. Radianza.** La radianza è il flusso radiante emesso da una sorgente estesa per unità di angolo solido e per unità di area proiettata su un piano normale alla direzione considerata: quindi misura la

distribuzione angolare della potenza uscente per unità di area. La radianza è una quantità a cinque dimensioni che varia con la posizione e la direzione ed è espressa come  $L(x, \vec{\Theta})$ 

$$L = \frac{d^2 \Phi}{d\omega \ d\sigma^{\perp}} = \frac{d^2 \Phi}{d\omega \ d\sigma \ \cos \theta}$$



FIGURA 1. Definizione di radianza: il flusso radiante emesso da una sorgente estesa per unità di angolo solido

Probabilmente la radianza è la quantità più importante negli algoritmi di illuminazione globale, dato che essa esprime l'intensità con cui appaiono illuminati gli oggetti nella scena.

### 2. Relazione tra le quantità radiometriche

In base alla definizione della quantità radiometrica data prima, abbiamo le seguenti relazioni:

$$\Phi = \int_{\sigma} \int_{\Omega} L(x \to \vec{\Theta}) \cos \theta \, d\omega(\vec{\Theta}) \, d\sigma(x)$$
$$E(x) = \int_{\Omega} L(x \leftarrow \Theta) \, \cos \theta \, d\omega(\vec{\Theta})$$
$$B(x) = \int_{\Omega} L(x \to \vec{\Theta}) \, \cos \theta \, d\omega(\vec{\Theta}) \, .$$

La notazione:  $L(x \to \vec{\Theta})$  indica la radianza che parte dal punto x e va in direzione  $\vec{\Theta}$ , mentre  $L(x \leftarrow \Theta)$  rappresenta la radianza in arrivo al punto x proveniente dalla direzione  $\vec{\Theta}$ . 2.1. Dipendenza dalla lunghezza d'onda. Le misure radiometriche e le quantità descritte precedentemente non dipendono sono solo dalla posizione e dalla direzione, ma anche dipendenti dalla lunghezza d'onda e dall'energia della luce. Se si specifica esplicitamente la dipendenza dalla lunghezza d'onda, ad esempio della radianza, la corrispondente quantità radiometrica si chiama radianza spettrale. La radianza viene allora calcolata integrando la radianza spettrale sull'intervallo di lunghezze d'onda della luce visibile. Per esempio,

$$L(x \to \vec{\Theta}) = \int_{\text{spettro}} L(x \to \vec{\Theta}, \lambda) \, d\lambda$$

La dipendenza della lunghezza d'onda dai valori radiometrici è spesso iincorporata implicitamente nell'equazione di illuminazione globale ma non viene menzionata esplicitamente, anche se poi occorre farlo quando si svolgono i calcoli o quando si sviluppa il codice di un applicativo di rendering.

#### 3. Proprietà della radianza

La radianza è una quantità radiometrica fondamentale ai fini del rendering. Le altre quantità radiometriche, il flusso, l'irradianza e la radiosità, possono essere derivate dalla radianza. Enunciamo e dimostriamo alcune proprietà della radianza.

**3.1. Reversibilità.** La radianza non varia lungo percorsi rettilinei, purché la luce viaggi nel vuoto, cioè senza diffusione o assorbimento da parte di alcun mezzo. La proprietà dell'invarianza della radianza equivale alla seguente espressione:

$$L(x \to y) = L(x \leftarrow y),$$

che afferma che la radianza che lascia il punto x diretta verso il punto y è uguale alla radianza che arriva al punto y dal punto x. Illustriamo questo in Figura 2. Dalla definizione di radianza, la potenza totale che parte l'area  $d\sigma(x)$  ed arriva all'area  $d\sigma(y)$  può essere scritta come segue:

$$L(x \to y) = \frac{d^2 \Phi}{(d\sigma(x) \cos \theta_x) d\omega(x \leftarrow d\sigma(y))},$$
$$d^2 \Phi = L(x \to y) \cos \theta_x d\omega(x \leftarrow d\sigma(y)) d\sigma(x),$$



FIGURA 2. Invarianza della radianza lungo i raggi

dove  $d\vec{\omega}(x \leftarrow d\sigma(y))$  è l'angolo solido sotteso da  $d\sigma(y)$ , visto da x. La potenza che arriva all'area  $d\sigma(y)$  dall'area  $d\sigma(x)$ , può essere espressa in maniera simile:

$$L(y \leftarrow x) = \frac{d^2 \Phi}{(\cos \theta_y \, d\sigma(y)) d\omega(y \leftarrow d\sigma(x))}$$

$$d^{2}\Phi = L(y \leftarrow x) \cos \theta_{y}) \, d\omega(y \leftarrow d\sigma(x)) \, d\sigma(y) \, .$$

I differenziali degli angoli solidi sono:

$$d\omega_{x \leftarrow \sigma(y)} = \frac{\cos \theta_y \, d\sigma(y)}{r_{xy}^2} \,,$$
$$d\omega_{y \leftarrow \sigma(x)} = \frac{\cos \theta_x \, d\sigma(x)}{r_{xy}^2} \,.$$

Assumiamo che non ci siano mezzi attraverso i quali si trasmette la luce in arrivo dall'area  $d\sigma(y)$ , ossia che le due superficie siano nel vuoto: quindi non c'è perdita di energia dovuta alla diffusione ambientale. 4. ESEMPI

Allora, per la legge della conservazione dell'energia, tutta l'energia uscente da  $d\sigma(x)$  nella direzione della superficie  $d\sigma(y)$  deve arrivare a destinazione:

$$L(x \to y) \cos \theta_x \, d\omega(x \leftarrow d\sigma(y)) \, d\sigma(x)$$
  
=  $L(y \leftarrow x) \cos \theta_y \, d\omega(y \leftarrow d\sigma(x)) \, d\sigma(y)$ ,

ovvero

$$L(x \to y) \cos \theta_x \frac{\cos \theta_y \, d\sigma(y)}{r_{xy}^2} = L(y \leftarrow x) \, \cos \theta_y \, \frac{\cos \theta_x \, d\sigma(x)}{r_{xy}^2} \, .$$

Quindi si ottiene

$$L(x \to y) \cos \theta_x = L(y \leftarrow x) \cos \theta_y$$

Ripetiamo: la radianza non varia lungo un tragitto rettilineo, e non si attenua con la distanza, grazie all'ipotesi che non ci siano mezzi di propagazione che possono assorbire o deviare l'energia tra le due superficie.

Da questa osservazione segue che, se la radianza incidente o uscente è nota per tutti i punti della superficie, allora la distribuzione della radianza è conosciuta anche per tutti i punti nello spazio tridimensionale. Quasi tutti gli algoritmi usati nella Illuminazione Globale si limitano a calcolare i valori di radianza sui punti alla superficie (assumendo ancora la propagazione nel vuoto).

**3.2. Rilevanza ottica.** La radianza è la grandezza a cui rispondono i sensori, come ad esempio le videocamere ed il sistema visivo umano. La risposta dei sensori è proporzionale alla radianza incidente: le costanti di proporzionalità dipendono dalla geometria del sensore. È chiaro quindi che la radianza è la grandezza che gli algoritmi di Illuminazione Globale devono calcolare.

## 4. Esempi

4.1. Emettitore diffusivo. Consideriamo l'esempio di un emettitore puramente diffusivo. Per definizione, esso emette uguale radianza in tutte le direzioni da ogni punto della sua superficie (come in Figura 3).

$$L(x \to \vec{\Theta}) = L$$





FIGURA 3. Emettitore diffusivo

La potenza emessa dall'emettitore diffusivo si calcola così:

$$\begin{split} \Phi &= \int_{\sigma} \int_{\Omega} L(x \to \vec{\Theta}) \cos \theta \, d\omega(\vec{\Theta}) \, d\sigma(x) \\ &= \int_{\sigma} \int_{\Omega} L \cos \theta \, d\omega(\vec{\Theta}) \, d\sigma(x) \\ &= L(\int_{\sigma} d\sigma(x)) (\int_{\Omega} \cos \theta \, d\omega(\vec{\Theta})) \\ &= L\sigma \int_{0}^{2\pi} d\phi \int_{0}^{\frac{\pi}{2}} \cos \theta \, \sin \theta \, d\theta \\ &= 2\pi L\sigma \int_{0}^{\frac{\pi}{2}} \frac{1}{2} \sin 2\theta \, d\theta \\ &= \frac{1}{2}\pi L\sigma \int_{0}^{\pi} \sin u \, du \\ &= \pi L\sigma \,. \end{split}$$

dove  $\sigma$  è l'area dell'emettitore diffusivo. Dall'identità precedente otteniamo la seguente relazione tra potenza, radianza e radiosità di una superficie diffusiva:

$$\Phi = L\sigma\pi = B\sigma. \tag{4.1}$$

4.2. Un emettitore non diffusivo. Consideriamo una sorgente luminosa di area  $10 \times 10 \ cm^2$ , ogni punto su questa sorgente emette radianza secondo la seguente distribuzione:

#### 5. EMISSIONE DELLA LUCE

$$L(x \to \vec{\Theta}) = 6000 \cos \theta (W/sr \cdot m^2)$$

Ricordiamo cha la funzione della radianza è definita per tutte le direzioni nell'emisfero, e tutti i punti sulla superficie. Questa particolare distribuzione è la stessa per tutti i punti nelle sorgenti luminose; comunque, per ogni punto della superficie, c'è un decadimento. La radiosità per ogni punto può essere calcolata come segue:

$$B = \int_{\Omega} L(x \to \vec{\Theta}) \cos \theta \, d\omega(\vec{\Theta})$$
  
= 
$$\int_{\Omega} 6000 \cos^2 \theta \, d\omega(\vec{\Theta})$$
  
= 
$$6000 \int_{0}^{2\pi} \int_{0}^{\pi/2} \cos^2 \theta \sin \theta \, d\theta \, d\phi$$
  
= 
$$6000 \cdot 2\pi \cdot \left[\frac{-\cos^3 \theta}{3}\right]_{0}^{\pi/2}$$
  
= 
$$4000\pi \, W/m^2$$
  
= 
$$12566 \, W/m^2 \, .$$

La potenza emessa dall'intera sorgente luminosa può essere calcolata come segue:

$$\begin{split} \Phi &= \int_{\sigma} \int_{\Omega} L(x \to \vec{\Theta}) \cos \theta \, d\omega(\vec{\Theta}) \, d\sigma(x) \\ &= \int_{\sigma} (\int_{\Omega} L \cos \theta \, d\omega(\vec{\Theta})) \, d\sigma(x) \\ &= \int_{\sigma} B(x) \, d\sigma(x) \\ &= 4000\pi \, W/m^2 \cdot 0.1 \cdot 0.1 \, m \\ &= 125.66 \, W \, . \end{split}$$

#### 5. Emissione della luce

Ci sono vari tipi di sorgenti di luce, per esempio una sorgente come il sole, o altre che emettono in base ad effetti quantistici come nel caso della fluorescenza, dove i materiali assorbono energia a qualche lunghezza d'onda e la riemettono a qualche altra. Come detto prima, per la Computer Graphics non serve una spiegazione dettagliata della luce al livello di meccanica quantistica. Nella maggior parte degli algoritmi di rendering si assume che la luce sia emessa da una sorgente con una dato range di lunghezze d'onda e con una data intensità al variare della lunghezza d'onda (spesso specificata, in maniera approssimata, per le sole componenti primarie rossa, verde e blu). Il calcolo accurato della Illuminazione Globale richiede di specificare le seguenti tre distribuzioni per ogni sorgente di luce: spaziale, direzionale e di intensità spettrale. Le luci puntiformi sono in realtà approssimazioni di sorgenti con estensione piccola, e quindi l'approssimazione puntiforme porta ad una distribuzione spaziale non completamente accurata: le sorgenti di luce estese sono più realistiche. Le distribuzioni direzionali di tipiche sorgenti di luce sono di solito determinate dalla forma della sorgente stessa (approssimata come un integrale di sorgenti puntiformi isotrope).

Sebbene la distribuzione spettrale della luce possa essere sintetizzata con precisione, motivi di efficienza gli algoritimi di Illuminazione Globale tipicamente la sintetizzano in RGB (o in uno spazio di colore analogo, a tre parametri).

#### 6. Interazione della luce con le superfici

L'energia luminosa emessa in una scena interagisce con i diversi oggetti della scena, tramite riflessioni o trasmissioni. Una parte dell'energia luminosa potrebbe essere assorbita dalle superficie e dissipata sotto forma di calore, ma questo fenomeno solitamente non è considerato negli algoritmi di rendering.

**6.1. BRDF.** I materiali interagiscono con la luce in modi differenti, e l'aspetto di materiali diversi appare diverso con le stesse condizioni di luce. Alcuni materiali sono speculari; altri sono diffusivi. L'aspetto dipende anche dalle proprietà rifrattive: gli oggetti semitrasparenti appaiono diversi da quelli opachi. Assumiamo che la luce incidente ad una superficie esca con la stessa lunghezza d'onda con cui era arrivata, ed allo stesso istante: ovvero, trascuriamo effetti come la fluorescenza e la fosforescenza. In molte situazioni fisiche, la luce arriva ad una superficie in un punto x da una direzione di incidenza  $\vec{\Theta}_i$ , e poi riparte dalla superficie da un altro punto y in una direzione  $\Theta_r$ . La funzione che definisce la percentuale di luce incidente a x dalla direzione  $\vec{\Theta}_i$  che viene riflessa o rifratta a y in direzione  $\vec{\Theta}_r$  si chiama *Bidirectional Scat*tering Surface Reflectance Distribution Function (BSSRDF). Facciamo l'ipotesi semplificatrice che la luce incidente in un certo punto esce allo stesso punto: quindi non trattiamo lo scattering interno agli oggetti, che provoca l'uscita di luce da un punto diverso.


FIGURA 4. Bidirectional Reflectance Distribution Function

Le proprietà di riflettività di una superficie sono descritte da una funzione chiamata *Bidirectional Reflectance Distribution Function* (BRDF). La BRDF nel punto x è definita come la frazione della radianza differenziale incidente da una direzione  $\vec{\Psi}$  che viene riflessa in una direzione  $\vec{\Theta}$ (o più precisamente, la derivata della radianza riflessa rispetto a quella incidente). Essa si indica con  $f_r(x, \vec{\Psi} \to \vec{\Theta})$ :

$$f_r(x, \vec{\Psi} \to \vec{\Theta}) = \frac{dL(x \to \vec{\Theta})}{dE(x \leftarrow \vec{\Theta})}$$
$$= \frac{dL(x \to \vec{\Theta})}{L(x \leftarrow \vec{\Psi}) \langle N_x, \vec{\Psi} \rangle \, d\omega(\vec{\Psi})}$$

dove  $\langle N_x, \vec{\Psi} \rangle = \cos(N_x, \vec{\Psi})$  è il coseno dell'angolo formato dal vettore normale  $N_x$  alla superficie nel punto x e la direzione del vettore incidente  $\vec{\Psi}$ .

Si osservi che in generale la BRDF è definita per tutte le direzioni (non solo per quelle nell'emisfero frontale): naturalmente, nel caso di corpi opachi, è nulla nelle direzioni interne (ossia fuori dell'emisfero frontale) ma non per le superficie semitrasparenti. Per abbracciare sia le direzioni trasparenti sia quelle riflettenti a volte si usa il termine *Bidirectional Scattering Distribution Function* (BSDF).

#### 6.2. Proprietà della BRDF.

6.2.1. *Range.* La BRDF può assumere qualsiasi valore positivo, e può variare con la lunghezza d'onda.

6.2.2. Dimensione. La BRDF è una funzione a quattro dimensioni definita da ogni punto sulla superficie; due dimensioni corrispondono alle direzioni in arrvo, e due alle drezioni in uscita. Talvolta la BRDF è anisotropa (il valore di  $f_r$  cambia se la superficie ruotata attorno alla direzione normale), ma molti materiali sono isotropi.

6.2.3. *Reciprocità*. Il valore della BRDF rimane invariato se scambiamo fra loro le direzioni di ingresso ed uscita. Questa proprietà si chiama reciprocità si Helmholtz. In formule:

$$f_r(x, \vec{\Psi} \to \vec{\Theta}) = f_r(x, \vec{\Psi} \leftarrow \vec{\Theta})$$

Pertanto, per indicare che le due direzioni possono essere scambiate, usiamo la seguente notazione:

$$f_r(x, \vec{\Theta} \leftrightarrow \vec{\Psi})$$

6.2.4. Relazione tra radianza incidente e riflessa. Il valore della BRDF per una specifica direzione incidente non dipende dalla irradianza lungo altri angoli incidenti: la BRDF è lineare rispetto alle direzioni incidenti. Pertanto, data una figura opaca non emissiva, per ottenere la radianza riflessa totale ad un suo punto x si deve integrare sull'emisfero frontale a x la radianza entrante usando come peso la BRDF, come segue:

$$dL(x \to \vec{\Theta}) = f_r(x, \vec{\Psi} \leftrightarrow \vec{\Theta}) \, dE(x \leftarrow \vec{\Psi})$$
$$L(x \to \vec{\Theta}) = \int_{\Omega_x} f_r(x, \vec{\Psi} \leftrightarrow \vec{\Theta}) \, dE(x \leftarrow \vec{\Psi})$$
$$L(x \to \vec{\Theta}) = \int_{\Omega} f_r(x, \vec{\Psi} \leftrightarrow \vec{\Theta}) \, L(x \leftarrow \vec{\Psi}) \, \langle N_x, \vec{\Psi} \rangle \, d\omega(\vec{\Psi})$$

6.2.5. Conservazione dell'energia. La legge di conservazione dell'energia richiede che l'ammontare totale dell'energia riflessa su tutte le direzioni deve essere inferiore o uguale alla quantità di energia incidente sulla superficie, l'eccesso di energia viene poi trasformato in calore o altre forme di energia. Per ogni distribuzione di radianza incidente  $L(x \leftarrow \vec{\Psi})$  nell'emisfero, l'energia totale incidente per unità di superficie è l'irradianza totale dell'emisfero stesso:

$$E = \int_{\Omega_x} L(x \leftarrow \vec{\Psi}) \langle N_x, \vec{\Psi} \rangle \, d\omega(\vec{\Psi})$$

L'energia totale riflessa M è un integrale doppio nell'emisfero. Supponiamo di avere una distribuzione di radianza uscente da una superficie, l'energia totale per unità di superficie che esce da M, è:

$$M = \int_{\Omega_x} L(x \leftarrow \vec{\Theta}) \langle N_x, \vec{\Theta} \rangle \, d\omega(\vec{\Theta}) \, .$$

Segue dalla definizione di BRDF che

$$dL(x \to \vec{\Theta}) = f_r(x, \vec{\Psi} \leftrightarrow \vec{\Theta}) L(x \leftarrow \vec{\Psi}) \langle N_x, \vec{\Psi} \rangle \, d\omega(\vec{\Theta})$$

Integrando questa equazione per ottenere il valore di  $L(x \leftrightarrow \vec{\Theta})$  e combinandola con l'espressione di M abbiamo:

$$M =$$

$$\int_{\Omega_x} \int_{\Omega_x} f_r(x, \vec{\Psi} \leftrightarrow \vec{\Theta}) L(x \rightarrow \vec{\Theta}) \langle N_x, \vec{\Theta} \rangle \langle N_x, \vec{\Psi} \rangle \, d\omega(\vec{\Psi}) \, d\omega(\vec{\Theta})$$

La BRDF soddisfa il principio di conservazione dell'energia per riflessione in un punto della superficie se e solo se, per qualsiasi distribuzione angolare  $L(x \leftarrow \vec{\Psi})$  della radianza incidente, vale la disuguaglianza  $M \leq E$ , ovvero:

$$\frac{\int_{\Omega_x} \int_{\Omega_x} f_r(x, \vec{\Psi} \leftrightarrow \vec{\Theta}) L(x \leftarrow \vec{\Psi}) \langle N_x, \vec{\Theta} \rangle \langle N_x, \vec{\Psi} \rangle d\omega(\vec{\Psi}) d\omega(\vec{\Theta})}{\int_{\Omega_x} L(x \leftarrow \vec{\Psi}) \langle N_x, \vec{\Psi} \rangle d\omega(\vec{\Psi})} \leqslant 1 \quad (6.1)$$

Questa disuguaglianza deve essere vera per qualsiasi distribuzione di radianza incidente. Mettiamoci nel caso in cui questa distribuzione angolare sia una delta di Dirac (ossia una distribuzione con supporto in una unica direzione), cosicché gli integrali scompaiono:

$$L(x \leftarrow \vec{\Psi}) = L_{in} \,\delta(\vec{\Psi} - \vec{\Theta}_0) \,.$$

Allora la disuguaglianza (6.1) può essere semplificata come segue: per ogni direzione  $\vec{\Theta}_0$ ,

$$\int_{\Omega_x} f_r(x, \vec{\Theta}_0) \leftrightarrow \vec{\Theta}) \langle N_x, \vec{\Theta} \rangle \, d\omega(\vec{\Theta}) \leqslant 1 \, .$$

Questa disuguaglianza è una condizione necessaria per la conservazione dell'energia: esprime la conservazione dell'energia solo per una specifica distribuzione di radianza incidente, quella puntiforme. Ma, grazie alla linearità della BRDF rispetto alla intensità ed alla direzione incidente, essa è anche una condizione sufficiente: le radianze incidenti da direzioni differenti non interferiscono, e quindi la radianza uscente per una data distribuzione di radianza incidenteè l'integrale delle radianze incidenti rispetto all'angolo di incidenza. In altre parole, la irradianza si può considerare come l'integrale di irradianze di una distribuzione angolare di sorgenti puntiformi sull'emisfero frontale. Allora la stessa disuguaglianza vale per per ogni integrale di valori di radianza incidente puntiforme: quindi per ogni distribuzione angolare. Però in modelli di riflessione in cui il valore della BRDF dipenda anche dalla luce in arrivo da altre direzioni occorre limitarsi alla disuguaglianza più generale (6.1).

Gli algoritmi di Illuminazione Globale spesso scelgono la BRDF in base a qualche modello empirico. Bisogna avere molta cura di accertarsi che questi modelli empirici siano una fisicamente sensati e plausibili: in particolare devono essere soddisfatti la conservazione dell'energia ed il principio di reciprocità di Helmoltz. Il principio di reciprocità di Helmoltz costituisce un vincolo particolarmente importante per gli algoritmi bidirezionali di Illuminazione Globale: questi algoritmi calcolano la distribuzione dell'energia della luce considerando sia percorsi a partire dalle fonti di luce, sia percorsi a partire dall'osservatore: si assume che il percorso della luce sia invertibile, e quindi è essenziale che l'espressione della BRDF soddisfai il principio di reciprocità di Helmoltz.

**6.3. Esempi di BRDF.** A seconda della natura della BRDF, l'aspetto del materiale può essere quello di una superficie diffusiva, uno specchio od una superficie lucida. Esaminiamo brevemente i tipi



FIGURA 5. Tre diversi tipi di BRDF: (a) Materiale diffusivo ideale (la luce viene riflessa in tutte le direzioni); (b) Materiale speculare ideale (la luce viene riflessa in una sola direzione); (c) La BDRF *glossy* riflette la luce in base ad una combinazione di diffusione e riflessione speculare

più comuni di BRDF.

6.3.1. Superficie diffusive. I materiali puramente diffusivi riflettono la luce uniformemente su tutto l'emisfero frontale: data una qualsiasi distribuzione di irradianza, la radianza riflessa è indipendente della direzione uscente, ossia il valore della BRDF è costante per tutti i valori di  $\vec{\Theta}$ . Ad un osservatore, un punto della superficie diffusa sembra della stessa luminosità da tutte le direzioni. La BRDF quindi è

$$f_r(x, \vec{\Psi} \leftrightarrow \vec{\Theta}) = \frac{\rho_d}{\pi}$$

dove  $\rho_d$  è il coefficiente di riflettenza, che misura la frazione di energia incidente che viene riflessa dai punti della superficie (complessivamente per tutti gli angoli di incidenza e riflessione); il suo valore varia ta 0 e 1. Superficie perfettamente speculari riflettono o rifrangono la luce solo in una direzione specifica.

# • *Riflessione speculare.*

Per uno specchio ideale, la direzione della luce riflessa si calcola grazie alla legge della riflessione: la direzioni incidente e uscente formano angoli uguali con la normale alla superficie e sono ad essa complanari. Se la direzione di incidenza si indica con  $\vec{\Psi}$  e la normale alla superficie con N, ne segue (come visto nella Sezione sulla *Riflessione speculare* del Capitolo sul Rendering nella Parte II) che la direzione della luce riflessa è data da:

$$R = 2\langle N, \vec{\Psi} \rangle N - \vec{\Psi} \tag{6.2}$$

Un riflettore speculare ideale ha solo una direzione uscente per la quale la BRDF è diversa da 0: quindi la sua BRDF è una distribuzione  $\delta$  di Dirac. I materiali reali non sono mai specchi ideali, ma solo approssimazioni.

• Rifrazione.

La direzione della rifrazione si calcola in base alla legge di Snell, che abbiamo presentato nella Sezione sulla Trasparenza Rifrattiva del Capitolo sul Rendering nella Parte II, e che qui richiamiamo. Chiamiamo T la direzione lungo la quale si trasmette la luce proviente da un mezzo con indice di rifrazione  $\eta_1$  che si rifrange in un mezzo con indice di rifrazione  $\eta_2$ . La legge di Snell è

$$\eta_1 \sin \theta_1 = \eta_2 \sin \theta_2$$

dove  $\eta_1 \in \eta_2$  sono rispettivamente gli angoli tra il raggio incidente e il raggio trasmesso e la normale alla superficie. Si noti che nel caso  $\theta_1 = 0^0$ , cioè il raggio risulta perpendicolare all'interfaccia, la soluzione è  $\theta_2 = 0^0$  per qualunque valore di  $\eta_1 \in \eta_2$ . In altri termini, un raggio che entra in un mezzo in direzione perpendicolare alla sua superficie non viene deviato. Se per semplicità scriviamo il prodotto interno fra la normale e la direzione di provenienza come  $N \cdot \vec{\Psi}$  invece che  $\langle N, \vec{\Psi} \rangle$ , la direzione T del raggio trasmesso è data da:

$$T = -\frac{\eta_1}{\eta_2} \vec{\Psi} + N(\frac{\eta_1}{\eta_2} \cos \theta_1 - \sqrt{1 - (\frac{\eta_1}{\eta_2})^2 (1 - \cos^2 \theta_1))}$$
  
=  $-\frac{\eta_1}{\eta_2} \vec{\Psi} + N(\frac{\eta_1}{\eta_2} (N \cdot \vec{\Psi}) - \sqrt{1 - (\frac{\eta_1}{\eta_2})^2 (1 - (N \cdot \vec{\Psi})^2)}), \quad (6.3)$ 

dove  $\cos \theta_1 = N \cdot \vec{\Psi}$ , è il prodotto interno fra la normale e la direzione di provenienza. Quando la luce viaggia da un mezzo denso ad un altro meno denso essa può subire una riflessione dentro il mezzo iniziale invece che rifrangersi nel mezzo meno denso. Questo processo è chiamato riflessione interna totale, ed accade per tutti gli angoli di incidenza maggiori o uguali dell'angolo critico  $\theta_c$  per cui la luce viene rifratta nella direzione radente la superficie di separazione:

$$\eta_1 \sin \theta_c = \eta_2 \sin \frac{\pi}{2} \,,$$

ossia

$$\sin \theta_c = \frac{\eta_2}{\eta_1}$$

Si può ricavare la stessa condizione dall'identità 6.3: la riflessione interna totale si comincia a verificare appena il termine sotto la radice quadrata,  $1 - (\frac{\eta_1}{\eta_2})^2 (1 - (N \cdot \vec{\Psi})^2)$ , diventa minore di 0.

• Reciprocità per superficie trasparenti. Dobbiamo stare attenti quando consideriamo la BSDF di un oggetto trasparente: per esse non vale la reciprocità. In conseguenza alla legge della rifrazione di Snell, quando un fascio di luce entra in un mezzo denso provenendo da un mezzo meno denso, viene piegato in modo da avvicinarsi alla direzione normale, e quindi la potenza luminosa per unità di area perpendicolare alla superficie (ossia la radianza) aumenta. Quando un fascio di luce parte da un mezzo denso e viene rifratto in un mezzo meno denso succede il contrario. In conseguenza della legge di Snell, la variazione della radianza è data dal quadrato del rapporto degli indici di rifrazione:  $(\eta_2/\eta_1)^2$ . Nel caso di superficie trasparenti occorre includere questo fattore.



FIGURA 6. Rifrazione e riflessione speculare

• Equazione di Fresnel. Le equazioni precedenti stabiliscono le direzioni di riflessione e rifrazione della luce incidente ad una superficie liscia ideale, ma non la quantità di energia luminosa che viene riflessa: a questo scopo sono necessarie le seguenti equazioni di Fresnel. Quando la luce colpisce una superficie liscia, la percentuale di energia luminosa che viene riflessa o rifratta dipende dalla lunghezza d'onda della luce, dalla geometria della superficie e dalla direzione di incidenza. Chiamiamo rispettivamente  $r_p \in r_s$  le percentuali di energia luminosa che viene riflessa per le due componenti della luce polarizzata: in base alla legge di Fresnel, esse valgono

$$r_p = \frac{\eta_2 \cos \theta_1 - \eta_1 \cos \theta_2}{\eta_2 \cos \theta_1 + \eta_1 \cos \theta_2} \tag{6.4}$$

## 6. INTERAZIONE DELLA LUCE CON LE SUPERFICI

$$r_p = \frac{\eta_1 \cos \theta_1 - \eta_2 \cos \theta_2}{\eta_1 \cos \theta_1 + \eta_2 \cos \theta_2}, \qquad (6.5)$$

dove  $\eta_1$  e  $\eta_2$  sono gli indici di rifrazione delle due superfici. Se la la luce non è polarizzata, allora la percentuale di energia riflessa è

$$F = \frac{|r_p|^2 + |r_s|^2}{2}.$$
(6.6)

Bisogna sottolineare che queste equazioni si applicano sia per materiali conduttivi (metalli) sia per dielettrici (non conduttori, quindi non metalli); per i metalli, l'indice di rifrazione diventa una variabile a valori complessi,  $\eta + ik$ , mentre per i dielettrici l'indice di rifrazione è un numero reale e k = 0. Il coefficiente k è l'esponente che determina il decadimento esponenziale all'aumentare della profondità dell'energia luminosa che entra in profondità nel metallo.

L'equazione di Fresnel si basa sul principio che la luce è o riflessa o rifratta quando interagisce con una superficie puramente speculare o rifrattiva, ma non assorbita. Poiché non vi è alcun assorbimento di energia luminosa, la somma dei coefficienti di riflessione e rifrazione vale 1.

• Superficie lucide.

La maggior parte delle superficie non consiste né di diffusori ideali né di specchi ideali, ma invece di casi intermedi; queste superficie sono chiamate superficie lucide. La loro BRDF è difficile da tradurre in formule.

6.4. Modelli di illuminazione. la BRDF dei materiali reali può essere abbastanza complessa; diversi modelli sono stati suggeriti in Computer Graphics per catturare la complessità della BRDF. Indichiamo con  $\vec{\Psi}$  la direzione di incidenza della luce e con  $\vec{\Theta}$  la direzione dell'osservatore.

6.4.1. *Modello di Lambert.* Il modello euristico più semplice materiali diffusivi è ttribuito a Lambert. In questo modello, la BRDF è constante, come precedentemente osservato.

$$f_r(x, \vec{\Psi} \leftrightarrow \vec{\Theta}) = k_d = \frac{\rho_d}{\pi}.$$

Il fattore  $\pi$  al denominatore è il fattore di normalizzazione calcolato in (4.1).



FIGURA 7. shading models geometry

6.4.2. *Modello di Phong.* Il modello euristico di illuminazione di Phong richiede solo prodotti interni, ovvero somme e moltiplicazioni, e quindi è supportabile in hardware, e per questo è molto usato. Come visto nella Sezione sulla *Illuminazione di Phong* del Capitolo sul Rendering nella Parte II, la BRDF per il modello di Phong è

$$f_r(x, \vec{\Psi} \leftrightarrow \vec{\Theta}) = k_s = \frac{\langle R, \vec{\Theta} \rangle^n}{N \cdot \vec{\Psi}} + k_d$$

dove R, il vettore riflesso, può essere calcolato dall'equazione 6.2. La semplicità di questo modello è interessante, ma esso ha serie limitazioni: come osservato njella Sezione citata prima, esso non conserva l'energia, non soddisfa la proprietà di reciprocità di Helmoltz nello scambio delle direzioni  $\vec{\Psi} \in \vec{\Theta}$ , e non riproduce accuratamente il comportamento dei materiali reali.

6.4.3. *Modello di Blinn-Phong.* Il modello di Blinn-Phong riscrive l'espressione precedente in termini del vettore medio (*halfway vector*) *H* fra  $\vec{\Psi} \in \vec{\Theta}$ , come segue:

$$f_r(x, \vec{\Psi} \leftrightarrow \vec{\Theta}) = k_s \frac{\langle N, H \rangle^n}{\langle N, \vec{\Psi} \rangle} + k_d$$

Questo elimina una delle cause della non conservazione dell'energia nel modello di Phong. Infatti, in tale modello compare il coseno dell'angolo fra la direzione dell'osservatore  $\vec{\Theta}$  ed il versore della riflessione speculare della luce R. Nel caso in cui luce ed osservatore sono in direzioni vicine e quasi radenti alla superficie (o almeno ad inclinazione maggiore di  $\pi/4$ ), il versore dell'osservatore forma un angolo di più di  $\pi/2$  con R, e quindi il coseno è negativo e viene rimpiazzato con zero (perché contributi negativi di illuminazione non hanno senso e non sono ammessi). Quindi in tali casi si perde un contributo di illuminazione che in un modello fisico e non euristico sarebbe positivo e tenuto in conto. Ma nel modello di Blinn-Phong l'angolo fra  $\vec{\Theta}$  e R è rimpiazzato da quello fra N e H, che è sempre inferiore a  $\pi/2$ , e quindi ha coseno positivo e non porta a troncamenti a zero che causano perdite di energia. Nondimeno, l'energia non si conserva ugualmente.

6.4.4. Modello di Blinn–Phong modificato. Il modello di Blinn–Phong modificato risolve il problema della perdita della reciprocità di Helmoltz grazie alla seguente formulazione euristica, che simmetrizza l'espressione rinunciando al denominatore (ma così facendo sottostima i contributi di sorgenti di luce ad alta deviazione, cioè quasi radente alla superficie illuminata):

$$f_r(x, \Psi \leftrightarrow \Theta) = k_s \langle N, H \rangle^n + k_d$$

6.4.5. Modello di Cook-Torrance. Il modello di Cook-Torrance approssima le superficie reali come microsfaccettate, ossia costituite da un insieme di piccole sfaccettature planari speculari la cui angolazione è casuale con una deterrminata distribuzione di probabilità la cui varianza dioende da quanto diffusiva è globalmente la superficie. Quindi ogni raggio incidente colpisce a caso uno di queste sfaccettature piatte. Ogni materiale ha la sua appropriata distribuzione di probabilità delle inclinazioni delle microsfaccettature. Nel modello viene calcolato il risultato medio della riflessione della luce in base ai termini di riflessione e rifrazione di Fresnel sulle microfacce ed alla distribuzione della inclinazione delle sfaccettature. Il risultato è

$$f_r(x, \vec{\Psi} \leftrightarrow \vec{\Theta}) = \frac{F(\beta)D(\vec{\Theta}_h)G}{\pi \langle N, \vec{\Psi} \rangle \langle N, \vec{\Theta} \rangle} + k_d \,,$$

dove i tre termini nella componente non diffusa della BRDF sono i coefficienti di riflessione di Fresnel F, la distribuzione delle sfaccettature D e il termine geometrico di ombreggiatura G. Ora spieghiamo ciascuno di questi termini.

Si fa uso dei termini di Fresnel nelle equazioni (6.4) e (6/5). Il modello assume che la luce non sia polarizzata: pertanto,  $F = \frac{|r_p|^2 + |r_s|^2}{2}$  (6.6). Il termine di riflettenza di Fresnel è calcolato rispetto all'angolo  $\beta$  tra la direzione incidente e lo *halfway vector*:  $\cos \beta = \langle \vec{\Psi}, H \rangle = \langle \vec{\Theta}, H \rangle$ (in effetti, in base alla definizione di *halfway vector*, questo angolo è lo stesso di quello tra la direzione riflessa uscente e lo *halfway vector*).

La funzione di distribuzione D specifica la distribuzione delle micro sfaccettature per il materiale. Ci sono varie scelte per questa distribuzione di probabilità: una delle più comuni è la distribuzione di Beckmann,

$$D(\vec{\Theta}_h) = \frac{1}{m^2 \cos^4 \vec{\Theta}_h} e^{-(\frac{\tan \vec{\Theta}_h}{m})^2}$$

dove  $\vec{\Theta}_h$  è l'angolo tra la normale e lo *halfway vector* e  $\cos \vec{\Theta}_h = \langle N, H \rangle$ . Il termine geometrico G tiene conto dell'occlusione reciproca fra le microfacce e della loro auto-copertura:

$$G = \min\left\{1, \frac{2\langle N, H \rangle \langle N, \vec{\Theta} \rangle}{\langle \vec{\Theta}, H \rangle}, \frac{2\langle N, H \rangle \langle N, \vec{\Psi} \rangle}{\langle \vec{\Theta}, H \rangle}\right\}$$

# 7. Equazione del rendering

Ora siamo pronti per formulare la condizione del trasporto della potenza luminosa in una scena tramite una equazione ricorsiva, l'equazione del rendering. L'obiettivo degli algoritmi di Illuminazione Globale è calcolare la distribuzione di equilibrio della potenza luminosa scambiata fra le varie parti della scena. Come precedentemente menzionato, consideriamo il caso in cui la luce si propaga nel vuoto, quindi in assenza di mezzi atmosferici. Su ogni punto della superficie x ed in ogni direzione  $\vec{\Theta}$ , l'equazione del rendering descrive la radianza uscente  $L(x \to \vec{\Theta})$  dal punto x in direzione  $\vec{\Theta}$  come integrale della radianza incidente dalle varie direzioni.

7.1. Formulazione emisferica. La formulazione più comune è quella emisferica: la spieghiamo in base alla conservazione dell'energia che passa per un punto x. Supponiamo che  $L_e(x \to \vec{\Theta})$  sia la radianza creata da una superficie nel punto x ed inviata nella direzione  $\vec{\Theta}$ , e  $L_r(x \to \vec{\Theta})$  sia la radianza che viene riflessa dalla superficie nel punto x nella direzione  $\vec{\Theta}$  dopo essere arrivata a x dalle varie direzioni incidenti. Per la conservazione dell'energia, la radianza totale uscente da un punto in una particolare direzione di uscita è la somma della radianza emessa

e della radianza riflessa in quella direzione a quel punto della superficie. Quindi:

$$L(x \to \vec{\Theta}) = L_e(x \to \vec{\Theta}) + L_r(x \to \vec{\Theta}).$$

Dalla definizione di BRDF abbiamo

$$f_r(x, \vec{\Psi} \leftrightarrow \vec{\Theta}) = \frac{dL_r(x \to \vec{\Theta})}{dE(x \leftarrow \vec{\Psi})},$$

$$L_r(x \to \vec{\Theta}) = \int_{\Omega_x} f_r(x, \vec{\Psi} \leftrightarrow \vec{\Theta}) L(x \leftarrow \vec{\Psi}) \langle N_x, \vec{\Psi} \rangle \, d\omega(\vec{\Psi})$$

Combinando queste equazioni, otteniamo l'equazione del rendering nella forma dell'integrale sull'emisfero:

$$L(x \to \vec{\Theta}) = L_e(x \to \vec{\Theta}) + \int_{\Omega_x} f_r(x, \vec{\Psi} \leftrightarrow \vec{\Theta}) L(x \leftarrow \vec{\Psi}) \langle N_x, \vec{\Psi} \rangle \, d\omega(\vec{\Psi}) \, .$$
(7.1)

Si tratta di una equazione integrale ricorsiva (una equazione di Fredholm di secondo tipo).

7.2. Formulazione di area. Una formulazione alternativa frequente si ottiene integrando i contributi della radianza inviata a x non sull'emisfero frontale, bensì su tutti i punti visibili da x nelle superficie degli oggetti della scena.

Prendiamo in considerazione i termini della equazione del rendering (7.1) nella precedente formulazione emisferica. Assumendo che non ci sia nessun mezzo di propagazione, sappiamo che la radianza che entra nel punto x in provenienza da  $\vec{\Psi}$  è la stessa che esce da y in direzione  $-\vec{\Psi}$ :

$$L(x \leftarrow \vec{\Psi}) = L(y \to -\vec{\Psi})$$

Inoltre, l'elemento di integrazione rispetto all'angolo solido può essere scritto come segue:

$$d\omega(\vec{\Psi}) = d\omega_{x\leftarrow d\sigma(y)} = \langle N_y, -\vec{\Psi} \rangle \; \frac{d\sigma(y)}{r_{xy}^2}$$

Sostituendo nell'equazione (7.1), trasformiamo l'integrale dell'equazione del rendering in un integrale su tutte le superficie della scena:

$$L(x \to \vec{\Theta}) = L_e(x \to \vec{\Theta}) + \int_{\sigma} f_r(x, \vec{\Psi} \leftrightarrow \vec{\Theta}) L(y \to -\vec{\Psi}) V(x, y) \frac{\langle N_x, \vec{\Psi} \rangle \langle N_y, \vec{\Psi} \rangle}{r_{xy}^2} d\sigma(y) .$$

Denotiamo con G(x, y) il termine che dipende solo dalla geometria degli oggetti nei punti  $x \in y$ :

$$G(x,y) = \frac{\langle N_x, \vec{\Psi} \rangle \langle N_y, \vec{\Psi} \rangle}{r_{xy}^2} \,.$$

Allora

$$\begin{split} L(x \to \vec{\Theta}) &= L_e(x \to \vec{\Theta}) \\ &+ \int_{\sigma} f_r(x, \vec{\Psi} \leftrightarrow \vec{\Theta}) \, L(y \to -\vec{\Psi}) \, V(x, y) \, G(x, y) \, d\sigma(y) \end{split}$$

Questa è la formulazione dell'equazione del rendering in termini di integrazione su tutte le superficie della scena.

**7.3. Radianza diretta e indiretta.** Possiamo riformulare l'equazione del rendering separando i termini di illuminazione diretti da quelli indiretti. L'illuminazione diretta è quella che arriva direttamente su una superficie, proveniente da una sorgente luminosa nella scena; l'illuminazione indiretta è la luce che arriva dopo essere rimbalzata su altri oggetti della scena.

Se separiamo nell'integrale i termini di illuminazione diretta da quelli di illuminazione indiretta, scomponiamo l'equazione del rendering come segue:

$$\begin{split} L(x \to \vec{\Theta}) &= L_e(x \to \vec{\Theta} + L_r(x \to \vec{\Theta}) \,, \\ L_r(x \to \vec{\Theta}) &= \int_{\Omega_x} f_r(x, \vec{\Psi} \leftrightarrow \vec{\Theta}) L(x \leftarrow \vec{\Psi}) \, \langle N_x, \vec{\Psi} \rangle \, d\omega(\vec{\Psi}) \,, \\ &= L_{diretta} + L_{indiretta} \,, \\ L_{diretta} &= \int_{\sigma} f_r(x, \vec{x} \vec{y} \leftrightarrow \vec{\Theta}) L_e(y \to \vec{y} \vec{x}) \, V(x, y) \, G(x, y) \, d\sigma(y) \,, \\ L_{indiretta} &= \int_{\Omega_x} f_r(x, \vec{\Psi} \leftrightarrow \vec{\Theta}) L_i(x \leftarrow \vec{\Psi}) \, \langle N_x, \vec{\Psi} \rangle \, d\omega(\vec{\Psi}) \,, \\ L_i(x \leftarrow \vec{\Psi}) &= L_r(r(x, \vec{\Psi}) \to -\vec{\Psi}) \,. \end{split}$$

Così, il termine diretto è il contributo della radianza creata in punti della scena  $y = r(x, \vec{xy})$  visibili da x ed inviata a x (ossia lungo la direzione  $\vec{xy}$ ); invece l'illuminazione indiretta è la radianza inviata a xda punti  $y = r(x, \vec{\Psi})$  nei quali non viene creata ma solo riflessa.

7.4. L'equazione dell'importanza. Ogni pixel di un'immagine ottenuta dal rendering di una scena si può pensare come un sensore che risponde alla luce che riceve. È opportuno determinare la sua funzione di risposta (o funzione potenziale, o anche funzione di importanza) che stabilisce quali parti o punti della scena sono maggiormente importanti nel determinare l'illuminazione del pixel. Chiamiamo W la funzione di importanza: allora  $W(x, \vec{\Theta})$  determina l'importanza del contributo della direzione  $\vec{\Theta}$  all'illuminazione del punto x. Ora, l'importanza si scinde come somma di due termini. Se un punto od una superficie isono visibili dal punto x, essi assumono per questo una importanza diretta, in quanto inviano luce direttamente a x: indichiamo questa importanza diretta con  $W_e(x, \vec{\Theta})$ . Ma se da *i* è visibile un altro punto o superficie i che è visibile anche da x, allora i invia luce anche a j, e parte di questa luce può essere riflessa su x. Pertanto, dal punto di vista di x, i acquisisce una importanza indiretta aggiuntiva, in quanto fonte di ulteriore illuminazione indiretta. Osserviamo due fatti:

il primo è che l'importanza si propaga in senso oppostro alla radianza: il fatto che la radianza, ossia l'energia, viaggi da i a j fa sì che l'importanza si trasmetta da j a i;

il secondo è che, a parte questa trasmissione all'indietro, l'importanza si propaga con lo stesso meccanismo di riflessioni ricorsive della radianza; in altre parole, l'importanza verifica la seguente equazione integrale ricorsiva analoga a (7.1):

$$W(x \to \vec{\Theta}) = W_e(x \to \vec{\Theta}) + \int_{\Omega_x} f_r(x, \vec{\Psi} \leftrightarrow \vec{\Theta}) W(x \leftarrow \vec{\Psi}) \langle N_x, \vec{\Psi} \rangle \, d\omega(\vec{\Psi})$$
(7.2)

# CAPITOLO 2

# Integrazione numerica con estimatori probabilistici

## 1. Introduzione alla probabilità

DEFINIZIONE 1.1 ( $\sigma$ -algebra). Sia  $\mathcal{Q}$  un insieme. Indichiamo con  $\mathcal{P}(\mathcal{Q})$  l'insieme delle parti di  $\mathcal{Q}$ , ovvero l'insieme formato dai suoi sottoinsiemi. Una famiglia  $\mathcal{A} \subset \mathcal{P}(\mathcal{Q})$  di parti di  $\mathcal{Q}$  si dice una  $\sigma$ -algebra su  $\mathcal{Q}$  se verifica le seguenti proprietà:

- (1)  $\{\emptyset, \mathcal{Q}\} \subset \mathcal{A}$
- (2)  $X \in \mathcal{A} \Longrightarrow X^{\complement} \in \mathcal{A}$  (dove  $X^{\complement}$  indica il complementare di X)
- (3) se  $\{X_i\}_{i\in\mathbb{Z}}\subset\mathcal{A}$ , allora

$$\bigcap_{i\in\mathbb{Z}}X_i\in\mathcal{A}$$

Si osservi che se  $\mathcal{A}$  è una  $\sigma$ -algebra, le proprietà (2) e (3) implicano che per ogni  $\{X_i\}_i \subset \mathcal{A}$  si abbia  $\bigcup_{i \in \mathbb{Z}} X_i \in \mathcal{A}$ . Infatti per ogni collezione numerabile di insiemi  $\{Y_k\}_k$ , vale l'identità  $\bigcup_k Y_k = \left(\bigcap_k Y_k^{\complement}\right)^{\complement}$ .

DEFINIZIONE 1.2 (**Probabilità**). Dato lo spazio misurabile  $(\mathcal{Q}, \mathcal{A})$ , dove  $\mathcal{Q}$  è un insieme e  $\mathcal{A}$  è una  $\sigma$ -algebra su  $\mathcal{Q}$ , una misura  $\mathbf{P}$  su  $(\mathcal{Q}, \mathcal{A})$ è detta *misura di probabilità* o, più spesso, *probabilità*. se verifica le seguenti proprietà:

- (1)  $\mathbf{P}: \mathcal{A} \longrightarrow [0,1]$
- (2)  $\mathbf{P}(\Omega) = 1$
- (3) **P** è  $\sigma$ -additiva, ovvero per ogni  $\{X_j\} \subset \mathcal{A}$  tale che  $X_n \cap X_k = \emptyset, \forall k \neq n$ , si ha

$$\mathbf{P}\left(\bigcup_{j\geqslant 0}X_j\right) = \sum_{j\geqslant 0}\mathbf{P}\left(X_j\right)$$

DEFINIZIONE 1.3 (**Spazio di probabilità**). Uno spazio di probabilità è una tripla  $(\Omega, \mathcal{A}, \mathbf{P})$ , dove  $\Omega$  è uno spazio, detto spazio dei campioni;  $\mathcal{A}$  è una  $\sigma$ -algebra su  $\Omega$  che rappresenta l'insieme di tutti gli eventi possibili (pertanto un elemento  $E \in \mathcal{A}$  si dice evento);  $\mathbf{P}$  è una probabilità su  $(\Omega, \mathcal{A})$ . Dalla definizione seguono alcune fondamentali proprietà

PROPOSIZIONE 1.4. Sia  $(\Omega, \mathcal{A}, \mathbf{P})$  uno spazio di probabilità.

- (1)  $\forall A, B \in \mathcal{A} \ tali \ che \ A \subseteq B, \ \mathbf{P}(A) \leq \mathbf{P}(B).$
- (2)  $\forall \{A_i\}_{i\in\mathbb{N}} \subset \mathcal{A} \ si \ ha$

$$\mathbf{P}\left(\bigcup_{i\in\mathbb{N}}A_i\right)\leqslant\sum_{i\in\mathbb{N}}\mathbf{P}(A_i).$$
(1.1)

(3) 
$$\forall A \in \mathcal{A}, \quad \mathbf{P}(A^{\mathsf{L}}) = 1 - \mathbf{P}(A).$$
  
(4)  $\forall A, B \in \mathcal{A}, \quad \mathbf{P}(A \cup B) = \mathbf{P}(A) + \mathbf{P}(B) - \mathbf{P}(A \cap B).$ 

DIMOSTRAZIONE. Per mostrare (1) basta osservare che  $B = A \cup (A^{\complement} \cap B)$ . Certamente  $A \cap (A^{\complement} \cap B) = \emptyset$ : pertanto dalla definizione di probabilità si ha subito

$$\mathbf{P}(B) = \mathbf{P}(A) + \mathbf{P}\left(A^{\complement} \cap B\right) \ge \mathbf{P}(A).$$

Mostrariamo dapprima la (2) nel caso di due insiemi. Sia  $C = A \cap B^{\complement}$  evidentemente  $C \subseteq A$  e  $B \cap C = \emptyset$ , inoltre  $B \cup C = B \cup A$  e pertanto segue dalla definizione di probabilità e da (1) che

$$\mathbf{P}(A \cup B) = \mathbf{P}(B \cup C) = \mathbf{P}(B) + \mathbf{P}(C) \leqslant \mathbf{P}(B) + \mathbf{P}(A)$$

Si può ragionare analogamente nel caso di più insiemi. Supponiamo data  $\{A_i\}_{i\in\mathbb{N}}$ , poniamo  $C_j = A_j \cap (\bigcup_{i=0}^{j-1}A_i)$ , ne segue  $\bigcup_i C_i = \bigcup_i A_i$  e  $C_i \subseteq A_i$ , dunque la tesi. La (3) è pressoché immediata. Certamente  $A \cap A^{\complement} = \emptyset$ , d'altra parte  $A \cup A^{\complement} = \Omega$ . Dalla definizione di probabilità segue, pertanto,  $1 = \mathbf{P}(\Omega) = \mathbf{P}(A) + \mathbf{P}(A^{\complement})$ . Infine, per provare (4) basta porre

$$A \cup B = \left(A \cap B^{\complement}\right) \cup \left(B \cap A^{\complement}\right) \cup \left(A \cap B\right)$$

dove i tre insiemi a destra dell'ugualianza sono disgiunti. Dall'osservare che  $A = (A \cap B^{\complement}) \cup (A \cap B)$  e  $B = (B \cap A^{\complement}) \cup (A \cap B)$  segue che

$$\mathbf{P}(A \cup B) = \left(\mathbf{P}(A \cap B^{\complement}) + (A \cap B)\right) + \mathbf{P}\left((B \cap A^{\complement}) + (A \cap B)\right) - \mathbf{P}(A \cap B)$$
$$= \mathbf{P}(A) + \mathbf{P}(B) - \mathbf{P}(A \cap B).$$

PROPOSIZIONE 1.5. Sia  $(\Omega, \mathcal{A}, \mathbf{P})$  uno spazio di probabilità . Siano  $\{E_i\}_{i \in \mathbb{N}} \subset \mathcal{A}$  ed  $\{F_i\}_{i \in \mathbb{N}} \subset \mathcal{A}$  due successioni di eventi tali che  $E_i \subset$ 

 $E_{i+1}, F_i \supset F_{i+1} \text{ per ogni } i \in \mathbb{N}.$  Allora

$$\mathbf{P}\left(\bigcup_{n\in\mathbb{N}}E_n\right) = \lim_{n\to+\infty}\mathbf{P}\left(E_n\right), \quad \mathbf{P}\left(\bigcap_{n\in\mathbb{N}}F_n\right) = \lim_{n\to+\infty}\mathbf{P}\left(F_n\right) \quad (1.2)$$

DIMOSTRAZIONE. Consideriamo la successione  $\mathcal{A} \supset \{\mathcal{E}_i\}_{i \in \mathbb{N}}$  ove  $\mathcal{E}_1 = E_1, \mathcal{E}_i = E_i \setminus E_{i-1}$ . Si ha  $\mathcal{E}_i \cap \mathcal{E}_j = \emptyset$ , per ogni  $i \neq j, \cup_{i \in \mathbb{N}} \mathcal{E}_i = \bigcup_{i \in \mathbb{N}} E_i$ . Allora

$$\mathbf{P}\left(\bigcup_{i\in\mathbb{N}}E_{i}\right) = \mathbf{P}\left(\bigcup_{i\in\mathbb{N}}\mathcal{E}_{i}\right) = \sum_{i\in\mathbb{N}}\mathbf{P}\left(\mathcal{E}_{i}\right) = \lim_{n\to+\infty}\sum_{i=1}^{n}\mathbf{P}\left(\mathcal{E}_{i}\right)$$
$$= \lim_{n\to+\infty}\mathbf{P}\left(\bigcup_{i=1}^{n}\mathcal{E}_{i}\right) = \lim_{n\to+\infty}\mathbf{P}\left(\bigcup_{i=1}^{n}E_{i}\right) = \lim_{n\to+\infty}\mathbf{P}\left(E_{n}\right)$$

Con un ragionamento del tutto analogo si dimostra la seconda ugualianza in (1.2)

DEFINIZIONE 1.6 (Indipendenza). Sia  $(\Omega, \mathcal{A}, \mathbf{P})$  uno spazio di probabilità, *n* eventi  $A_1, \ldots, A_n \in \mathcal{A}$  si dicono *indipendenti* se per ogni sottoinsieme  $\{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$  si ha

$$\mathbf{P}\left(\bigcap_{j=1}^{k} A_{i_j}\right) = \prod_{j=1}^{k} \mathbf{P}\left(A_{i_j}\right)$$

DEFINIZIONE 1.7 (**Probabilità condizionale**). Sia  $(\Omega, \mathcal{A}, \mathbf{P})$  uno spazio di probabilità . Siano  $A, B \in \mathcal{A}$  con  $\mathbf{P}(A) > 0$ , si dice *probabilità condizionale di B dato A* il valore

$$\mathbf{P}(B \mid A) = \frac{\mathbf{P}(A \cap B)}{\mathbf{P}(A)}$$

## 1.1. Variabili aleatorie, distribuzioni, funzioni di ripartizione.

DEFINIZIONE 1.8 (Variabile aleatoria). Sia  $(\Omega, \mathcal{A}, \mathbf{P})$  uno spazio di probabilità . Una variabile aleatoria è una funzione  $X : \Omega \longrightarrow \mathbb{R}$  tale che, per ogni  $x \in \mathbb{R}$ , si abbia

$$\{\omega \in \Omega \mid X(\omega) \leqslant x\} \in \mathcal{A} \tag{1.3}$$

OSSERVAZIONE 1.9. Per semplicità di notazione, nel seguito, indicheremo con  $\{X \leq x\}$  l'insieme  $\{\omega \in \Omega \mid X(\omega) \leq x\}$  e scriveremo  $\mathbf{P}(X \leq x)$ al posto di  $\mathbf{P}(\{X \leq x\})$ .

Una variabile aleatoria X si dice discreta se assume al più una infinità numerabile di valori, ovvero se  $X(\Omega) \subseteq \{x_1, x_2, \ldots, x_n, x_{n+1}, \ldots\},\$  dove con  $X(\Omega)$  intendiamo l'immagine di X; si dice *continua* altrimenti. Si osserva facilmente che se X è una variabile aleatoria discreta, per la definizione data, gli insiemi  $\{X = x_i\} = \{\omega \in \Omega \mid X(\omega) = x_i\}$  sono eventi (ovvero sono in  $\mathcal{A}$ ), pertanto ha senso la definizione che segue

DEFINIZIONE 1.10 (**Distribuzione discreta di probabilità**). Sia Xuna variabile aleatoria discreta su  $(\Omega, \mathcal{A}, \mathbf{P})$ , spazio di probabilità . Siano  $\{x_i\}_{i\in\mathbb{N}}$  i valori assunti da X. La funzione  $p_X : \mathbb{R} \longrightarrow [0, 1]$  definita da  $p_X(x) = \mathbf{P}(X = x)$  si dice distribuzione discreta di probabilità di X.

Osserviamo che valgono le due proprietà seguenti

(1)  $p_X(x) = 0$  per quasi tutti gli  $x \in \mathbb{R}$ , tranne al più un insime numerabile.

$$\sum_{i \in \mathbb{N}} p_X(x_i) = 1 \tag{1.4}$$

Infatti gli eventi  $\{X = x_j\}$  sono a due a due disgiunti al variare di  $j \in \mathbb{N}$ , inoltre, essendo  $X(\Omega) = \{x_i\}_{i \in \mathbb{N}}$ , necessariamente  $\bigcup_{i \in \mathbb{N}} \{X = x_i\} = \Omega$ . Ne segue che

$$\sum_{i \in \mathbb{N}} p_X(x_i) = \sum_{i \in \mathbb{N}} \mathbf{P}(X = x_i) = \mathbf{P}\left(\bigcup_{i \in \mathbb{N}} \{X = x_i\}\right) = \mathbf{P}(\Omega) = 1$$

DEFINIZIONE 1.11 (Funzione di ripartizione). La funzione di ripartizione di una variabile aleatoria X è la funzione così definita

$$F_X : \mathbb{R} \longrightarrow [0, 1]; \quad F_X(x) = \mathbf{P}(X \leqslant x)$$

Osserviamo che una funzione di ripartizione è sempre non-decrescente, infatti per ogni coppia  $(x, y) \in \mathbb{R}^2$  tali che  $x \leq y$  si ha  $\{X \leq x\} \subseteq \{X \leq y\}$  e dalla proposizione 1.4 punto (1) segue  $F_X(x) \leq F_X(y)$ . Inoltre, essendo  $\{X = x_i\} \cap \{X = x_j\} = \emptyset$  per ogni  $i \neq j$ , risulta

$$F_X(x) = \sum_{x_i \leqslant x} p_X(x_i). \tag{1.5}$$

Tale formula mostra come la funzione di ripartizione e la distribuzione della variabile aleatoria X, siano sostanzialmente equivalenti. Infatti da (1.5) e dalla Proposizione 1.4 punto (4), si ha

$$p_X(x_i) = \mathbf{P}\left(\{X \le x_i\} \cap \{X > x_{i-1}\}\right) = \mathbf{P}\left(\{X \le x_i\} \cap \{X \le x_{i-1}\}^{\complement}\right)$$
  
=  $\mathbf{P}\left(X \le x_i\right) + (1 - \mathbf{P}\left(X \le x_{i-1}\right)) - \mathbf{P}(\Omega) = F_X(x_i) - F_X(x_{i-1})$   
(1.6)

<sup>(2)</sup> 

Dunque, conoscere  $p_X$  equivale a conoscere  $F_X$  e viceversa.

Come è intuibile, una variabile aleatoria non ha ragione di essere a valori in  $\mathbb{R}$  piuttosto che in  $\mathbb{R}^2$  o  $\mathbb{R}^n$ , si parlerà in questi casi di vettori aletori *n*-dimensionali. Diamone una definizione rigorosa.

DEFINIZIONE 1.12 (Vettore aleatorio discreto). Siano  $X_1, \ldots, X_n$ delle variabili aleatorie discrete tali che  $X_i : \Omega \longrightarrow \mathbb{R}$  per ogni *i*. Poniamo

$$X := (X_1, \dots, X_n), \quad X : \Omega \longrightarrow \mathbb{R}^n.$$
(1.7)

X si dice vettore aleatorio discreto n-dimensionale.

Sia  $\mathbf{x} \in \mathbb{R}^n$ , analogamente al caso monodimensionale, indicheremo con  $p_X(\mathbf{x})$  la probabilità seguente

$$p_X(\mathbf{x}) = \mathbf{P}(X = \mathbf{x}) = \mathbf{P}(\{X_1 = x_1\} \cap \dots \cap \{X_n = x_n\})$$
 (1.8)

dove  $x_1, \ldots, x_n$  sono le componenti del vettore **x**. La distribuzione del vettore X si dice distribuzione congiunta delle variabili  $X_1, \ldots, X_n$ . Analogamente si definisce la funzione di ripartizione di un vettore aleatorio.

OSSERVAZIONE 1.13. Nel seguito, per semplicità di notazione, scriveremo  $\mathbf{P}(X = x, Y = y)$  al posto di  $\mathbf{P}(\{X = x\} \cap \{Y = y\})$ .

In analogia con la definizione 1.6 di eventi indipendenti, diamo la seguente

DEFINIZIONE 1.14 (Variabili aleatorie indipendenti). Siano  $X_1, \ldots, X_n$ variabili aleatorie. Se per ogni possibile *n*-upla di intervalli  $\mathcal{J}^{(1)}, \ldots, \mathcal{J}^{(n)}, \mathcal{J}^{(i)} \subset \mathbb{R}$ , si ha

$$\mathbf{P}\left(X_{1} \in \mathcal{J}^{(1)}, \dots, X_{n} \in \mathcal{J}^{(n)}\right) = \prod_{i=1}^{n} \mathbf{P}\left(X_{i} \in \mathcal{J}^{(i)}\right)$$
(1.9)

diremo che le variabili aleatorie  $X_1, \ldots, X_n$  sono indipendenti.

Se le variabili aleatorie considerate sono un infinità numerabile  $\{X_1, \ldots, X_n, X_{n+1}, \ldots\}$ , diremo che sono indipendenti se e solo se lo sono un qualsiasi loro sottoinsieme finito.

OSSERVAZIONE 1.15. Dalla definizione di indipendenza fra variabili aleatorie segue che, per ogni  $X_1, \ldots, X_n$  variabili aleatorie indipendenti, posto  $X = (X_1, \ldots, X_n)$ , si ha

$$p_X(\mathbf{x}) = p_{X_1}(x_1) \dots p_{X_n}(x_n), \qquad F_X(\mathbf{x}) = F_{X_1}(x_1) \dots F_{X_n}(x_n)$$
(1.10)

e viceversa.

OSSERVAZIONE 1.16. Se X, Z sono due variabili aleatorie indipendenti e  $\lambda : \mathbb{R} \to \mathbb{R}, \mu : \mathbb{R} \to \mathbb{R}$  due funzioni, allora le variabili aleatorie  $\lambda(X), \mu(Z)$  sono indipendenti.

DIMOSTRAZIONE. Volgiamo verificare la (1.10). Indichiamo con  $\lambda^{-1}(x)$ ,  $\mu^{-1}(x)$  la controimmagine di x rispettivamente tramite  $\lambda \in \mu$ . Essendo  $X \in Z$  indipendeti, si ha

$$p_{(\lambda(X),\mu(Z))}(\xi_{i},\zeta_{i}) = \mathbf{P}\left(\lambda(X) = \xi_{i},\mu(Z) = \zeta_{i}\right)$$
$$= \mathbf{P}\left(X \in \lambda^{-1}(\xi_{i}), Z \in \mu^{-1}(\zeta_{i})\right) = \mathbf{P}\left(X \in \lambda^{-1}(\xi_{i})\right) \mathbf{P}\left(Z \in \mu^{-1}(\zeta_{i})\right)$$
$$= \mathbf{P}\left(\lambda(X) = \xi_{i}\right) \mathbf{P}\left(\mu(Z) = \zeta_{i}\right) = p_{\lambda(X)}\left(\xi_{i}\right) p_{\mu(Z)}\left(\zeta_{i}\right)$$

# 1.2. Attesa, varianza e covarianza.

DEFINIZIONE 1.17 (Attesa di una variabile aleatoria). Sia X una variabile aleatoria discreta con distribuzione  $p_X$ . Siano  $\{x_0, x_1, \ldots, x_n, x_{n+1}, \ldots\}$  i valori assunti da X. Se  $\{x_k p_X(x_k)\}_k \in \ell^1$ , si dice che X ha media finita. La quantità

$$E(X) := \sum_{i=0}^{+\infty} x_i p_X(x_i).$$
 (1.11)

si chiama attesa di X.

Se X è un vettore aleatori<br/>on-dimensionale, si definisce attesa di X la quantità

$$E(X) = \sum_{i_1=0}^{+\infty} \cdots \sum_{i_n=0}^{+\infty} (x_1)_{i_1} \dots (x_n)_{i_n} p_X((x_1)_{i_1}, \dots, (x_n)_{i_n})$$
(1.12)

Si osserva facilmente che l'operatore *attesa* è lineare, ovvero che  $E(\alpha X + \beta Y) = \alpha E(X) + \beta E(Y), \forall \alpha, \beta \in \mathbb{R}$ , e per ogni coppia di variabili aleatorie X e Y.

PROPOSIZIONE 1.18. Siano X una variabile aleatoria con distribuzione  $p_X \ e \ \Phi : \mathbb{R}^n \longrightarrow \mathbb{R}$  una funzione. Posta  $U = \Phi(X)$ , se U ha media finita, allora

$$E(U) = E(\Phi(X)) = \sum_{i=1}^{+\infty} \Phi(x_i) p_X(x_i)$$
 (1.13)

DIMOSTRAZIONE. Siano  $\{u_i\}_{i\in\mathbb{N}}$  i valori assunti da U. Indichiamo con  $\mathcal{R}(u_i)$  la controimmagine di  $u_i$  tramite  $\Phi$ , ovvero

$$\mathcal{R}(u_i) = \Phi^{-1}(u_i) = \{ x_j \in X(\Omega) \mid u_i = \Phi(x_j) \}.$$

Allora, per ogni  $u_i$  si ha  $\{U = u_i\} = \bigcup_{x_j \in \mathcal{R}(u_i)} \{X = x_j\}$  e dunque  $\mathbf{P}(U = u_i) = \sum_{x_j \in \mathcal{R}(u_i)} \mathbf{P}(X = x_j) = p_U(u_i)$ , essendo l'unione disgiunta. Avendo supposto finita la media di U, ne segue che

$$E(U) = \sum_{i=1}^{+\infty} u_i p_U(u_i) = \sum_{i=1}^{+\infty} u_i \sum_{x_j \in \mathcal{R}(u_i)} p_X(x_j)$$
$$= \sum_{i=1}^{+\infty} \sum_{x_j \in \mathcal{R}(u_i)} u_i p_X(x_j) = \sum_{i=1}^{+\infty} \sum_{x_j \in \mathcal{R}(u_i)} \Phi(x_j) p_X(x_j)$$
$$= \sum_{i=1}^{+\infty} \Phi(x_i) p_X(x_i)$$

OSSERVAZIONE 1.19. Ricalcando i passaggi della dimostrazione precedente si può osservare che, se  $U = \Phi(X)$ , la media di U è finita se e solo se la serie  $\sum_{i\geq 0} \Phi(x_i)p_X(x_i)$  è assolutamente convergente.

OSSERVAZIONE 1.20. Se due variabili aleatorie U e V sono indipendenti ed a media finita, allora la variabile X = UV ha media finita e si ha

$$E(X) = E(UV) = E(U) E(V)$$
 (1.14)

DIMOSTRAZIONE. Sia  $\Phi : \mathbb{R}^2 \to \mathbb{R}$  definita da  $\Phi(a, b) = ab$ , per la proposizione 1.18 e la definizione di indipendenza si ha

$$E(X) = E\left(\Phi(U, V)\right) = \sum_{i,j} u_i v_j p_{(U,V)}(u_i, v_j)$$
$$= \sum_i u_i p_U(u_i) \sum_j v_j p_V(v_j) = E(U)E(V)$$

DEFINIZIONE 1.21 (Varianza). Sia X una variabile aleatoria discreta con media finita. L'operatore definito come segue

$$\operatorname{var}(X) = \sigma^{2}(X) = E\left[(X - E(X))^{2}\right]$$
 (1.15)

si chiama varianza della variabile X. La radice quadrata della varianza di X prende il nome di deviazione standard e si indica con  $\sigma(X)$ .

PROPOSIZIONE 1.22. Sia X una variabile aleatoria con media finita. Risulta

$$\operatorname{var}(X) = E(X^2) - E(X)^2$$
 (1.16)

DIMOSTRAZIONE. Sviluppando il quadrato nella definizione 1.21 di varianza e ricordando che l'attesa è un operatore lineare, si ha

$$var(X) = E \left[ X^{2} + E(X)^{2} - 2XE(X) \right]$$
  
=  $E(X^{2}) + E(X)^{2} - 2E(X)E(X) = E(X^{2}) - E(X)^{2}$ 

DEFINIZIONE 1.23 (**Covarianza**). Siano  $X \in Y$  due variabili aleatorie con media finita. Si definisce *covarianza* del vettore aleatorio (X, Y)la quantità

$$cov(X,Y) := E(X)E(Y) - E(XY).$$
 (1.17)

Due variabili aleatorie X, Y si dicono *scorrelate* se risulta

$$\operatorname{cov}(X,Y) = 0.$$

OSSERVAZIONE 1.24. Per ogni coppia di variabili aleatorie (X, Y) scorrelate, risulta

$$\operatorname{var}(X+Y) = \operatorname{var}(X) + \operatorname{var}(Y)$$

DIMOSTRAZIONE. Per la (1.17) si ha E(XY) = E(X)E(Y) - cov(X, Y) = E(X)E(Y). Ne segue che

$$\operatorname{var}(X+Y) = E(X^{2} + Y^{2} + 2XY) - (E(X) + E(Y))^{2}$$
  
=  $\operatorname{var}(X) + \operatorname{var}(Y) + 2E(XY) - 2E(X)E(Y) = \operatorname{var}(X) + \operatorname{var}(Y)$ 

OSSERVAZIONE 1.25. Se due variabili aleatorie X, Y sono indipendenti, allora sono scorrelate.

DIMOSTRAZIONE. Dalla definizione di indipendenza e di media, risulta

$$\operatorname{cov}(X,Y) = \left(\sum_{i} x_{i} p_{X}(x_{i})\right) \left(\sum_{i} y_{i} p_{Y}(y_{i})\right) - \left(\sum_{i,j} x_{i} y_{j} p_{(X,Y)}(x_{i},y_{j})\right)$$
$$= \left(\sum_{i} x_{i} p_{X}(x_{i})\right) \left(\sum_{i} y_{i} p_{Y}(y_{i})\right) - \left(\sum_{i,j} x_{i} p_{X}(x_{i}) y_{j} p_{Y}(y_{j})\right) = 0$$

Si noti che, in generale, l'implicazione inversa è falsa.

Concludiamo il paragrafo con una disugualianza, di cui si fa spesso uso quando si opera con quanità casuali. Per provarne la veridicità faremo uso del seguente LEMMA 1.26. Siano U, V due variabili aleatorie con media finita. Se  $U \ge V$  allora  $E(U) \ge E(V)$ .

DIMOSTRAZIONE. Si ponga X = U - V, per ipotesi  $X \ge 0$  pertanto  $\mathbf{P}(X \ge 0) = 1$  e l'immagine di X è fatta di soli valori positivi. Ne segue che

$$E(X) = E(U) - E(V) = \sum_{x_i \in X(\Omega)} x_i p_X(x_i) > 0$$

PROPOSIZIONE 1.27 (Disugualianza di Tschebyscheff). Sia X una variabile aleatoria con media e varianza finite. Per ogni  $\varepsilon > 0$  risulta

$$\mathbf{P}\left(|X - E(X)| > \varepsilon\right) \leqslant \frac{\sigma^2(X)}{\varepsilon^2} \tag{1.18}$$

DIMOSTRAZIONE. Introduciamo la variabile aleatoria

$$U = \varepsilon^2 \, \chi_{\{|X - E(X)| > \varepsilon\}} \,,$$

che vale costantemente  $\varepsilon^2$  se  $|X - E(X)| > \varepsilon$  e 0 altrimenti. Osserviamo che  $(X - E(X))^2 \ge U$  poiché se  $|X - E(X)| > \varepsilon$  allora  $U = \varepsilon^2$  e vale con il maggiore stretto, se  $|X - E(X)| \le \varepsilon$  allora U = 0 mentre  $(X - E(X))^2 \ge 0$ . Pertanto, per il lemma 1.26,

$$\sigma^{2}(X) = E\left((X - E(X))^{2}\right) \ge E(U)$$
$$= E\left(\varepsilon^{2}\chi_{\{|X - E(X)| > \varepsilon\}}\right) = \varepsilon^{2}\mathbf{P}\left(|X - E(X)| > \varepsilon\right)$$

**1.3. Variabili aleatorie continue.** Molto di frequente si ha a che fare con processi aleatori che possono assumere più che una infinità numerabile di valori. Questo è il caso di variabili aleatorie continue, di cui ci occuperemo in questa sottosezione.

Nella definizione 1.11 abbiamo osservato che per ogni variabile aleatoria X ed ogni  $x \in \mathbb{R}$ , è ben definita la probabilità che si verifichi l'evento  $\{X \leq x\}$ , probabilità che abbiamo indicato con  $F_X(x)$ , funzione di ripartizione di X. A partire da questa si può definire la densità di distribuzione di X, nel modo seguente DEFINIZIONE 1.28 (**Densità di distribuzione**). Sia X una variabile aleatoria sullo spazio  $(\Omega, \mathcal{A}, \mathbf{P})$ . Sia  $F_X(x)$  la sua funzione di ripartizione. La funzione  $p_X : \mathbb{R} \longrightarrow [0, 1]$  tale che

$$F_X(x) = \int_{-\infty}^x p_X(\xi) \, d\xi, \quad \forall x \in \mathbb{R}$$
(1.19)

è detta densità di distribuzione (o, più semplicemente distribuzione) della variabile X.

Da tale definizione si evince la proprietà seguente, che caratterizza una distribuzione continua di probabilità

PROPOSIZIONE 1.29. Sia X una variabile aleatoria e sia  $p_X$  la sua densità di distribuzione. Allora

$$\int_{\mathbb{R}} p_X(x) \, dx = 1 \tag{1.20}$$

DIMOSTRAZIONE. Mostrare la tesi equivale a mostrare che

$$\lim_{x \to +\infty} F_X(x) = 1.$$

Sia  $\{\xi_n\}_n \subset \mathbb{R}$  una successione non decrescente tale che  $\xi_n \to +\infty$  per *n* che diverge. Consideriamo l'insieme

$$\Delta = \bigcup_{n \in \mathbb{N}} \Delta_n = \bigcup_{n \in \mathbb{N}} \{ \omega : X(\omega) \leq \xi_n \}$$

Mostriamo che la probabilità  $\mathbf{P}(\Delta)$  è pari ad 1. L'inclusione  $\Delta \subseteq \Omega$  è vera per definizione, viceversa supponiamo che esista  $\omega \in \Omega$  tale che  $\omega \notin \Delta$  allora si avrebbe  $X(\omega) \ge \xi_n$  per ogni  $n \in \mathbb{N}$ . D'altra parte per come è definita  $\{\xi_n\}_n$ , per ogni  $\mu \in \mathbb{R}$  esiste un intero  $n_{\mu}$  tale che, per ogni  $n \ge n_{\mu}$  si ha  $\xi_n > \mu$ . Ne segue che per ogni  $\mu \in \mathbb{R}$ dovrebbe essere  $X(\omega) > \mu$  e questo è un assurdo, essendo  $X(\omega)$  un valore fissato, immagine di  $\omega$  tramite X. Pertanto  $\Omega \subseteq \Delta$  e dunque  $\Omega = \Delta$ ,  $\mathbf{P}(\Delta) = \mathbf{P}(\Omega) = 1$ . Dunque, per la proposizione 1.5, risulta

$$\lim_{x \to \infty} F_X(x) = \lim_{n \to \infty} \mathbf{P} \left( X \leqslant \xi_n \right) = \mathbf{P} \left( \bigcup_{n \in \mathbb{N}} \{ X \leqslant \xi_n \} \right) = \mathbf{P} \left( \Delta \right) = 1$$

Si noti che la prima ugualianza è vera perché  $F_X$  è una funzione non decrescente.

Definita la distribuzione di una variabile aleatoria continua, possiamo facilmente estendere le definizioni di *attesa*, *varianza*, *covarianza* date nel caso di variabili aleatorie discrete. DEFINIZIONE 1.30. Sia X una variabile aleatoriea con distribuzione  $p_X$ . Se  $(x p_X(x)) \in L^1(\mathbb{R})$  si dice che la variabile X ha media finita e la quantità

$$E(X) := \int_{\mathbb{R}} x \, p_X(x) \, dx \tag{1.21}$$

è detta attesa o media di X.

OSSERVAZIONE 1.31. Con questa definizione restano vere tutte le proprietà che abbiamo osservato per il caso discreto, in particolare se  $\Phi$ :  $\mathbb{R} \longrightarrow \mathbb{R}$  è una funzione e X una variabile aleatoria, posta  $U = \Phi(X)$ , risulta

$$E(U) = \int_{\mathbb{R}} \Phi(x) p_X(x) \, dx. \tag{1.22}$$

DEFINIZIONE 1.32 (**Densità marginale**). Sia  $X = (X_0, \ldots, X_n)$  un vettore aleatorio con densità di distribuzione congiunta p. La quantità

$$p_{X_i}(x) = \int_{\mathbb{R}^n} p(x_0, \dots, x_n) \, dx_0, \dots, dx_{i-1}, dx_{i+1}, \dots, dx_n$$

si dice densità marginale della variabile  $X_i$ .

Si noti che tale definizione ha senso anche nel caso di variabili (vettori) aleatorie diescrete, con la sola differenza che in luogo dell'integrale si pone la sommatoria (serie) sull'insieme dei valori assunti dalle n variabili integrande.

DEFINIZIONE 1.33 (**Densità condizionale**). Siano X, Y due variabili aleatorie con densità (marginale)  $p_X e p_Y$ , rispettivamente. Sia  $p_{(X,Y)}$ la densità congiunta del vettore aleatorio (X, Y). Dalla definizione 1.7 segue che le densità delle variabili aleatorie X|Y e Y|X sono date da

$$p_{X|Y}(x) = \frac{p_{(X,Y)}(x,y)}{p_Y(y)}, \quad p_{Y|X}(y) = \frac{p_{(X,Y)}(x,y)}{p_X(x)}$$

ed si dicono, rispettivamente, densità condizionale di X data Y e di Y data X.

OSSERVAZIONE 1.34. La definizione appena data è ben posta. Infatti certamente  $p_{X|Y}(x) \ge 0$ ; inoltre

$$\int_{\mathbb{R}} p_{X|Y}(x) \, dx = \frac{1}{p_Y(y)} \int_{\mathbb{R}} p_{(X,Y)}(x,y) \, dx = \frac{p_Y(y)}{p_Y(y)} = 1$$

ed analogamente per  $p_{Y|X}(y)$ .

#### 2. IL PROBLEMA DELL'INTEGRAZIONE

#### 2. Il problema dell'integrazione

In questa sezione presentiamo il metodo di integrazione numerica basato sulla tecnica di *Monte Carlo*. L'idea è quella di costruire una approssimazione numerica, entro una certa precisione prefissata, dell'integrale di una funzione f su un dominio di integrazione  $\mathcal{Q} \subset \mathbb{R}^d$ , limitato. Tale tecnica, a differenza delle formule di quadratura deterministiche, si basa su scelte probabilistiche dei *nodi* sui quali "campionare" f per costruire l'approssimazione voluta. Questo tipo di approccio, come vedremo, assicura un notevole vantaggio computazionale quando la dimensione d > 0 del dominio della funzione da integrare cresce.

DEFINIZIONE 2.1. Sia  $\mathcal{Q} \subset \mathbb{R}^d$  un insieme limitato. Data una funzione  $f : \mathbb{R}^d \longrightarrow \mathbb{R}$  tale che  $f \in L^1(\mathcal{Q})$ , indichiamo con  $\mathcal{J}$  la quantià

$$\mathcal{J} = \mathcal{J}(f, \mathcal{Q}) = \int_{\mathcal{Q}} f(\mathbf{x}) \, d\mathbf{x}$$
(2.1)

dove  $\mathbf{x} = (x_1, \ldots, x_d) \in \mathcal{Q}$ .

Vogliamo produrre una formula che approssimi  $\mathcal{J}$ . Supponiamo di considerare *n* vettori aleatori  $X_1, \ldots, X_n$  *d*-dimensionali, ovvero  $X_i = (X_i^{(1)}, \ldots, X_i^{(d)})$  per  $i = 1, \ldots, n$ , con stessa legge di distribuzione congiunta  $p = p_{X_1} = p_{X_n}$  ed indipendenti (*indipendenti identicamente distribuiti*). È ben definita la variabile aleatoria

$$\langle \mathcal{J} \rangle_n = \langle \mathcal{J}(f, \mathcal{Q}) \rangle_n := \frac{1}{n} \sum_{i=1}^n \frac{f(X_i)}{p(X_i)} \chi_{\mathcal{Q}}(X_i)$$
 (2.2)

dove indichiamo con  $\chi_{\mathcal{Q}}$  la funzione caratteristica dell'insieme  $\mathcal{Q}$ , tale che  $\chi_{\mathcal{Q}}(\mathbf{y}) = 1$  se  $\mathbf{y} \in \mathcal{Q}$ , mentre vale zero altrimenti.

Tale variabile aleatoria è detta *estimatore* di  $\mathcal{J}$ .

PROPOSIZIONE 2.2. Siano  $f : \mathbb{R}^d \longrightarrow \mathbb{R} \ e \ \mathcal{Q} \subset \mathbb{R}^d$  un insieme limitato. Per ogni intero n > 0 si ha

$$E\left(\left\langle \mathcal{J}\right\rangle_{n}\right) = \mathcal{J} \tag{2.3}$$

DIMOSTRAZIONE. Per la proposizione 1.16, essendo i vettori  $\{X_i\}_{i=1,...,n}$ indipendenti identicamente distribuiti, anche le  $\left\{\frac{f(X_i)}{p(X_i)}\chi_I(X_i)\right\}_{i=1,...,n}$ lo sono. Allora, dalla Proposizione 1.18 e dalla linearità dell'operatore E, segue che

$$E\left(\langle \mathcal{J} \rangle_n\right) = \frac{1}{n} \sum_{i=1}^n E\left(\frac{f\left(X_i\right)}{p\left(X_i\right)} \chi_{\mathcal{Q}}(X_i)\right)$$
$$= \frac{1}{n} \sum_{i=1}^n \int_{\mathbb{R}^d} \frac{f\left(\mathbf{x}\right)}{p\left(\mathbf{x}\right)} \chi_{\mathcal{Q}}(\mathbf{x}) p(\mathbf{x}) \, d\mathbf{x} = \int_{\mathcal{Q}} f(\mathbf{x}) \, d\mathbf{x} = \mathcal{J}$$

Abbiamo pertanto prodotto una famiglia di variabili casuali  $\{\langle \mathcal{J} \rangle_n\}_n$ il cui valore atteso, indipendentemente da *n* e dalla dimensione *d* dello spazio su cui *f* agisce, coincide con l'integrale che volgiamo calcolare.

Essendo le  $\langle \mathcal{J} \rangle_n$  aleatorie, anche l'errore che commettiamo nell'approssimare  $\mathcal{J}$  facendo uso di esse, è aleatorio. Indichiamo con

$$\rho_n := \mathcal{J} - \langle \mathcal{J} \rangle_n \tag{2.4}$$

la variabile aleatoria dell'errore. Affinché la scelta di approssimare l'integrale con gli estimatori considerati sia una buona scelta, è necessario che  $\rho_n$  assuma, in media, valori piccoli. Osserviamo che vale la seguente

PROPOSIZIONE 2.3. Siano  $\mathcal{J}, \langle \mathcal{J} \rangle_n$  e  $\rho_n$  come precedentemente definiti. Per ogni n > 0 si ha

$$E(\rho_n^2) = \frac{1}{n} \sigma^2 \left( \frac{f(X)}{p(X)} \chi_{\mathcal{Q}}(X) \right) = \frac{1}{n} \int_{\mathcal{Q}} \left( \frac{f(\mathbf{x})}{p(\mathbf{x})} - \mathcal{J} \right)^2 p(\mathbf{x}) d\mathbf{x} \quad (2.5)$$

dove X è uno qualsiasi degli n vettori aleatori  $X_i$  considerati e p è la sua densità di distribuzione congiunta.

DIMOSTRAZIONE. Notiamo anzitutto che, per le osservazioni 1.24 e 1.25, se  $U_1, \ldots, U_m$  sono delle variabili aleatorie indipendenti con stessa legge di distribuzione allora

$$\sigma^2\left(\sum_{i=1}^m U_i\right) = \sum_{i=1}^m \sigma^2\left(U_i\right) = m\,\sigma^2(U)$$

essendo  $\sigma^2(U) = \sigma^2(U_i)$  per ogni  $i = 1, \ldots, m$ . Allora, indicata con  $\Phi(U) := \frac{f(U)}{p(U)} \chi_Q(U)$ , si ha

$$E(\rho_n^2) = \sigma^2 \left( \langle \mathcal{J} \rangle_n \right)$$
$$= \sigma^2 \left( \frac{1}{n} \sum_{i=1}^n \Phi(X_i) \right) = \frac{1}{n^2} \sum_{i=1}^n \sigma^2(\Phi(X_i)) = \frac{1}{n} \sigma^2(\Phi(X))$$

essendo  $\Phi(X) = \Phi(X_i)$  per ogni i = 1, ..., n.

La seconda ugualianza è un immediata conseguenza della definizione di varianza e della relazione  $E(\Phi(X)) = \mathcal{J}$ .

OSSERVAZIONE 2.4. Dalla proposizione precedente segue il segue che, indipendentemente dalla dimensione dello spazio  $\mathbb{R}^d$  ed indipendentemente dalla regolarità della funzione f,

$$\lim_{n \to +\infty} E\left(\rho_n^2\right) = \lim_{n \to +\infty} \mathcal{O}\left(n^{-1}\right) = 0$$
(2.6)

poiché abbiamo assunto  $f \in L^1(\mathcal{Q})$  e  $0 \leq \int_{\mathcal{Q}} p(\mathbf{y}) d\mathbf{y} < 1$  essendo p una distribuzione di probabilità congiunta su  $\mathbb{R}^d$ .

Pertanto gli estimatori  $\langle \mathcal{J} \rangle_n$  sono, al crescere del numero *n* dei nodi, scelti casualmente su  $\mathbb{R}^d$ , delle buone approssimazioni dell'integrale  $\mathcal{J}$ .

OSSERVAZIONE 2.5. La disuguaglianza di Tschebyscheff consente di dare una stima ulteriore sull'accuratezza della scelta degli estimatori  $\langle \mathcal{J} \rangle_n$ , infatti segue da (1.18) che

$$\mathbf{P}\left(\left|\left\langle \mathcal{J}\right\rangle_{n}-\mathcal{J}\right|>\varepsilon\right)\leqslant\frac{\sigma^{2}\left(\left\langle \mathcal{J}\right\rangle_{n}\right)}{\varepsilon^{2}}.$$
(2.7)

Se indichiamo con  $\Phi(U) = \frac{f(U)}{p(U)}\chi_{\mathcal{Q}}(U)$ , per la Proposizione 2.3 si ha  $\sigma^2(\langle \mathcal{J} \rangle_n) = E(\rho_n^2) = \frac{1}{n}\sigma^2(\Phi(X))$ , dove X è una delle *n* variabili aleatorie indipendenti identicamente distribuite che definiscono  $\langle \mathcal{J} \rangle_n$ . Pertanto, fissata una tolleranza  $\tau > 0$ , per ogni  $\varepsilon > 0$ , scelto  $n(\tau) \ge \frac{1}{\tau \varepsilon^2} \sigma^2(\Phi(X))$  si ha

$$\mathbf{P}\left(\left|\left\langle \mathcal{J}\right\rangle_{n(\tau)} - \mathcal{J}\right| > \varepsilon\right) \leqslant \tau \tag{2.8}$$

e dunque con scelte opportune del numero dei nodi n si può ridurre arbitrariamente la probabilità che l'estimatore differisca dall'integrale effettivo.

D'altra parte la quantità  $n(\tau)$  non è in generale calcolabile, perchè richiede di conoscere  $\sigma^2(\Phi(X))$ , ovvero di sapere il valore dell'integrale  $\mathcal{J}$ . In taluni casi è possibile dare delle stime per il valore di  $\sigma^2(\Phi(X))$ , ad esempio costruendo a sua volta un estimatore  $\langle \mathcal{I} \rangle_m$  che approssimi l'integrale  $\mathcal{I} = \sigma^2(\Phi(X))$ .

ESEMPIO 2.6. Vogliamo calcolare l'irradianza dovuta ad una fonte di luce  $\mathbf{x}$ . Dobbiamo calcolare l'integrale

$$\mathcal{J} = \int_{\Omega_{\mathbf{x}}} L(\mathbf{x} \to \Theta) \cos(N_{\mathbf{x}}, \Theta) \, d\omega(\Theta).$$
 (2.9)

Poichè limitiamo il nostro interesse alla sola luce emessa da  $\mathbf{x}$ , indichiamo con  $L_{\mathbf{x}}(\Theta) = L(\mathbf{x} \to \Theta)$  la luce emessa dal diffusore. L'integrale è esteso sul dominio

$$\Omega_{\mathbf{x}} = \left\{ \Theta \in \mathbb{R}^3 : \|\mathbf{x} - \Theta\| = 1, \, \langle N_{\mathbf{x}}, \Theta \rangle \ge 0 \right\}$$

ovvero l'emisfero "sovrastante"  $\mathbf{x}$ , di raggio unitario. Il consueto cambio di coordinate in funzione dei due parametri angolari di Eulero,  $\theta \in \varphi$ , permette di riscrivere l'integrale  $\mathcal{J}$  nel modo seguente

$$\mathcal{J} = \int_0^{2\pi} \int_0^{\pi/2} f(\theta, \varphi) \, d\varphi \, d\theta = \int_0^{2\pi} \left\{ \int_0^{\pi/2} L_{\mathbf{x}}(\theta, \varphi) \cos \varphi \sin \varphi \, d\varphi \right\} d\theta.$$

Consideriamo uno spazio di probabilità  $(\Omega, \mathcal{A}, \mathbf{P})$ , scegliamo *n* vettori aleatori  $(\theta_1, \varphi_1), \ldots, (\theta_n, \varphi_n)$  indipendenti identicamente distribuiti con densità di distribuzione congiunta *p*, tali che  $(\theta_i, \varphi_i) : \Omega \times \Omega \rightarrow$  $[0, 2\pi] \times [0, \pi/2]$ . Costruiamo l'estimatore per  $\mathcal{J}$ ,

$$\langle \mathcal{J} \rangle_n = \frac{1}{n} \sum_{i=1}^n \frac{f(\theta_i, \varphi_i)}{p(\theta_i, \varphi_i)} = \frac{1}{n} \sum_{i=1}^n \frac{L_{\mathbf{x}}(\theta_i, \varphi_i) \cos \varphi_i \sin \varphi_i}{p(\theta_i, \varphi_i)}$$

Resta da scegliere con che tipo di distribuzione generare i vettori  $(\theta_i, \varphi_i)$ . Una possibilità , che semplifica la complessità di calcolo dell'estimatore, è generare casualmente valori con distribuzione

$$p(u,v) = \frac{1}{\pi} \cos v \sin v,$$

(la presenza del fattore  $1/\pi$  assicura che  $\int_0^{2\pi} \int_0^{\pi/2} p(u, v) du dv = 1$ ), con la quale si ottiene

$$\langle \mathcal{J} \rangle_n = \frac{\pi}{n} \sum_{i=1}^n L_{\mathbf{x}}(\theta_i, \varphi_i).$$
 (2.10)

Procediamo con la nostra stima, senza preoccuparci, per ora, di come sia possibile generare campioni con una tale distribuzione. Affronteremo il problema più in là, nel paragrafo 4.

Poniamo  $\Phi = \Phi(\mathbf{y}) = \frac{f(\mathbf{y})}{p(\mathbf{y})} \chi_{\Omega_{\mathbf{x}}}(\mathbf{y})$ , ne segue che

$$\sigma^{2}(\Phi) \leqslant E(\Phi^{2}) = \int_{\Omega_{\mathbf{x}}} \frac{f^{2}(\theta,\varphi)}{p(\theta,\varphi)} \, d\theta \, d\varphi \leqslant \pi \, \|L_{\mathbf{x}}\|_{L^{2}(\Omega_{\mathbf{x}})}^{2} \, ,$$

allora per la (2.8), scegliere  $n(\alpha, \beta) \ge \pi \|L_{\mathbf{x}}\|_{L^2(\Omega_{\mathbf{x}})}^2 10^{2\alpha+\beta}$  è una condizione sufficiente affinché

$$\mathbf{P}\left(\left|\langle \mathcal{J} \rangle_{n(\alpha,\beta)} - \mathcal{J}\right| > 10^{-\alpha}\right) \leqslant 10^{-\beta}$$

per ogni  $\alpha > 0, \beta > 0$ 

**2.1. Formule di quadratura deterministiche.** L'importanza dei metodi di integrazione probabilistici presentati è evidente se paragonati alle formule di quadratura deterministiche. La costruzione di una formula che approssimi un integrale numericamente può avvenire seguendo due strade distinte: una possibilità è quella di scegliere in maniera "intelligente" i nodi di integrazione all'interno del dominio in modo da rendere minimo l'errore numerico della formula, fissato il numero di nodi n all'interno del dominio di integrazione; una seconda possibilità, computazionalmente spesso più agevole, è quella di minimizzare l'errore aumentando il numero di nodi di integrazione su cui basare la formula. Le formule della prima classe vengono dette formule di quadratura Gaussiane, le seconde formule di quadratura Newtoniane. Presentiamo brevemente, a titolo comparativo, le due tipoligie di metodi nel caso unidmensionale, benché il ventaggio dell'utilizzo dei metodi di Monte Carlo piuttosto che metodi deterministici sia marcatamente più evidente quando si vogliono approssimare integrali su domini ddimensionali con d > 1.

Siano a < b due valori reali ed  $f \in L^1([a, b])$  una funzione da [a, b]in  $\mathbb{R}$ , supponiamo di voler spprossimare il valore dell'integrale

$$I = \int_{a}^{b} f(x) dx \tag{2.11}$$

tramite una formula di quadratura del tipo

$$S_n(f) = \sum_{i=0}^n \omega_i f(x_i).$$
 (2.12)

I punti  $x_i \in \mathbb{R}$  per  $i = 0, \ldots, n$  si dicono nodi della formula di quadratura, i valori  $\omega_i \in \mathbb{R}$  per  $i = 0, \ldots, n$ , pesi della formula.

DEFINIZIONE 2.7 (**Ordine di una formula di quadratura**). Indicato con  $r_n(f) = I - S_n(f)$ , una formula del tipo (2.12) si dice di ordine m se  $r_n(p) = 0$  per ogni polinomio p di grado minore o uguale ad m e  $r_n(p) \neq 0$  se deg(p) > m.

Si può dimostrare che maggiore è l'ordine di una formula di quadratura, migliore è l'approssimazione che questa fornisce.

TEOREMA 2.8 (Formula di quadratura di Gauss). Sia  $p_{n+1}$  il polinomio di grado minore uguale ad n+1 ortogonale ad ogni polinomio di grado minore di n+1 in [a,b]. Allora  $p_{n+1}$  esiste, è unico ed ha esattamente n+1 radici reali distinte  $\xi_0, \ldots, \xi_n$  nell'intervallo [a,b]. Inoltre, la formula

$$\mathcal{G}_{n}(f) = \sum_{i=0}^{n} \omega_{i} f(\xi_{i}), \quad \omega_{i} = \int_{a}^{b} \prod_{j \neq i} \frac{(x - \xi_{j})}{(\xi_{i} - \xi_{i})} dx$$
(2.13)

ha ordine 2n + 1.

PROPOSIZIONE 2.9. L'ordine massimo di una formula del tipo (2.12)è 2n + 1.

DIMOSTRAZIONE. Se poniamo  $\pi(x) = \prod_{i=0}^{n} (x - x_i)^2$  dove gli  $x_i$  sono i nodi di (2.12), allora deg $(\pi) = 2n + 2$  e si ha  $S_n(\pi) = 0$ , mentre  $\int_a^b \pi(x) dx > 0$ , pertanto  $r_n(\pi) \neq 0$  e per il teorema 2.8 si ha la tesi.

Ciò che rende spesso poco agevole l'utilizzo delle formule di Gauss è il calcolo del polinomio  $p_{n+1}$ , delle sue radici  $\xi_i$  e dei pesi  $\omega_i$ , che da esse dipendono.

Un approccio computazionalmente più semplice è quello delle formule di Newton-Cotes, che si basano sull'idea di suddividere il dominio di integrazione [a, b] in m sottointervalli  $[\nu_k, \nu_{k+1}], k = 0, \ldots, m-1$ tali che  $\nu_0 = a, \nu_m = b$  ed approssimare la funzione f in ognuno di tali intervalli con il polinomio  $p_{n,k}$ , di grado n, interpolante f in  $x_0^{(k)}, \ldots, x_n^{(k)}$ , nodi distinti in  $[\nu_k, \nu_{k+1}]$ . Possiamo scrivere  $p_{n,k}$  nella forma di Lagrange, ovvero

$$p_{n,k}(x) = \sum_{i=0}^{n} f\left(x_i^{(k)}\right) \ell_{i,n}^{(k)}(x), \quad \ell_{i,n}^{(k)}(x) = \prod_{j \neq i} \frac{x - x_j^{(k)}}{x_i^{(k)} - x_j^{(k)}}.$$
 (2.14)

Dalla linearità dell'integrale segue che

$$I = \int_{a}^{b} f(x) \, dx = \sum_{k=0}^{m-1} \int_{\nu_{k}}^{\nu_{k+1}} f(x) \, dx, \qquad (2.15)$$

la formula interpolatoria di quadratura di Newton–Cotes è data da

$$\mathcal{N}_{n}(f) = \sum_{k=0}^{m-1} \int_{\nu_{k}}^{\nu_{k+1}} p_{n,k}(x) \, dx = \sum_{i=0}^{n} \left( \sum_{k=0}^{m-1} \omega_{ik} \right) f\left(x_{i}^{(k)}\right),$$
$$\omega_{ik} = \int_{\nu_{k}}^{\nu_{k+1}} \ell_{i,n}^{(k)}(x) \, dx \quad (2.16)$$

I casi n = 1 ed n = 2 sono anche noti come formula dei trapezi e formula di Cavalieri-Simpson, rispettivamente. Il primo corrisponde ad approssimare l'area del sottografico di f in [a, b] con dei trapezi passanti per i vari punti dei nodi, il secondo con degli pseudo-rettangoli dove una delle due basi è sostituita da una parabola che approssima f.



FIGURA 1. Metodi di quadratura deterministica più comuni

In generale questo tipo di metodi danno luogo ad un errore  $e_n(f) = I - N_n(f)$  che decresce come  $H^n$  dove  $H = \max_{k=0,\dots,m-1} \{|\nu_{k+1} - \nu_k|\}$ . Se volessimo applicare queste formule ad un integrale esteso su un dominio  $\mathcal{Q} \subset \mathbb{R}^d$  con d > 1, nel caso delle formule di Gauss si andrebbe in contro a problemi di complessità troppo elevata, nel caso di Newton-Cotes dovremmo suddividere  $\mathcal{Q}$  in m sotto intervalli  $S_i \subset \mathbb{R}^d$  (ammesso che questo sia possibile) e su di ognuno interpolare f con polinomi  $\pi : \mathbb{R}^d \to \mathbb{R}$ , il che richiederebbe, per un polinomio di grado n, un numero di nodi dell'ordine di  $n^d$  (ad esempio per d = 2 sono necessari  $2^{-1}(n+1)(n+2)$  nodi). Questo rende immediatamente evidenti i vantaggi dell'integrazione probabilistica su quella deterministica quando d > 1. Come è già stato fatto notare, integrare con metodi di Monte Carlo garantisce gli stessi risultati di convergenza indipendentemente dalla dimensione d dello spazio e dalla regolarità del dominio  $\mathcal{Q}$  di integrazione.

#### 3. Riduzione dell'errore

Quando ci si appresta ad approssimare un integrale con il metodo di Mote Carlo ci si può trovare in due situazioni: il caso in cui non si hanno informazioni sulla funzione f da integrare ed il caso in cui, invece, si conosce in parte o totalmente f. Nel primo caso in generale si tende ad utilizzare variabili aleatorie uniformamente distribuite per la scelta dei campioni, mentre nel secondo caso si possono fare delle scelte function based per la distribuzione dei campioni, in modo da ridurre la varianza dell'estimatore  $\langle \mathcal{J} \rangle_n$ . In questa sezione ci occuperemo dapprima del secondo caso, poi del primo caso dando delle stime per la varianza che questo comporta.

**3.1.** Moltiplicatori di Lagrange. Quella dei moltiplicatori di Lagrange è una tecnica analitica per la ricerca dei punti estremali vincolati di una funzione f. Ci limitiamo, per semplicità , al caso di  $\mathbb{R}^2$ . Sia  $A \subset \mathbb{R}^2$  aperto e siano  $f : A \to \mathbb{R}, v : A \to \mathbb{R}$  due funzioni in  $\mathcal{C}^1(A)$  con  $\|\nabla v(x, y)\|^2 > 0$  per ogni  $(x, y) \in A$ . Indichiamo con W l'insieme degli zeri di v, detta condizione di *vincolo*, ovvero

$$W = \{(x, y) \in A : v(x, y) = 0\}.$$

Supponiamo che  $W \neq \emptyset$ , vogliamo cercare delle condizioni per i punti di massimo o minimo di f(x, y) quando  $(x, y) \in W$ .

Sia  $(x_0, y_0)$  un punto di W tale che si verifichino una delle due condizioni

$$\partial_x v(x_0, y_0) \neq 0, \quad \partial_y v(x_0, y_0) \neq 0$$

per il teorema della funzione implicita, l'equazione v(x, y) = 0 definisce localmente una funzione  $x = \alpha(y)$  oppure una  $y = \beta(x)$  tali che  $x_0 = \alpha(y_0)$  oppure  $y_0 = \beta(x_0)$ . Supponiamo che si abbia la seconda condizione (il ragionamento nel caso della prima condizione è del tutto analogo): allora il punto  $x_0$  è estremale per la funzione di una sola variabile  $f(x, \beta(x))$  e pertanto soddisfa la condizione  $f'(x_0) = \partial_x f(x_0) +$  $\partial_y f(x_0)\beta'(x_0) = 0$ . Inoltre, derivando la condizione  $v(x, \beta(x)) = 0$ , si ottiene  $\beta'(x) = -(\partial_y v(x, y))^{-1} \partial_x v(x, y)$  in un intorno in cui  $y = \beta(x)$ . Ne segue che

$$\partial_x f(x_0, y_0) - \partial_y f(x_0, y_0) \frac{\partial_x v(x_0, y_0)}{\partial_y v(x_0, y_0)} = 0.$$
(3.1)

Poniamo

$$\Psi(x, y, \lambda) = f(x, y) - \lambda v(x, y).$$
(3.2)

Se in  $(x_0, y_0 = \beta(x_0))$  si ha  $\partial_y f \neq 0$ , ne segue che esiste

$$\lambda_0 = (\partial_x v(x_0, y_0))^{-1} \partial_y v(x_0, y_0) = (\partial_x f(x_0, y_0))^{-1} \partial_y f(x_0, y_0)$$

soluzione del sistema

$$\nabla \Psi(x_0, y_0, \lambda_0) = 0. \qquad (3.3)$$

Pertanto la ricerca di un punto estremale  $(x_0, y_0)$  di f vincolato alla condizione v(x.y) = 0 si può ricondurre alla ricerca dei valori di  $\lambda \in \mathbb{R}$  (detti *moltiplicatori di Lagrange*),  $x \in y$ , che risolvono il problema di estremo non vincolato dato dal sistema di tre equazioni in tre incognite (3.3).

Benché non sia del tutto immediato, è possibile estendere questo ragionamento al caso di due funzionali  $F \in L^1(A)^* \in V \in L^1(A)^*$  ed ottenere un risultato analogo. Precisamente, se indichiamo con

$$\Lambda = \left\{ p \in L^1(A) : V(p) = 0 \right\},\$$

la ricerca di  $p \in \Lambda$  estremale per il funzionale F, può essere ricondotta alla ricerca delle soluzioni in  $L^1(A)$  del sistema

$$\nabla \Psi(p,\lambda) = 0 \iff \begin{cases} D_p F(p) - \lambda D_p V(p) = 0\\ V(p) = 0 \end{cases}$$
(3.4)

dove  $\lambda \in \mathbb{R}$ ,  $\Psi(p, \lambda) = F(p) - \lambda V(p)$  e la derivata rispetto a p si fa formalmente come se operasse su funzioni di variabile reale. Assumiamo veri questi risultati di analisi funzionale e vediamo come si applicano al problema della riduzione dell'errore di un estimatore di Monte Carlo  $\langle \mathcal{J} \rangle_n$ .

Per la proposizione 2.3 la media dell'errore dovuto all'approssimare  $\mathcal{J}$  con  $\langle \mathcal{J} \rangle_n$  è dato da  $E(\rho_n^2) = (1/n)$  var  $(\Phi(X))$ , dove indichiamo con  $\Phi(X) = f(X)p(X)^{-1}\chi_{\mathcal{Q}}(X)$ . Cerchiamo la distribuzione di probabilità p che minimizzi il funzionale  $E(\rho_n^2)$  con il vincolo  $V(p) = \int_{\mathcal{Q}} p(x) dx - 1$ . Il funzionale  $\Psi$  da minimizzare assume la forma

$$\Psi(p,\lambda) = \lambda + \frac{1}{n} \int_{\mathcal{Q}} \left\{ \left(\frac{f}{p} - \mathcal{J}\right)^2 p - \lambda p \right\}.$$
 (3.5)

La derivata formale di  $\Psi$  rispetto a p è nulla se è nulla la derivata della funzione integranda: pertanto la distribuzione p ottimale per l'estimatore  $\langle \mathcal{J} \rangle_n$  deve essere tale che

$$\partial_p \Psi(p,\lambda) = 0 \Leftrightarrow \frac{f^2}{p^2} + \lambda n - \mathcal{J}^2 = 0 \Leftrightarrow p = \frac{|f|}{\sqrt{\mathcal{J}^2 - \lambda n}}$$
(3.6)

Imponendo anche la condizione del vincolo, ovvero  $\partial_{\lambda}\Psi(p,\lambda) = 0$ , segue che  $\sqrt{\mathcal{J}^2 - \lambda n} = \int_{\mathcal{Q}} |f(x)| dx$ . Dunque la densità di distribuzione ottima per stimare l'integrale  $\mathcal{J}$  con il metodo di Monte Carlo è

$$p_X(x) = \frac{|f(x)|}{\int_{\mathcal{Q}} |f(x)| \, dx}$$
(3.7)

Questa scelta per la distribuzione degli n campioni dell'estimatore darebbe luogo ad un errore a media nulla (si avrebbe  $E(\rho_n^2) = 0$ ); tuttavia la formula ottenuta richiede essa stessa la conoscenza dell'integrale  $\mathcal{J}$ , ed è, pertanto, inutilizzabile. Non è possibile scegliere la distribuzione p ottima per l'estimatore di Monte Carlo. Quello che l'intuito ci suggerisce è che, in generale, la scelta di p per stimare l'integrale di f è tanto migliore quanto più le funzioni p ed f si "assomigliano", in un senso che può essere in parte chiarito osservando come è stata scelta la distribuzione congiunta  $p(\theta, \varphi)$  nell'Esempio 2.6.

**3.2.** Campionamento stratificato. Un altro approccio possibile per ridurre la varianza dell'estimatore è quello del *campionamento stratificato*. L'idea è di suddividere il dominio di integrazione in m sotto domini e forzare i campioni aleatori a stare almeno uno in ognuno di tali sotto domini. Operare in questo modo garantisce che la varianza dell'estimatore così ottenuto sia minore che quella dell'estimatore di Monte Carlo puro. Osserviamo questo fatto nel caso unidimensionale. Cosideriamo gli m + 1 punti  $a = \nu_0, \nu_1, \ldots, \nu_m = b$  e suddividiamo l'integrale  $\mathcal{J}$  nel modo seguente

$$\mathcal{J} = \int_{a}^{b} f(x) \, dx = \sum_{k=0}^{m-1} \int_{\nu_{k}}^{\nu_{k+1}} f(x) \, dx = \sum_{k=0}^{m-1} \mathcal{J}_{k} \,. \tag{3.8}$$

Per ogni  $k = 0, \ldots, m-1$  introduciamo n(k) variabili aleatorie  $X_1^{(k)}, \ldots, X_{n(k)}^{(k)}$  tali che  $X_i^{(k)} : \Omega \to [\nu_k, \nu_{k+1}]$  per  $i = 1, \ldots, n(k)$ , indipendenti identicamente distribuite, con densità  $p_k = p_{X_1^{(k)}} = \cdots = p_{X_{n(k)}^{(k)}}$ . Indichiamo con  $\chi_k = \chi_{[\nu_k, \nu_{k+1}]}$  la funzione caratteristica dell'intervallo  $[\nu_k, \nu_{k+1}]$  e costruiamo m estimatori

$$\langle \mathcal{J}_k \rangle_{n(k)} = \frac{1}{n(k)} \sum_{i=1}^{n(k)} \frac{f\left(X_i^{(k)}\right)}{p_k\left(X_i^{(k)}\right)} \chi_k\left(X_i^{(k)}\right), \quad k = 0, \dots, m-1 \quad (3.9)$$

L'estimatore complessivo di  $\mathcal{J}$  è dato da  $\langle \mathcal{J}^{(S)} \rangle = \sum_{k=0}^{m-1} \langle \mathcal{J}_k \rangle_{n(k)}$ . Se le variabili  $X_i^{(k)}$ ,  $k = 0, \ldots, m-1$ ,  $i = 1, \ldots, n(k)$  sono indipendenti, per l'osservazione 1.24 e la Proposizione 2.3, ne segue che

$$\operatorname{var}\left(\left\langle \mathcal{J}^{(S)}\right\rangle\right) = \sum_{k=0}^{m-1} \left(\frac{1}{n(k)} \int_{\nu_k}^{\nu_{k+1}} \frac{f(x)^2}{p_k(x)} dx - \left(\mathcal{J}_k\right)^2\right).$$

Per confrontare la varianza dell'estimatore stratificato e di quello classico assumiamo che le variabili  $X_i^{(k)}$  siano identicamente distribuite. Se indichiamo con n il numero totale di campioni (ovvero  $\sum_k n(k) = n$ ), e con  $\langle \mathcal{J} \rangle_n$  l'estimatore di Monte Carlo standard, pesato con la stessa distribuzione di probabilità delle variabili  $X_i^{(k)}$ , rinormalizzata opportunamente, risulta

$$\sum_{k=0}^{m-1} \frac{1}{n(k)} \left( \mathcal{J}_k \right)^2 \ge \frac{1}{n} \left( \sum_{k=0}^{m-1} \mathcal{J}_k \right)^2 = \frac{1}{n} \mathcal{J}^2,$$

da cui segue che

$$\operatorname{var}\left(\langle \mathcal{J}^{(S)} \rangle\right) - \operatorname{var}\left(\langle \mathcal{J} \rangle_{n}\right) \\ \leqslant \sum_{k=0}^{m-1} \left(\frac{1}{n(k)} \int_{\nu_{k}}^{\nu_{k+1}} \frac{f(x)^{2}}{p_{k}(x)} \, dx\right) - \frac{1}{n} \int_{a}^{b} \frac{f(x)^{2}}{p(x)} \, dx \\ = \int_{a}^{b} f(x)^{2} \sum_{k=0}^{m-1} \left(\frac{1}{n(k)p_{k}(x)} - \frac{1}{n p(x)}\right) \, dx \,. \quad (3.10)$$

Vogliamo una condizione sufficiente affinché la varianza si riduca facendo uso dell'estimatore stratificato. Imponiamo a tale scopo che la quantità (3.10) non sia positiva, il che equivale ad imporre

$$\frac{m}{n \, p(x)} \ge \sum_{k=0}^{m-1} \frac{1}{n(k) \, p_k(x)} \quad \forall x \in [a, b]$$
(3.11)

La condizione (3.11) va studiata caso per caso, non è di facile verifica in generale, tuttavia esiste almeno una scelta delle variabili  $X_i^{(k)}$  e dei campioni n(k) per cui è soddisfatta. Infatti, supponiamo che le variabili  $X_1^{(k)}, \ldots, X_{n(k)}^{(k)}$  siano uniformemente distribuite nei loro rispettivi intervalli  $[\nu_k, \nu_{k+1}]$  e che in ognuno di questi consideriamo un solo campione (ovvero n(k) = 1). Ne segue che m = n e che

$$\frac{m}{np(x)} = b - a = \sum_{k=0}^{n-1} \left(\nu_{k+1} - \nu_k\right) = \sum_{k=0}^{m-1} \frac{1}{n(k)p_k(x)}$$

ovvero nella (3.11) vale esattamente l'ugualianza. Tale scelta, dunque, mostra che l'utilizzo dell'estimatore stratificato garantisce una stima migliore dell'estimatore classico (comporta un errore medio minore), indipendentemente dalla funzione cui questo si applichi.

In realtà , benché si sia certi di "fare meglio" solo con variabili uniformemente distribuite, ci si può accontentare di questo risultato: spesso, come già è stato accennato, non si hanno informazioni analitiche sulla funzione f da integrare e la scelta standard, in questi casi, è proprio quella di distribuzioni uniformi sul dominio di integrazione.

#### 4. Come campionare variabili aleatorie

Abbiamo presentanto i caratteri generali del metodo di Monte Carlo per l'integrazione numerica. Al fine di implementare tale metodo è necessario poter generare valori casuali entro un intervallo con una particolare distribuzione di probabilità , essendo noto come generare campioni uniformemente distribuiti in maniera casuale (o, più precisamente,
*pseudo-casuale*). Consideriamo dapprima il caso discreto e poi quello continuo, nonstante il modo di operare sia sostanzialmente analogo.

Variabili discrete. Consideriamo la variabile aleatoria discreta U, uniformemente distribuita nell'intervallo [0, 1). Data una variabile aleatoria discreta X con distribuzione  $p_X$ , che assume valori  $1, 2, \ldots, n, \ldots$  vogliamo selezionare un campione j con probabilità  $p_X(j)$ . Se  $F_X$  è la funzone di ripartizione di X, dalla (1.6) risulta  $p_X(i) = F_X(i) - F_X(i-1)$  per ogni i nell'immagine di X, inoltre, essendo U uniformemente distribuita, per ogni intervallo  $[\alpha, \beta) \subset [0, 1)$  si ha  $\mathbf{P}(U \in [\alpha, \beta)) = F_U(\beta) - F_U(\alpha) = \beta - \alpha$ , ne segue che la probabilità che un campione j nell'immagine di U sia tale che  $F_X(j) \leq U \leq F_X(j-1)$  è

$$\mathbf{P}\left(U \in \left[F_U(j-1), F_U(j)\right)\right) = F_U(j) - F_U(j-1) = p_X(j),$$

ovvero la probabilità voluta.

Pertanto, un possibile algoritmo per la generazione di campioni con una data distribuzione di probabilità p è il seguente:

## **Algorithm 1** Generare un campione k con probabilità $p_X(k)$

1: Sia X aleatoria con distribuzione  $p_X$ . 2: u = random(0, 1)3: k = random(0, 1)4: while  $u \notin [F_X(k-1), F_X(k)]$  do 5: k = random(0, 1)6: end while 7: k è stato scelto con probabilità  $p_X(k)$ .

Variabili continue. Nel caso di variabili aleatorie continue si procede in maniera del tutto analoga. Supponiamo di voler costruire un campione X con densità  $p_X$ , ovvero tale che  $\mathbf{P}(X \leq x) = \int_{-\infty}^x p_X(\xi) d\xi$ , nota la densità  $p_X$ . Se U è uniformemente distribuita in [0, 1] allora  $F_U(u) = \mathbf{P}(U \leq u) = u$ , posto  $X = F_X^{-1}(U)$ , essendo una funzione di ripartizione non-decrescente, si ha

$$\mathbf{P}(X \leqslant x) = \mathbf{P}(F_X^{-1}(U) \leqslant x) = \mathbf{P}(U \leqslant F_X(x)) = F_X(x) = \int_{-\infty}^x p_X(\xi) \, d\xi$$

Dunque basta scegliere  $X = F_X^{-1}(U)$ , con U variabile aleatoria unforme su [0, 1], per ottenere un campione con la probabilità cercata. Si noti, tuttavia, che tale metodo per la generazione di campioni aleatori richiede di conoscere analiticamente la funzione di ripartizione  $F_X$  e di saperla invertire, cosa che non è sempre possibile.



FIGURA 2. Campionamento di variabili aleatorie continue

ESEMPIO 4.1. Vediamo come, nel caso dell'esempio 2.6, avremmo potuto produrre dei campioni sull'emisfero che fossero distribuiti secondo la densità scelta. Parametrizziamo l'emisfero con gli angoli di Eulero  $0 \leq \varphi < 2\pi, 0 \leq \theta < \pi$ .

Per ridurre l'errore di approssimazione con l'estimatore (ovvero ridurre la varianza) avevamo scelto intuitivamente la distribuzione

$$p(\xi,\zeta) = \frac{1}{\pi}\cos(\zeta)\sin(\zeta).$$

Calcoliamo la sua funzione di ripartizione  $F(\theta, \varphi)$ 

$$F(\theta,\varphi) = \frac{1}{\pi} \int_0^\theta \int_0^\varphi p(\xi,\zeta) \, d\zeta \, d\xi$$
$$= \frac{1}{\pi} \int_0^\theta d\xi \int_0^\varphi \cos\zeta \sin\zeta \, d\zeta =$$
$$= \frac{\theta}{2\pi} \left(1 - \cos^2\varphi\right) = f(\theta)g(\varphi). \quad (4.1)$$

Osserviamo che esistono f e g tali che  $F = f \cdot g$ , ne segue che le variabili aleatorie  $\theta$ ,  $\varphi$  sono indipendenti. Pertanto possiamo considerare due generatori casuali  $U : \Omega \to \{u_i\}_i$  e  $V : \Omega \to \{v_i\}_i$ , indipendenti, uniformemente distribuiti in [0,1], e calcolare i campioni invertendo le funzioni  $f(x) = x/2\pi$  e  $g(x) = 1 - \cos^2 x$ . Si noti che la funzione g è la composizione delle funzioni  $u : x \mapsto \cos x, v : y \mapsto y^2$  e  $w : z \mapsto 1 - z$ , in questo ordine, e che l'inversa di v(y) è la funzione  $\sqrt{y}$  (la radice va presa col segno positivo, perché la variabile y qui vale  $\cos x = \cos \theta$  e nel dominio assegnato per x, ossia  $0 \leq x = \theta < \pi$ , la quantità  $\cos x$ è non negativa). Infine, la funzione w coincide con la propria inversa (se la si applica due volte si trova l'identità, come è anche ovvio dal fatto che w è la riflessione rispetto al punto 1/2). Pertanto la funzione inversa di  $g(x) = 1 - \cos^2 x$  è  $g^{-1}(v) = \arccos\left(\sqrt{1-v}\right)$ . Riassumendo,

$$\theta_i = f^{-1}(u_i) = 2\pi u_i, \quad \varphi_i = g^{-1}(v_i) = \arccos\left(\sqrt{1-v_i}\right), \\
i = 1, 2, 3, \dots$$

Tali valori saranno distribuiti secondo la densità pinizialmente considerata. $\hfill \Box$ 

Un metodo euristico. Non sempre è possibile inverire la funzione di ripartizione di una varibile aleatoria, ancor più se questa è definita su uno spazio a più dimensioni (se si tratta di un vettore aleatorio, come nell'esempio poc'anzi proposto). Vi è un altro metodo per generare campioni con una distribuzione assegnata, detto *rejection sampling*.

Consideriamo il vettore aleatorio  $X = (X_1, \ldots, X_n)$ , con densità di distribuzione congiunta  $p : \mathcal{Q} \subset \mathbb{R}^n \longrightarrow [0, \mu]$ . Indichiamo con  $\mathcal{S}(p)$  il sottografico di p, ovvero l'insieme

$$\mathcal{S}(p) := \{ (\mathbf{x}, \gamma) \in \mathcal{Q} \times [0, \mu] \mid p(\mathbf{x}) \ge \gamma \}, \qquad (4.2)$$

il metodo consiste nel generare campioni  $\mathbf{u}_i = \left(u_1^{(i)}, \ldots, u_n^{(i)}, u_{n+1}^{(i)}\right)$ uniformemente distribuiti in  $\mathcal{Q} \times [0, \mu]$  considerando, al variare di i, solo quelli tali che  $\mathbf{u}_i \in \mathcal{S}(p)$ , scartando gli altri. La distribuzione dei campioni così considerati è una buona approssimazione della distribuzione p che volevamo campionare. Questo metodo, d'altra parte, è fortemente condizionato dalla forma della funzione p o, analogamente, dall'insieme  $\mathcal{S}(p)$ . Infatti se  $|\mathcal{S}(p)| << \mu |\mathcal{Q}|$ , la maggior parte dei campioni  $\mathbf{u}_i$  generati viene scartata e per ottenere una distribuzione che bene approssimi p sono neccessari molti campioni, causando un aumento consistente della varianza del metodo.



 $FIGURA \ 3. \ Rejection \ Sampling$ 

# CAPITOLO 3

# Strategie per il calcolo dell'illuminazione globale

# 1. Formulazione dell'equazione del rendering

Il problema dell'illuminazione globale è soprattutto un problema di trasporto. L'energia emessa da fonti luminose in un ambiente tridimensionale viene propagata da mezzi riflettenti e rifrangenti. Siamo interessati all'equilibrio energetico dell'illuminazione in un ambiente. Dato che il nostro sistema visivo è sensibile al valore della radianza (ossia, determina l'intensità di luce e la sua provenienza direzionale siulla base di questo valore), e dato che vogliamo calcolare immagini fotorealistiche, siamo soprattutto interessati al calcolo dei valori medi della radianza sulle varie aree o angoli solidi della scena. A questo scopo bisogna calcolare i valori del flusso per le diverse aree ed angoli solidi di interesse. La suddivisione in aree scinde la scena in parti (patches) la cui estensione viene stabilita a seconda del livello di accuratezza richiesto. Ad esempio, l'algoritmo di ray tracing scompone la scena in aree che corrispondono a quelle parti delle superficie che sono visibili attraverso i pixel, dal punto di vista dell'osservatore (associate quindi all'angolo solido che esse sottendono viste dall'osservatore); l'algoritmo della radiosità scinde la scena in aree che corrispondono ad opportuni elementi di superficie dove la radiosità si presume varii poco, ciascuno associato all'angolo solido dato dall'intero emisfero frontale (si veda la Figura 1). In ogni algoritmo la scena viene scissa in un numero finito di combinazioni di elementi di superficie ed angoli solidi.

Abbiamo visto come trasformare l'equazione del rendering da un integrale sull'emisfero a un integrale sulla superficie della scena nella sottosezione 7.2 del Capitolo 1. Sia la formula dell'emisfera che quella dell'area contengono funzioni di radianza uscente e incidente. Sappiamo che la radianza rimane invariata lungo percorsi rettilinei, così possiamo facilmente trasformare la radianza uscente in radianza incidente e viceversa, ed ottenere una nuova versione dell'equazione del rendering che contiene solo o la radianza uscente, o quella incidente. Dalla combinazione di queste opzioni con l'integrazione sull'emisfero



FIGURA 1. Esempi di patches per ray tracing e radiosità

o sull'area, otteniamo quattro diverse formulazioni dell'equazione del rendering.



FIGURA 2. Equazione del rendering: integrazione della radianza incidente sull'emisfero

1.1. Radianza uscente, integrazione sull'emisfero. Nella forma classica dell'equazione del rendering, la radianza incidente viene sostituita dalla radianza uscente dal punto della scena in direzione  $\vec{\Theta}$  $y = r(x, \vec{\Theta})$ 



FIGURA 3. Trasporto di radianza uscente usando l'integrazione sull'emisfera

$$L(x \to \vec{\Theta}) = L_e(x \to \vec{\Theta}) + \int_{\Omega_x} f_r(x, \vec{\Psi} \leftrightarrow \vec{\Theta}) L(y \to -\vec{\Psi}) \cos(N_x, \vec{\Psi}) \, d\omega(\vec{\Psi}) \,,$$

 $\operatorname{con} \, y = r(x, \vec{\Theta}).$ 

Quando si progetta un algoritmo basato su questa formula, bisogna integrare sull'emisfero, e per ogni punto del dominio dell'integrale, bisogna tracciare un raggio per trovare il punto di intersezione più vicino con il resto della scena.

1.2. Radianza uscente, integrazione sulle aree. L'equazione sull'emisfero viene trasformata in integrale sui punti delle superficie della scena, vedi figura 4.

$$L(x \to \vec{\Theta}) = L_e(x \to \vec{\Theta}) + \int_{\sigma} f_r(x, \vec{\Psi} \leftrightarrow \vec{\Theta}) L(y \to \vec{xy}) V(x, y) G(x, y) \, d\sigma(y)$$



FIGURA 4. Trasporto della radianza uscente mediante integrazione sulle aree della scena

dove:

$$G(x,y) = \frac{\cos(N_x,\vec{\Psi})\cos(N_y,-\vec{\Psi})}{r_{xy}^2}$$

La differenza con la precendente formula è che la radianza incidente in x è visibile da tutte le superficie della scena, e non solo dall'emisfero. Gli algoritmi che usano questa formulazione devono determinare la visibilità V(x, y) per ogni altro punto y della scena. Punti y diversi possono corispondere alla stessa direzione di angolo  $\vec{\Theta}$ ; quelli in formulazione emisfero, invece, tracciano un solo raggio da x in direzione di  $\vec{\Theta}$ .

1.3. Radianza incidente, integrazione sull'emisfero. Al fine di trasformare l'equazione del rendering nella formulazione emisferica per la radianza incidente, dobbiamo di nuovo usare il fatto che la radianza è costante su un percorso rettilineo, ed è la stessa indipendentemente dal verso di percorrenza. Quindi la radianza uscente si può opportunamnete riscrivere come radianza incidente. (Il concetto di radianza creata incidente  $L_e$  in direzione  $\Theta$  è relativamente facile da immaginare per un punto della superficie: è la radianza che arriva ad x direttamente da una sorgente disposta di fronte a x in direzione  $\Theta$ . Invece il concetto di radianza creata incidente ad un punto x in una sorgente di luce è più inaspettato: significa la radianza che arriva ad un punto x di una sorgente di luce direttamente dal suo punto dirimpettaio nella direzione  $\Theta$ ).

Otteniamo in tal modo l'equazione:

$$L(x \leftarrow \vec{\Theta}) = L_e(x \leftarrow \vec{\Theta}) + \int_{\Omega_x} f_r(y, \vec{\Psi} \leftrightarrow \vec{\Theta}) L(y \leftarrow \vec{\Psi}) \cos(N_y, \vec{\Psi}) \, d\omega(\vec{\Psi}) \,,$$
  
con  $y = r(x, \vec{\Theta}).$ 



FIGURA 5. Trasporto di radianza incidente usando l'integrazione emisferica

**1.4. Radianza incidente, integrazione sulle aree.** Seguendo un processo simile a qello della sottosezione 1.2, otteniamo la seguente formula:

$$L(x \leftarrow \vec{\Theta}) = L_e(x \leftarrow \vec{\Theta}) + \int_{\sigma} f_r(y, \vec{\Psi} \leftrightarrow \vec{xy}) L(y \leftarrow \vec{xy}) V(y, z) G(y, z) \, d\sigma(z)$$
  
con  $y = r(x, \vec{\Theta})$ .



FIGURA 6. Trasporto di radianza incidente usando integrazione su superficie

**1.5. Flusso radiante.** La soluzione teorica per il problema dell'illuminazione globale consiste nel trovare tutti i valori della funzione della radianza per tutte le possibili superficie e tutte le direzioni relative ai punti di queste superfici. È chiaro che questo non è possibile in pratica, in quanto richiederebbe di calcolare un integrale per ogni coppia punto-direzione, e queste coppie sono infinite.

Gli algoritmi di illuminazione globale hanno l'obiettivo di calcolare la radianza *media* su insiemi di punti e direzioni in un'area. Un modo per calcolare il valore medio della radianza è quello di calcolare su quell'area il flusso della varianza. Assumiamo che la radianza cambi lentamente nell'area: quindi il valore medio può essere approssimato dividendo il flusso per l'angolo solido totale e la superficie totale dell'area.

Possiamo esprimere il flusso radiante in termini di radianza integrando la distribuzione della radianza su tutti i possibili punti della superficie e le direzioni attorno a questi punti. Sia  $S = \sigma_s \times \Omega_s$ , l'area dei punti della superficie  $\sigma_s$ , il flusso  $\Phi(S)$  si calcola così:

#### 2. LA FUNZIONE DI IMPORTANZA

$$\Phi(S) = \int_{\sigma_s} \int_{\Omega_s} L(x \to \vec{\Theta}) \cos(N_x, \vec{\Theta}) \, d\omega(\vec{\Theta}) d\sigma(x) \, .$$

Possiamo riscrivere questa formula integrando su tutta la superficie  $\sigma$ e su tutto l'emisfero  $\Omega$  per tutti i punti della scena, nel modo seguente

$$\Phi(S) = \int_{\sigma} \int_{\Omega} L(x \to \vec{\Theta}) W_e(x \leftarrow \vec{\Theta}) \cos(N_x, \vec{\Theta}) \, d\omega(\vec{\Theta}) d\sigma(x)$$

dove abbiamo posto

$$W_e(x \leftarrow \vec{\Theta}) = \begin{cases} 1, & \text{se } (x, \vec{\Theta}) \in S, \\ 0, & \text{se } (x, \vec{\Theta}) \notin S. \end{cases}$$

Il valore della varianza media associato all'insieme di punti e direzioni si esprime allora così:

$$L_{media} = \frac{\int_{\sigma} \int_{\Omega} L(x \to \vec{\Theta}) W_e(x \leftarrow \vec{\Theta}) \cos(N_x, \vec{\Theta}) \, d\omega(\vec{\Theta}) \, d\sigma(x)}{\int_{\sigma} \int_{\Omega} W_e(x \leftarrow \vec{\Theta}) \cos(N_x, \vec{\Theta}) \, d\omega(\vec{\Theta}) \, d\sigma(x)}$$

Dipendendo dalla geometria dell'area S, il denominatore di questa frazione spesso viene calcolato analiticamente.

Il problema dell'illuminazione globale può essere specificato calcolando i flussi radianti per un determinato numero di aree ben definite. Queste aree generalmente sono continue e appartengono allo spazio  $\sigma \times \Omega$ . I flussi sono calcolati con l'equazione (1.1). L'integrale contiene la radice della funzione L, che ha bisogno di essere calcolata usando una delle quattro posibili equazioni ricorsive di Fredholm. Risolvere queste equazioni richiede algoritmi numerici, che sono la base di molti algoritmi di iluminazione globale.

#### 2. La funzione di importanza

2.1. Definizione. Finora abbiamo considerato il problema dell'illuminazione globale come un calcolo della radianza incidente o uscente in un punto x e in una direzione  $\vec{\Theta}$ , data una specifica distribuzione delle sorgenti di luce. Una funzione conosciuta  $L_e$ , definita su tutti i punti e le direzioni che appartengono ad  $\sigma \times \Omega$ , determina una funzione L, anch'essa definita su  $\sigma \times \Omega$ , e che rappresenta il trasporto della radianza. La relazione tra  $L_e$  ed L è data dall'equazione di rendering. Questa sezione affronta queste importanti funzioni e definirà una quantità di trasporto. Daremo una prima occhiata a queste funzioni in maniera intuitiva e in seguito le affronteremo matematicamente inmaniera più approfondita. La funzione di importanza fu introdotta per la prima volta in computer grafica da Pattanaik. Qualche autore la chiama anche funzione potenziale.

Supponiamo di essere interessati al flusso  $\Phi(S)$  che parte da un insieme di punti S, ogniuno dei quali consistente di un sottoinsieme di punti e di direzioni. Inveci di iniziare da una distribuzione fissa  $L_e$ , calcoliamo la possibile influenza di ogni coppia  $(x, \Omega)$  sul flusso  $\Phi(S)$ . Più precisamente: sia un singolo valore di radianza (una fonte di luce che ricopre un'area differenziale e un angolo solido differenziale)  $L(x \to \vec{\Theta})$ , è posizionato su  $(x, \Omega)$ , e se non ci sono altre fonti di illuminazione presenti, quanto sarà grande il valore del flusso  $\Phi(S)$ ? Il peso che dobbiamo attribuire ad  $L(x, \to \Omega)$  per ottenere il flusso  $\Phi(S)$ viene chiamato l'importanza di  $(x, \vec{\Theta})$  tenendo in considerazione S, e viene scritto come  $W(x \leftarrow \Omega)$ .

Il valore di importanza non dipende dalla grandezza di  $L(x \to \vec{\Theta})$ , dato che ogni flusso risultante si ridimensiona linearmente dovuto alla linearità della BRDF prendendo in considerazione la riflessione della radianza incidente. Questo è vero indipendentemente dal numero di possibili percorsi di luce tra  $(x, \vec{\Theta})$  ed S, e senza tener conto del numero di riflessioni. Così  $W(x \leftarrow \vec{\Theta})$  dipende solo dalle geometrie della scena e dalle proprietà di riflessione del materiale.

Nel prossimo passo si definisce un'espressione o un'equazione che descrive l'importanza di  $W(x \to \vec{\Theta})$ . Questa espressione può essere scritta in due modi nei quali i contributi provenienti dal flusso risultante da  $L(x \to \vec{\Theta})$  possono essere create su  $\Phi(S)$ : auto contributo, e contributo attrvero una o più riflessioni.

2.1.1. Auto contributo. Se $(x, \vec{\Theta}) \in S$ , allora  $L(x \to \vec{\Theta})$  contribuisce totalmente a  $\Phi(S)$ . Questa viene chiamata l'auto importanza dell'insieme S e viene scritta come  $W_e(x \leftarrow \vec{\Theta})$  (come nell'equazione 1.1):

$$W_e(x \leftarrow \vec{\Theta}) = \begin{cases} 1, & \text{se } (x, \vec{\Theta}) \in S, \\ 0, & \text{se } (x, \vec{\Theta}) \notin S. \end{cases}$$

2.1.2. Contributo attraverso una o più riflessioni. Dobbiamo anche tenere in considerazione tutti i contributi indiretti al flusso. È possibile che qualche parte di  $L(x \to \vec{\Theta})$  contribuisca a  $\Phi(S)$  attravero o più riflessioni su diverse superficie. Sappiamo che la radianza  $L(x \to \vec{\Theta})$  viaggia lungo un percorso rettilineo e raggiunge un punto  $r(x, \vec{\Theta})$  di una superficie. L'energia viene riflessa su questo punto secondo una distribuzione emisferica determinata dalla BRDF. Così abbiamo un emisfera di direzioni su  $r(x, \vec{\Theta})$ , ogniuna emette un valore di radianza differenziale come il risultato della riflessione della radianza  $L(r(x, \vec{\Theta}) \leftarrow -\vec{\Theta})$ . Integrando i valori importanti per tutte queste nuove direzioni, abbiamo un nuovo termine per  $W(x \leftarrow \vec{\Theta})$ . Prendendo in considerazione la riflessione, e i valori di BRDF, otteniamo la seguente equazione per entrambi i termini combinati:

$$W(x \leftarrow \vec{\Theta}) = W_e(x \leftarrow \vec{\Theta}) + \int_{\Omega_x} f_r(z, \vec{\Psi} \leftrightarrow -\vec{\Theta}) W(z, \leftarrow \Psi) \cos(N_{r(x,\vec{\Theta})}, \vec{\Psi}) \, d\omega(\vec{\Psi}) \,,$$
(2.1)

dove  $z = r(x, \vec{\Theta})$ .

2.2. Importanza incidente e uscente. Matematicamente, l'equazione 2.1 è identica all'equazione del trasporto della radianza incidente. Quindi possiamo associare il concetto di incidenza all'importanza, dal momento che si comportano allo stesso modo.

La funzione sorgente  $W_e$  dipende dalla natura dell'insieme S. Se si vogliono calcolare i valori individuali del flusso per pixel,  $W_e(x \leftarrow \Omega) = 1$  se x è visibile attraverso il pixel e  $\vec{\Theta}$  è una direzione che punta attraverso il pixel di un'immagine della camera virtuale.

Per rafforzare maggirmente l'analogia, possiamo anche introdurre l'importanza usente definendo  $W(x \to \vec{\Theta})$  come segue:

$$W(x \to \vec{\Theta}) = W(r(x, \vec{\Theta}) \leftarrow -\vec{\Theta}).$$

Questa definizione implica l'attribuzione della proprietà dell'invariabilità lungo una linea retta anche all'importanza, che è in accordo con la definizione di radianza e importanza. Quindi è facile provare che l'importanza uscente ha esattamente la stessa equazione di trasporto della radianza uscente:

$$W(x \to \vec{\Theta}) = W_e$$

Poiché non è molto intuitivo pensare ad una radianza incidente autoemessa, la definiamo come importanza di superficie uscente autoemessa. Nel caso di un piccolo insieme con stretto dominio di angoli solidi, l'importanza autoemessa è a volte più facile da visualizzare introducendo un rilevatore di luce. Tale rilevatore non influenza la propagazione dell'energia luminosa nella scena, ma rivela solo il flusso dell'insieme analizzato. L'importanza uscente autoemessa può quindi essere pensata come emessa da questo ipotetico rilevatore, il quale agisce come una sorgente di importanza. Il concetto dei rilevatori di luce funziona estremamente bene quando applicato ad un algoritmo digenerazione di un'immagine, come il ray tracing. Il punto di vista può allora essere considerato come una sorgente di importanza uscente, diretta verso pixel differenti.

**2.3.** Flusso. Possiamo ora dedurre un'espressione per il flusso di un insieme basato sulla funzione di importanza. Le sorgenti di luce sono i soli punti che forniscono energia luminosa all'ambiente. I loro valori di radianza giustificano l'illuminazione dell'intera scena. Quando viene calcolato il flusso, vanno considerati solo i valori di importanza di punti sulle sorgenti luminose. Dato un certo insime S,

$$\Phi(S) = \int_{\sigma} \int_{\Omega_x} L_e(x \to \vec{\Theta}) W((x \leftarrow \vec{\Theta}) \cos(N_x, \vec{\Theta}) \, d\omega(\vec{\Theta}) d\sigma(x) \, .$$

Possimao integrare sopra tutti i punti della superficie  $\sigma$ , dal momento in cui  $L_e$  viene definito 0 su punti e direzioni che non appartengono alle sorgenti di luce. Questa equazione con l'equazione di trasporto della funzione di importanza, fornisce una via alternativa alla soluzione del problema dell'illuminazione globale.

È anche possibile scrivere  $\Phi(S)$  nella seguente forma:

$$\Phi(S) = \int_{\sigma} \int_{\Omega_x} L_e(x \leftarrow \vec{\Theta}) W((x \to \vec{\Theta}) \cos(N_x, \vec{\Theta}) \, d\omega(\vec{\Theta}) d\sigma(x)$$

e anche:

$$\Phi(S) = \int_{\sigma} \int_{\Omega_x} L(x \to \vec{\Theta}) W((x \leftarrow \vec{\Theta}) \cos(N_x, \vec{\Theta}) \, d\omega(\vec{\Theta}) d\sigma(x)$$
$$\Phi(S) = \int_{\sigma} \int_{\Omega_x} L(x \leftarrow \vec{\Theta}) W((x \to \vec{\Theta}) \cos(N_x, \vec{\Theta}) \, d\omega(\vec{\Theta}) d\sigma(x)$$

Così il flusso di un dato insieme può essere calcolato con quattro diversi integrali, ciascuno dei quali può essere calcolato attraverso una doppia integrazione su tutte le superficie o gli emisferi.

Ora siamo in possesso di due diversi approcci per risolvere il problema dell'illuminazione globale. Il primo approccio inizia con la definizione di un insieme e richiede il calcolo dei valori di radianza per i punti e le drezioni che appartengono all'insieme medesimo. I valori di radianza sono calcolati risolvendo una delle equazioni di trasporto descritte dalla radianza. In questo modo, inizamo dall'insieme lavorando verso le sorgenti di luce. Il secondo approccio calcola il flusso di un dato insieme iniziando dalle sorgenti di luce, ed elabora per ogni sorgente di luce il corrispondente valore di importanza con riguardo all'insieme. Il valore di importanza richiede anche l'uso di uno degli integrali ricorsivi. Questo tipo di algoritmo inizia dalle sorgenti di luce e lavorando verso l'insieme segue le ricorsive equazioni di trasporto per importanza.

# CAPITOLO 4

# Algoritmi stocastici di Path Tracing

In questo capitolo viene illustrata una classe di algoritmi per il calcolo dell'illuminazione globale conosciuti con il nome di Path Tracing. La caratteristica peculiare di tali algoritmi è generare dei tracciati di trasporto della luce fra le sorgenti di luce, i punti sui quali vogliamo calcolare il valore della radianza ed il punto di osservazione della scena stessa. Un'altra importante caratteristica di tali algoritmi è che il valore della radianza viene calcolato direttamente per ogni singolo pixel, per questa ragione vengono definiti *basati sui pixel*.

All'interno del capitolo verranno presentati: una breve storia dell'algoritmo Path Tracing nel contesto degli algoritmi dell'illuminazione globale, il setup della camera e vari metodi per il calcolo dell'illuminazione diretta e indiretta. Infine, viene presentato l'algoritmo Light-Tracing evidenziando la dualità con il Ray Tracing.

## 1. Breve storia sulla nascita degli algoritmi Path Tracing

Gli algoritmi Path Tracing, come metodo risolutivo del calcolo dell'illuminazione globale, furono introdotti negli articoli sul Ray Tracing di Whitted [1]. Questi articoli descrivevano un nuovo modo per estendere gli algoritmi a tracciamento di raggi per determinare la visibilità delle superfici in una scena e per includere riflessioni e rifrazioni perfettamente speculari. Al tempo il Ray Tracing era un algoritmo veramente molto lento, con tempo di calcolo direttamente dipendente dal numero di raggi che venivano tracciati attraverso la scena. Il Path Tracing rappresenta una delle tante tecniche sviluppate per migliorare la velocità del calcolo di queste scene. Nel 1984, Cook descrisse il Ray Tracing stocastico<sup>[2]</sup>. I raggi venivano tracciati su percorsi multidimensionali cosí da poter rendere realistici fenomeni come: riflessioni e rifrazioni, sfumatura e la profondità di campo. Kajiya applicò la tecnica del Ray Tracing [3] per risolvere l'equazione del rendering (descritta nel Capitolo 1) che, per essere calcolata, segue esattamente i principi dell'illuminazione globale, includendo tutte le possibili riflessioni tra ogni tipo di superficie all'interno della scena. Altre tecniche di campionamento, che usano il metodo Monte Carlo, furono applicate all'equazione del

rendering. Il metodo bidirezionale per il calcolo del Ray Tracing più completo ed efficiente fu introdotto in seguito da Lafortune e Veach [4].

## 2. Fase iniziale del procedimento di Ray Tracing

In un'immagine calcolata con il metodo dell'illuminazione globale si deve attribuire a ciascun pixel il valore della radianza  $L_{pixel}$ . Questo valore rappresenta una stima pesata della radianza incidente sul piano dell'immagine lungo il raggio proveniente dalla scena, passante attraverso il pixel p e diretto verso l'osservatore, come mostrato in Figura 1.



FIGURA 1. Costruzione geometrica del piano dell'immagine

Questo valore è descritto più accuratamente dall'integrale pesato sul piano dell'immagine, calcolato nel modo seguente:

$$L_{pixel} = \int_{\text{piano}} L(p \leftarrow \text{oss.}) h(p) dp$$
  
= 
$$\int_{\text{piano}} L(\mathbf{x} \leftarrow \text{oss.}) h(p) dp, \qquad (2.1)$$

dove p rappresenta un punto dell'immagine ed h(p) il peso della funzione filtro, mentre  $\mathbf{x}$  è il punto visibile dall'osservatore attraverso p. Spesso h(p) rappresenta un semplice filtro che media il valore della radianza incidente sull'area del pixel. Il settaggio del Ray Tracing si riferisce quindi alla specifica configurazione della scena, del posizionamento della camera e del piano sul quale viene costruita l'immagine. Dobbiamo quindi conoscere il posizionamento esatto della camera, il suo orientamento e la risoluzione del piano dell'immagine. Presupponiamo che l'immagine sia centrata sull'asse di visuale, per stimare  $L(p \to \Delta)$ , viene tracciato un raggio dall'osservatore  $\Delta$  attraverso il punto p, sulla direttrice del punto  $\mathbf{x}$ . Fintanto che rispettiamo la formula  $L(p \to oss.) = L(\mathbf{x} \to \vec{xp})$ , possiamo calcolare il valore della radianza tramite l'equazione del rendering. Un algoritmo di rendering, per disegnare l'intera scena, deve tracciare un raggio attraverso ogni pixel, tramite il quale può stimare la radianza per quel punto. Per ottenere il rendering finale l'algoritmo deve tracciare un raggio primario lungo il quale calcolare la radianza emessa dal punto  $\mathbf{x}$  direttamente verso l'osservatore per ogni punto p del piano dell'immagine.

#### 3. Ray Tracing stocastico semplice

**3.1. Contributo esatto dei tracciati.** Nel metodo a partire dai pixel sopra descritto la radianza viene calcolata esattamente utilizzando l'equazione del rendering. L'algoritmo piú semplice per calcolare tale valore è basato sul metodo di integrazione Monte Carlo nella sua forma standard, applicato all'equazione del rendering. Supponiamo di voler stimare la radianza dal punto  $\mathbf{x}$  lungo il versore  $\vec{\theta}$ , ossia  $L(\mathbf{x} \to \vec{\theta})$ :

$$L(\mathbf{x} \to \vec{\theta}) = L_e(\mathbf{x} \to \vec{\theta}) + L_r(\mathbf{x} \to \vec{\theta}),$$
 (3.1)

dove

$$L_r(\mathbf{x} \to \vec{\theta}) = \int_{\Omega_{\mathbf{x}}} L(\mathbf{x} \leftarrow \vec{\psi}) f_r(\mathbf{x}, \vec{\theta} \leftrightarrow \vec{\psi}) \langle \vec{\psi}, \vec{n}_{\mathbf{x}} \rangle \ d\omega(\vec{\psi}) .$$
(3.2)

L'integrale può essere stimato utilizzando il metodo di Monte Carlo, generando N direzioni casuali  $\vec{\psi_i}$  sull'emisfero  $\Omega_i$ , distribuite secondo la funzione di probabilità  $p(\vec{\psi})$ .

Il valore stimato della radianza  $L(\mathbf{x} \to \vec{\theta})$  è dato da:

$$\left\langle L_r(\mathbf{x} \to \vec{\theta}) \right\rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{L(\mathbf{x} \leftarrow \vec{\psi_i}) f_r(\mathbf{x}, \vec{\theta} \leftrightarrow \vec{\psi_i}) \langle \vec{\psi_i}, \vec{n_x} \rangle}{p(\vec{\psi_i})} \,. \tag{3.3}$$

Il prodotto scalare  $\langle \vec{\psi}_i, \vec{n}_{\mathbf{x}} \rangle$  e la BRDF nell'integrando possono essere calcolati direttamente dai dati geometrici della scena. Il termine  $L(\mathbf{x} \leftarrow \vec{\psi_i})$ , che rappresenta la radianza incidente nel punto  $\mathbf{x}$  è ancora sconosciuto. Considerando il termine

$$L(\mathbf{x} \leftarrow \vec{\psi_i}) = L(r(\mathbf{x}, \vec{\psi_i}) \to -\vec{\psi_i}), \qquad (3.4)$$

dobbiamo tracciare il raggio uscente dal punto  $\mathbf{x}$  in direzione  $\vec{\psi_i}$  attraverso la geometria della scena per trovare eventuali punti di intersezione  $r(\mathbf{x}, \vec{\psi_i})$  con altri elementi della scena stessa. In questo modo abbiamo ottenuto una stima ricorsiva della radianza  $L(\mathbf{x} \leftarrow \Psi_i)$ , ed un raggio, o un albero di raggi, viene tracciato attraverso la scena. Ognuno dei raggi sparati in questo modo fornisce una stima della radianza diversa da 0 solo se la superficie colpita ha una  $L_e$  diversa da 0. In altre parole, il raggio, durante il suo cammino, per ottenere una stima >0 deve colpire una superficie luminosa o una sorgente di luce che di solito ricoprono un'area molto piccola della scena. Il risultato finale è quindi un'immagine quasi completamente nera. Solamente quando un raggio colpisce una sorgente di luce, al corrispondente pixel viene associato un colore. In teoria questo tipo di algoritmi può essere reso più efficiente scegliendo in modo opportuno il termine  $p(\psi)$ , in maniera proporzionale al termine coseno o alla BRDF. In pratica, a causa del grandissimo numero di campioni (ossia percorsi tracciati) con contributo zero, questo miglioramento non aumenta significativamente l'efficienza. In ogni modo, se aumentiamo enormemente il numero di raggi sparati, questo algoritmo elementare è in grado di produrre una scena correttamente illuminata.

**3.2. Il metodo della Roulette Russa.** Il generatore di percorsi stocastici descritto precedentemente riproduce il suo funzionamento in maniera ricorsiva fino ad un punto di arresto. Arrestare le iterazioni dopo un determinato numero di rimbalzi significa però introdurre un *errore sistematico* nell'algoritmo. Fisicamente la luce si riflette infinite volte all'interno di una scena ed alcuni rimbalzi potrebbero essere molto importanti al fine di una corretta visualizzazione. Il calcolo del nostro algoritmo, però, deve avere un numero di ricorsioni finito senza apportare un errore sistematico al calcolo della scena. Il metodo della Roulette Russa riduce sufficientemente il numero di iterazioni per rendere la scena computabile e mantiene casuale la lunghezza dei tracciati percorsi in modo da non introdurre un errore sistematico. Per illustrare tale metodo ci avvaliamo del seguente esempio. Supponiamo di voler calcolare l'integrale:

$$I = \int_0^1 f(\mathbf{x}) \, d\mathbf{x} \,. \tag{3.5}$$

Il metodo d'integrazione Monte Carlo calcola la media dei valori  $f(\mathbf{x}_i)$ nei punti generati a caso  $\mathbf{x}_i$  appartenenti al dominio [0, 1]. Se il calcolo dell'integrale risultasse particolarmente difficile verrebbero generati numerosi punti  $\mathbf{x}_i$ , per ridurre il numero di stime di  $f(\mathbf{x}_i)$  diventa quindi necessario stimare I scalando la funzione  $f(\mathbf{x})$  orizzontalmente tramite un fattore P e verticalmente tramite un fattore 1/P.

$$I_{RR} = \int_0^P \frac{1}{P} f\left(\frac{\mathbf{x}}{P}\right) \, d\mathbf{x},\tag{3.6}$$

con  $P \leq 1$ . È chiaro che  $I_{RR} = I$ . Questo principio è illustrato nel grafico della Figura 2.



FIGURA 2. Principio della Roulette Russa

L'applicazione del medodo Monte Carlo produce il seguente estimatore dell'integrale:

$$\langle I_{RR} \rangle = \begin{cases} \frac{1}{P} f(\frac{\mathbf{x}}{P}) & se \ \mathbf{x} \leqslant P \\ 0 & se \ \mathbf{x} > P \end{cases}.$$
(3.7)

Se  $f(\mathbf{x})$  è un altro integrale ricorsivo, come nel calcolo dell'equazione del rendering, il risultato di questo metodo è che la recursione viene interrotta con una probabilità  $\alpha = 1 - P$  per ogni punto stimato. Di conseguenza  $\alpha$  è chiamata probabilità di assorbimento.

Ovviamente i campioni che cadranno nell'intervallo [P, 1] avranno valore 0, ma questo effetto viene compensato pesando la funzione nell'intervallo [0, P] con il fattore 1/P. In questo modo non aggiungiamo all'estimatore alcun errore sistematico. Se  $\alpha$  é piccolo l'estimatore risulta accurato perché avverranno un gran numero di recursioni, se  $\alpha$  è grande le recursioni si fermeranno prima e quindi avremo una varianza maggiore. In ogni caso non abbiamo introdotto alcun errore sistematico nel calcolo dell'estimatore. La varianza dell'estimatore è di conseguenza:

$$\sigma^{2} = \frac{1}{P^{2}} \int_{0}^{\frac{1}{P}} \frac{f^{2}}{P} \left(\frac{\mathbf{x}}{P}\right) d\mathbf{x} - I^{2} = \frac{1}{P} \int_{0}^{1} f^{2} d\mathbf{x} - \left(\int_{0}^{1} f d\mathbf{x}\right)^{2}.$$

Da questa espressione segue immediatamente che la varianza aumenta al decrescere di P, come del resto è intuitivo visto che, per valori di P via via minori, una percentuale via via più elevata di campioni non contribuisce al calcolo dell'integrale.

La Figura 3 mostra il funzionamento dell'algoritmo Path-Tracyng per il calcolo della radianza nel punto  $\mathbf{x}$ : da  $\mathbf{x}$  partono 4 percorsi di lunghezza stocastica, il percorso a ed il percorso b daranno un contributo positivo perché durante, o alla fine del loro cammino, incontrano la sorgente luminosa, i percorsi  $c \in d$ , invece, non daranno nessun contributo alla radianza del punto  $\mathbf{x}$ .

#### 4. Illuminazione diretta

L'algoritmo di Path Tracing precedentemente illustrato è molto inefficiente, in quanto non tiene assolutamente conto del posizionamento delle sorgenti di luce che di solito coprono una piccola parte della scena: quindi la maggior parte dei tracciati stocastici restituisce un contributo nullo. Questo spreca le risorse di calcolo ed aumenta la varianza del risultato.

4.1. Illuminazione diretta ed illuminazione indiretta. Come illustrato nel Capitolo 1, per il calcolo della radianza emessa da un punto  $\mathbf{x}$  in una direzione  $\vec{\theta}$ , l' equazione del rendering suddivide in maniera esatta le componenti della luce che incidono nel punto  $\mathbf{x}$  in

#### 4. ILLUMINAZIONE DIRETTA



FIGURA 3. Schema di tracciati generati dal Ray Tracing stocastico semplice

luce diretta e luce indiretta. Prendiamo in considerazione la radianza riflessa:

$$L_{r}(\mathbf{x} \to \vec{\theta}) = \int_{\Omega_{\mathbf{x}}} L(\mathbf{x} \leftarrow \vec{\psi}) f_{r}(\mathbf{x}, \vec{\theta} \leftrightarrow \vec{\psi}) \left\langle \vec{\psi}, \vec{n}_{\mathbf{x}} \right\rangle d\omega(\vec{\psi})$$
  
$$= \int_{\Omega_{\mathbf{x}}} L(r(\mathbf{x}, \vec{\psi}) \to -\vec{\psi}) f_{r}(\mathbf{x}, \vec{\theta} \leftrightarrow \vec{\psi}) \left\langle \vec{\psi}, \vec{n}_{\mathbf{x}} \right\rangle d\omega(\vec{\psi}).$$
(4.1)

Riscrivendo  $L(r(\mathbf{x}, \vec{\psi}) \rightarrow -\vec{\psi})$  come la somma della radianza autoemessa e della radianza riflessa  $r(\mathbf{x}, \vec{\psi})$  otteniamo:

$$L_{r}(\mathbf{x} \to \vec{\theta}) = \int_{\Omega_{\mathbf{x}}} L_{e}(r(\mathbf{x}, \vec{\psi}) \to -\vec{\psi}) f_{r}(\mathbf{x}, \vec{\theta} \leftrightarrow \vec{\psi}) \left\langle \vec{\psi}, \vec{n}_{\mathbf{x}} \right\rangle d\omega(\vec{\psi}) + \int_{\Omega_{\mathbf{x}}} L_{r}(r(\mathbf{x}, \vec{\psi}) \to -\vec{\psi}) f_{r}(\mathbf{x}, \vec{\theta} \leftrightarrow \vec{\psi}) \left\langle \vec{\psi}, \vec{n}_{\mathbf{x}} \right\rangle d\omega(\vec{\psi})$$
(4.2)  
$$= L_{diretta}(\mathbf{x} \to \vec{\theta}) + L_{indiretta}(\mathbf{x} \to \vec{\theta}) .$$

Il termine di illuminazione diretta  $L_{diretta}(\mathbf{x} \to \vec{\theta})$  esprime il contributo incidente sulla superficie proveniente direttamente dalla sorgente di luce, quindi possiamo trasformare l'integrale emisferico del punto  $\mathbf{x}$  in un integrale sull'area della sorgente luminosa:

$$L_{dir}(\mathbf{x} \to \vec{\theta}) = \int_{A_s} L_e(r \to \vec{x}\vec{y}) f_r(\mathbf{x}, \vec{\theta} \leftrightarrow \vec{x}\vec{y}) G(\mathbf{x}, \mathbf{y}) V(\mathbf{x}, \mathbf{y}) dA_{\mathbf{y}}.$$
(4.3)

Nel caso in cui sono presenti nella scena diverse sorgenti luminose basta sommare i loro integrali:

$$L_{dir}(\mathbf{x} \to \vec{\theta}) = \sum_{k=1}^{N_L} \int_{A_s} L_e(r \to \vec{x}\vec{y}) f_r(\mathbf{x}, \vec{\theta} \leftrightarrow \vec{x}\vec{y}) G(\mathbf{x}, \mathbf{y}) V(\mathbf{x}, \mathbf{y}) dA_{\mathbf{y}}.$$
(4.4)

Nelle precedenti formule il termine G rappresenta il fattore geometrico che contiene il termine *cos* mentre il termine V rappresenta il fattore di visibilità.

Nella Figura 4 è rappresentato un semplice schema di interazione del modello trattato. In questo modo il numero di percorsi che interessano ciascun punto viene ridotto enormemente e quindi possiamo sparare un numero di campioni maggiore per unità di area, rendendo molto più accurato il calcolo dell'estimatore  $L_r$  rispetto al Path-Tracyng classico. I punti  $\mathbf{x}$  che vedono interamente le sorgenti luminose sono semplici da calcolare e quindi non richiedono un gran numero di campioni, questo metodo, però, permette anche la stima accurata delle aree di penombra. Per queste zone la stima diventa più accurata e quindi, per ottenere una buona resa, dovremmo concentrare qui un maggior numero di tracciati per diminuire la varianza dell'estimatore. Si veda la Figura 6.

4.2. Illuminazione proveniente da una singola sorgente. Per calcolare l'illuminazione diretta del punto  $\mathbf{x}$  proveniente da una singola superficie luminosa dobbiamo definire una densità di probabilità  $p(\mathbf{y})$  relativa all'emettitore. Applicando il metodo Monte Carlo all'equazione 5.3 otteniamo il seguente estimatore:



FIGURA 4. Integrazione per area sulle sorgenti luminose per l'illuminazione diretta

$$\left\langle L_{dir}(\mathbf{x} \to \vec{\theta}) \right\rangle = \frac{1}{N_s} \sum_{i=1}^{N_s} \frac{L_e(\mathbf{y}_i \to \vec{y}_i \vec{x}) f_r(\mathbf{x}, \vec{\theta} \leftrightarrow \vec{x} \vec{y}_i) G(\mathbf{x}, \mathbf{y}_i) V(\mathbf{x}, \mathbf{y}_i)}{p(\mathbf{y}_i)}$$
(4.5)

Per ogni punto  $\mathbf{y}_i$  della sorgente dobbiamo calcolare il valore dell'energia irradiata al punto  $\mathbf{x}$ . Per fare questo dobbiamo tenere in considerazione la BRDF del punto  $\mathbf{x}$ , il fattore di visibilità V, il fattore geometrico G e la radianza emessa nel punto  $\mathbf{y}_i$ . La varianza dell'estimatore dipende quindi direttamente dalla densità di campioni definita da  $p(\mathbf{y})$ .

Teoricamente, per essere accurati, dovremmo campionare tutti i punti della sorgente di luce, ma questo renderebbe il calcolo troppo oneroso. Si può procedere ad esempio campionando in maniera uniforme la superficie luminosa (Figura 5). In questo caso i punti  $\mathbf{y}_i$ 



FIGURA 5. Campionamento uniforme della sorgente di luce per l'illuminazione diretta

sono equamente distribuiti sulla superficie luminosa e la loro densità di probabilità é  $p(\mathbf{y}) = \frac{1}{A_{sorgente}}$ . Questa soluzione è la piu' veloce da calcolare, ma è anche quella che genera la varianza più elevata. Come si può vedere nella Figura 6 è calcolata tracciando per ogni punto  $\mathbf{x}$  rispettivamente 1, 2, 10 e 40 raggi, le differenze si possono apprezzare soprattutto nelle aree di penombra. In queste zone il fattore V di occlusione  $V(\mathbf{x}, \mathbf{y}_i) = 0$  determina se il raggio colpisca o meno la sorgente, di conseguenza, se il numero di raggi è basso la varianza è elevata.

4.3. Illuminazione proveniente da più sorgenti. In una scena reale di solito abbiamo più sorgenti luminose nello stesso ambiente. Per il calcolo dell'illuminazione globale possiamo utilizzare semplicemente il metodo appena descritto per il calcolo dell'illuminazione diretta da ogni singola sorgente. Il calcolo del contributo dalle varie sorgenti luminose puó essere effettuato direttamente su ogni punto  $\mathbf{x}$ , oppure calcolando l'intera scena per ogni singola sorgente e mediare i contributi alla fine.

#### 4. ILLUMINAZIONE DIRETTA



1 raggio di ombra





10 raggi di ombra

40 raggi di ombra

FIGURA 6. Campionamento uniforme della sorgente di luce. Le immagini sono generate utilizzando rispettivamente 1,2,10,40 raggi di ombra per ogni pixel.

Tra i due metodi non varia la complessità di calcolo. Ovviamente per ogni singola stima si deve tener conto della diversa potenza delle varie sorgenti e dell'inverso del quadrato della distanza dal punto  $\mathbf{x}$ .

Un altro possibile approccio è quello di considerare le varie superfici luminose come un'unica sorgente integrata ed utilizzare il metodo Monte Carlo per ottenere dei campioni non condizionati. Questo metodo genera però una varianza molto elevata, si veda la Figura 7. Tipicamente il metodo utilizzato èun algoritmo a doppio passo per ogni raggio di ombra.



FIGURA 7. Campionamento di sorgenti di luce multiple per il calcolo dell'illuminazione diretta.

- **Primo passo.** Viene assegnata una funzione di probabilità discreta  $p_L(k)$  per ogni sorgente luminosa  $k_i$ . Tipicamente questa funzione di probabilità è la stessa per tutte le sorgenti di luce, ma può essere diversa per i vari punti  $\mathbf{x}$  sui quali vogliamo calcolare l'ombreggiatura.
- Secondo passo. In questa fase ogni punto  $\mathbf{y}_i$  appartenente ad una superficie k viene scelto in base ad una  $PDFp(\mathbf{y}/k_i)$ condizionale. La natura di questo PDF è condizionata dalla funzione di probabilità associata alla sorgente stessa. Questo processo è fondamentale per concentrare l'attenzione dei campionamenti sulle sorgenti di luce che irraggiano direttamente le zone interessate dal calcolo ed escludono quelle occluse da un altro oggetto della geometria.

Il PDF combinato per ogni punto  $\mathbf{y}_i$  della superficie emettitrice combinata è:  $p_L(k)p(\mathbf{y}|k)$ . Per N raggi che concorrono al calcolo dell'estimatore avremo quindi:

$$\left\langle L_{dir}(\mathbf{x} \to \vec{\theta}) \right\rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{L_e(\mathbf{y}_i \to \overline{y_i x}) f_r(\mathbf{x}, \vec{\theta} \leftrightarrow \overline{xy_i}) G(\mathbf{x}, \mathbf{y}_i) V(\mathbf{x}, \mathbf{y}_i)}{p_L(k_i) p(\mathbf{y}_i | k_i)}$$

$$(4.6)$$

In ogni caso la scelta del PDF dipendente da  $p_L(k)$  e da  $p(\mathbf{y}|k)$  produce una stima non viziata da errori sistematici. Una corretta scelta dell'estimatore può influire decisamente sulla varianza dei dati e quindi sul disturbo che si avverte nell'immagine calcolata. Alcune delle scelte più comuni sono:

Campionamento uniforme delle sorgenti e della superficie luminosa. In questo caso entrambe le BDF sono uniformi, la probabilità di scelta della sorgente è  $p_L(k) = \frac{1}{N_L}$  e la probabilità di scelta del punto sulla superficie luminosa è  $p(\mathbf{y}|k) = \frac{1}{A_L k}$ . In questo modo ogni sorgente di luce riceve un egual numero di raggi tramite i quali calcolare la radianza equamente distribuiti sull'area dell'emettitore. Questo metodo è di semplice e rapida implementazione, ma le sorgenti luminose non vengono pesate in base alla loro importanza e ricevono un egual numero di raggi, anche se ci troviamo in area di penombra e la geometria della scena ostruisce alcune sorgenti rispetto al punto esaminato. Sostituendo i PDF nell'equazione 4.6 otteniamo il seguente estimatore della radianza emessa in direzione  $\vec{\theta}$ :

$$\left\langle L_{dir}(\mathbf{x} \to \vec{\theta}) \right\rangle = \frac{N_L}{N} \sum_{i=1}^N A_{L_k} L_e(\mathbf{y}_i \to \vec{y}_i \vec{x}) f_r(\mathbf{x}, \vec{\theta} \leftrightarrow \vec{x} \vec{y}_i) G(\mathbf{x}, \mathbf{y}_i) V(\mathbf{x}, \mathbf{y}_i)$$

$$(4.7)$$

Campionamento proporzionale alla potenza di emissione della sorgente con uniforme campionamento della superficie luminosa. In questo metodo il  $PDFp_L(k) = P_k/P_{totale}$  dove  $P_k$  è la potenza della sorgente k e  $P_{totale}$  è la somma della potenza di tutte le sorgenti della scena. Questo metodo assegna quindi un numero di raggi maggiore alle sorgenti di grande intensità riducendo notevolmente la varianza rispetto al metodo precedente. L'estimatore diventa quindi:

$$\left\langle L_{dir}(\mathbf{x} \to \vec{\theta}) \right\rangle = \frac{P_{tot}}{N} \sum_{i=1}^{N} \frac{A_{L_k} L_e(y_i \to \vec{y_i x}) f_r(\mathbf{x}, \vec{\theta} \leftrightarrow \vec{x} \vec{y_i}) G(\mathbf{x}, \mathbf{y}_i) V(\mathbf{x}, \mathbf{y}_i)}{P_k}$$

$$\tag{4.8}$$

Nel caso in cui tutte le sorgenti luminose sono diffusive  $P_k = \pi A_k L_e, k$ l'estimatore della radianza si semplifica nel modo seguente:

$$\left\langle L_{dir}(\mathbf{x} \to \vec{\theta}) \right\rangle = \frac{P_{tot}}{\pi N} \sum_{i=1}^{N} f_r(\mathbf{x}, \vec{\theta} \leftrightarrow \vec{xy_i}) G(\mathbf{x}, \mathbf{y}_i) V(\mathbf{x}, \mathbf{y}_i).$$
 (4.9)

Questo metodo produce ottimi risultati eccetto il caso in cui stiamo analizzando una zona che viene ostruita dalla geometria della scena rispetto alle sorgenti più forti e che potrerbbe ricevere luce da una sorgente debole. Questo problema può essere risolto solamente applicando una strategia di controllo di occlusione del punto rispetto alle sorgenti. Un altro problema di questo metodo consiste nel fatto che vengono generati una grande quantità di indici casuali, vengono generati alberi casuali per i cammini ottici, un numero stocastico per la selezione delle sorgenti k e due numeri casuali per la selezione del punto  $\mathbf{y}_i$  relativo alla superficie luminosa. Questo procedimento genera di conseguenza un gran numero di dati stratificati risultando di difficile implementazione. Esistono però dei metodi matematici, non descritti in questa trattazione, che riducono il problema a due sole variabili casuali restringendo il problema dell'integrazione dei dati ad un dominio bidimensionale. In questo caso diventa molto più semplice stratificare i campionamenti ed applicare efficaci metodi di riduzione della varianza.

#### 5. Illuminazione indiretta

Questa sezione affronta il calcolo della stima dell'illuminazione indiretta di una scena. L'illuminazione indiretta in un punto  $\mathbf{x}$  qualsiasi della scena é un problema di calcolo molto più difficile da affrontare di quella diretta perché il contributo di questa componente, incidente su un punto  $\mathbf{x}$  può essere dato da tutti gli altri punti della scena. Restringere il calcolo ad un particolare dominio di interesse diventa quindi particolarmente difficile.

Il calcolo della componente indiretta interessa l'interazione luminosa che avviene tra il punto esaminato e l'intera scena al più di una interazione tra il punto e le restanti superfici. Questo richiede spesso la maggior parte del tempo di calcolo richiesto da un algoritmo di illuminazione globale, ma é essenziale per ottenere un risultato fotorealistico della scena.

5.1. Campionamento uniforme per l'illuminazione indiretta. Nella sezione 4.1 per stimare l'illuminazione totale emessa da un punto  $\mathbf{x} \ L(\mathbf{x} \to \vec{\theta})$  il contributo incidente sul punto viene diviso nelle componenti diretta ed indiretta. La componente indiretta é rappresentata da:

$$L_{ind}(\mathbf{x} \to \vec{\theta}) = \int_{\Omega_{\mathbf{x}}} L_r(r(\mathbf{x}, \vec{\psi}) \to -\vec{\psi}) f_r(\mathbf{x}, \vec{\theta} \leftrightarrow \vec{\psi}) \left\langle \vec{\psi}, \vec{n}_{\mathbf{x}} \right\rangle \, d\omega(\vec{\psi}) \,.$$
(5.1)

L'integrando contiene il termine  $L_r$  che rappresenta la radianza riflessa da tutti gli altri punti della scena che sono composti a loro volta da una componente diretta ed una indiretta. Il valore  $L_r$  in un ambiente chiuso produce un contributo non nullo per tutte le coppie di punti  $(\mathbf{x}, \vec{\psi})$  che compongono la scena. Quindi dobbiamo integrare l'intero emisfero che ricopre il punto di interesse  $\mathbf{x}$  ed il calcolo di tale integrale deve essere accurato.

Il metodo Monte Carlo applicato alla stima dell'estimatore per l'illuminazione indiretta, viene utilizzato per generare N direzioni casuali  $\vec{\psi_i}$  con probabilità  $p(\vec{\psi})$  sull'emisfero in oggetto. L'estimatore generato sarà:

$$\left\langle L_{ind}(\mathbf{x} \to \vec{\theta}) \right\rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{L_r(r(\mathbf{x}, \vec{\psi_i}) \to -\vec{\psi_i}) f_r(\mathbf{x}, \vec{\theta} \leftrightarrow \vec{\psi_i}) \left\langle \vec{\psi_i}, \vec{n_x} \right\rangle \, d\omega(\vec{\psi})}{p(\vec{\psi_i})}$$
(5.2)

Per la valutazione di questo estimatore dobbiamo calcolare la BRDF ed il prodotto scalare per ogni direzione generata  $\vec{\psi_i}$ , tracciare un raggio dal punto **x** verso la direzione  $\vec{\psi_i}$  e stimare la radianza riflessa  $L_r(r(\mathbf{x}, \vec{\psi_i}) \rightarrow -\vec{\psi_i})$  nel punto di intersezione più vicino ad **x** della retta  $\vec{\psi_i}$ . Da questa considerazione si evince facilmente la natura ricorsiva dell'algoritmo per l'illuminazione indiretta. Ovviamente il numero di rimbalzi può essere arrestato con il metodo della Roulette Russa e quindi il dato rimane privo di errori sistematici. Tipicamente il valore della riflessività locale viene utilizzato per stimare un'appropriata probabilità di assorbimento. La Figura 8 mostra il calcolo per il punto **x**.

5.2. L'importanza del campionamento nell'illuminazione indiretta. La scelta più semplice del fattore  $p(\vec{\psi})$  è quella di prendere



FIGURA 8. Tracciati generati durante il calcolo dell'illuminazione indiretta. I raggi d'ombra utilizzati per il calcolo dell'illuminazione diretta sono rappresentati con le linee bianche.

una PDF uniforme, con  $p(\vec{\psi}) = \frac{1}{2\pi}$ . In questo modo le direzioni ven-gono campionate in maniera uniforme rispetto all'emisfero che ricopre il punto esaminato. Questo metodo inoltre è di semplice e rapida implementazione, però puo' causare un disturbo piuttosto evidente nell'immagine finale perché la varianza della BRDF, della stima del coseno e della radianza riflessa non vengono ottimizzate.

Per ottenere una riduzione del rumore si possono seguire diversi approcci, dobbiamo costruire un PDF sull'emisfero proporzionale ad ognuno dei seguenti fattori:

- Il prodotto scalare  $\langle \vec{\psi_i}, \vec{n_x} \rangle$ . La BRDF  $f_r(\mathbf{x}, \vec{\theta} \leftrightarrow \vec{\psi_i})$ .
- Il fattore della radianza incidente  $L_r(r(\mathbf{x}, \vec{\psi_i}))$ .
- Una combinazione di ognuno dei suddetti elementi.

Le direzioni di campionamento proporzionali al lobo del coseno attorno alla normale  $\vec{n}_{\mathbf{x}}$ evitano di prendere molti punti di campionamento dell'emisfero vicini all'orizzonte, ovvero dove il prodotto scalare  $\langle \vec{\psi}_i, \vec{n}_{\mathbf{x}} \rangle$ 

è 0. Ci aspetteremo una riduzione del rumore, perché in questo modo vengono eliminate le componenti che contribuiscono poco o per nulla al valore dell'estimatore, che diventa:

$$p(\vec{\psi}) = \langle \vec{\psi}_i, \vec{n}_{\mathbf{x}} \rangle / \pi \,. \tag{5.3}$$

Se assumiamo che la BRDF  $f_r$  è diffusivo nel punto **x**, otterremo il seguente estimatore:

$$\left\langle L_{ind}(\mathbf{x} \to \vec{\theta}) \right\rangle = \frac{\pi f_r}{N} \sum_{i=1}^N L_r(r(\mathbf{x}, \vec{\psi_i}) \to -\vec{\psi_i}).$$
 (5.4)

In questo estimatore la componente che puó generare disturbo rimane soltanto la variazione della radianza incidente nell'emisfero esaminato.

5.3. Il campionamento della BRDF. Quando le direzioni di campionamento  $\psi$  sono proporzionali al prodotto scalare, possiamo trascurare la duplice natura della BRDF nel punto **x**, teoricamente le direzioni con un alta BRDF verranno campionate con maggiore probabilità. Questo è un ottimo modo per ridurre il rumore quando siamo in presenza di superfici riflettenti o con un alto valore della BRDF. Per essere maggiormente accurati dovremmo stimare i campionamenti in base al prodotto tra la funzione BRDF e il prodotto scalare. L'esempio illustrato di seguito deriva dal modello di illuminazione di Phong e determina una stima proporzionale alla BRDF, come sopra citato:

$$f_r(\mathbf{x}, \vec{\theta} \leftrightarrow \vec{\psi}) = k_d + k_s \langle \vec{\psi_i}, \vec{\theta_s} \rangle^n , \qquad (5.5)$$

dove il termine  $\vec{\theta}_s$  rappresenta la direzione perfettamente speculare a  $\vec{\theta}$  rispetto a  $\vec{n}_x$ . Questa BRDF ha una parte diffusiva  $k_d$  ed una parte speculare  $k_s \langle \vec{\psi}_i, \vec{\theta}_s \rangle^n$ . L'integrale dell'illuminazione indiretta puo' essere ora diviso in due parti e scritto nella maniera seguente:

$$L_{ind}(\mathbf{x} \to \vec{\theta}) = \int_{\Omega_{\mathbf{x}}} L_r(r(\mathbf{x}, \vec{\psi}) \to -\vec{\psi}) k_d \left\langle \vec{\psi}, \vec{n}_{\mathbf{x}} \right\rangle \, d\omega(\vec{\psi}) \\ + \int_{\Omega_{\mathbf{x}}} L_r(r(\mathbf{x}, \vec{\psi}) \to -\vec{\psi}) k_s \left\langle \vec{\psi}_i, \vec{\theta}_s \right\rangle^n \left\langle \vec{\psi}, \vec{n}_{\mathbf{x}} \right\rangle \, d\omega(\vec{\psi}) \,.$$
(5.6)

5.4. Campionamento di area. Campionare l'area dell'emisfero è il metodo più diretto per calcolare l'integrale dell'illuminazione indiretta. Per ogni direzione campionata viene inviato un raggio per determinare il punto più vicino d'intersezione con la scena. Questa operazione è computazionalmente costosa, ma saremo sicuri di non incorrere in disturbo provocato dall'interazione con oggetti invisibili dal punto  $\mathbf{x}$ .

Come per l'illuminazione diretta ci sono molti modi per descrivere l'integrale dell'illuminazione indiretta, nel caso in cui utilizziamo il metodo sopra descritto otteniamo:

$$L_{ind}(\mathbf{x} \to \vec{\theta}) = \int_{A_{scena}} L_r(\mathbf{y} \to \vec{y}\vec{x}) f_r(\mathbf{x}, \vec{\theta} \leftrightarrow \vec{x}\vec{y}) G(\mathbf{x}, \mathbf{y}) V(\mathbf{x}, \mathbf{y}) dA_{\mathbf{y}}.$$
(5.7)

Il corrispondente estimatore utilizzando una PDF  $p(\mathbf{y})$  è:

$$\left\langle L_{ind}(\mathbf{x} \to \vec{\theta}) \right\rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{L_r(\mathbf{y}_i \to \vec{y}_i \vec{x}) f_r(\mathbf{x}, \vec{\theta} \leftrightarrow \vec{x} \vec{y}_i) G(\mathbf{x}, \mathbf{y}_i) V(\mathbf{x}, \mathbf{y}_i)}{p(\mathbf{y}_i)} \,. \tag{5.8}$$

**5.5.** Costruzione di un algoritmo completo. A questo punto della trattazione abbiamo tutti gli argomenti necessari per costruire un renderer di illuminazione globale completo utilizzando il tracciamento di raggi stocastico. L'efficienza e l'accuratezza dell'algoritmo generato dipenderanno dai seguenti fattori:

Numero di raggi per pixel. Il numero di raggi sparati attraverso un pixel del piano dell'immagine o, più generalmente, il supporto di h(p) è determinante per la riduzione dei fenomeni di aliasing e di rumore presenti nell'immagine finale.

Illuminazione diretta. Per questo fattore sono necessarie una serie di scelte che determinano l'efficienza e l'accuratezza del metodo:

- Il numero totale dei raggi d'ombra  $N_d$  tracciati per ogni punto **x**.
- L'importanza di una singola sorgente di luce rispetto alla somma di tutte le sorgenti della scena.

• La distribuzione dei raggi di ombra sull'area della singola sorgente di luce.

**Illuminazione indiretta.** Di solito questa tecnica si implementa campionando l'emisfero attorno al punto interessato, le componenti di cui dobbiamo tener conto sono:

- Il numero di raggi  $N_i$  di illuminazione indiretta distribuiti sull'emisfero  $\Omega_{\mathbf{x}}$ .
- La distribuzione dei raggi sull'emisfero (coseno, BRDF, ..)
- La probabilità di assorbimento dei raggi effettuata tramite il metodo della roulette russa.

Ovviamente aumentare il numero di raggi per ogni punto, il numero di rimbalzi ed il campionamento dell'emisfero contribuisce alla realizzazione di una immagine priva di disturbo e di altissima accuratezza. Tutto questo però va a discapito della velocità di esecuzione. Di volta in volta si deve trovare il giusto compromesso tra i settaggi del nostro renderer ed i tempi di calcolo.

## 6. Light Tracing

L'algoritmo di Ray Tracing stocastico, descritto nel paragrafo precedente, deriva dall'applicazione del metodo Monte Carlo alla stima dell'equazione del rendering. Il risultato è che l'algoritmo traccia raggi casuali dall'osservatore della scena attraverso il piano dell'immagine e dopo una serie di rimbalzi, i raggi che colpiscono una sorgente di luce, vengono utilizzati per stimare il valore associato al pixel in esame. Questo processo appare del tutto innaturale perchè la luce normalmente parte dalla sorgente emettititrice e dopo una serie di rimbalzi colpisce la retina dell'occhio che percepisce il segnale e ricostruisce l'immagine. L'algoritmo Light Tracing funziona esattamente in questo modo, ossia esamina i raggi uscenti dalla sorgente luminosa. In realtà sfrutta un doppio Ray Tracing combinando i tracciati uscenti dalle luci e dall'occhio e di conseguenza utilizza tutti i metodi di ottimizzazione del calcolo affrontati nei paragrafi precedenti.

**6.1.** Algoritmo Light-Tracing. Questo algoritmo stima il *Potenziale* o *L'equazione di importanza* per ogni singolo pixel. L' equazione di importanza, duale a quella del rendering, è:

$$W(\mathbf{x} \to \vec{\theta}) = W_e(\mathbf{x} \to \vec{\theta}) + \int_{\Omega_{\mathbf{x}}} W(\mathbf{x} \leftarrow \vec{\psi}) f_r(\mathbf{x}, \vec{\theta} \leftrightarrow \vec{\psi}) \left\langle \vec{\psi}, \vec{n}_{\mathbf{x}} \right\rangle \, d\omega(\vec{\psi}) \, .$$

$$\tag{6.1}$$

L'equazione per un singolo pixel è:

$$P = \int_{sorgenti} W(\mathbf{x} \to \vec{\psi}) L_e(\mathbf{x} \to \vec{\psi}) \left\langle \vec{\psi}, \vec{n}_{\mathbf{x}} \right\rangle \, dA(\mathbf{x}) \, d\omega(\vec{\psi}) \,. \tag{6.2}$$

Il funzionamento di questo algoritmo viene mostrato in Figura 9. L'equazione 6.2 viene stimata utilizzando il metodo di integrazione Monte Carlo. I punti  $\mathbf{x}_i$  e le direzioni  $\vec{\theta}_i$  sulla sorgente luminosa vengono generati in base al peso derivante dalla stima dell'importanza  $W(\mathbf{x} \leftarrow \vec{\theta})$ . Questa stima alla fine stabilisce il flusso luminoso attraverso un pixel.

La stima dell'importanza  $W(\mathbf{x} \leftarrow \vec{\theta})$  necessita di una integrazione con il metodo Monte Carlo dell'equazione dell'importanza 6.1. Questa stima è molto simile a quella necessaria per l'equazione del rendering nell'algoritmo Ray Tracing stocastico e può essere utilizzato lo stesso schema: emisfero, area, BRDF, etc.



FIGURA 9. Schema dei tracciati utilizzati dall'algoritmo Light Tracing.

## 6. LIGHT TRACING

Anche i raggi di ombra vengono calcolati in modo analogo al Ray Tracing stocastico, si tiene conto della geometria che occlude la sceda a determinati raggi provenienti dalla sorgente con un fattore di occlusione.

L'algoritmo appena descritto è piuttosto inefficiente in quanto il calcolo deve essere implementato per ogni singolo pixel dell'immagine. Ovviamente il calcolo puo' essere condotto in parallelo e quindi possiamo ottenere un raffinamento prograssivo dell'intera scena.
# CAPITOLO 5

# Radiosità stocastica

# 1. Radiosità classica

Come è noto il metodo di calcolo della radiosità si basa sulla suddivisione della scena in *patches*, cui vengono attribuite luminosità costanti, e che possono a loro volta emettere luce propria. L'algoritmo produce un approssimazione della radiosità della scena a seguito della risoluzione dell'equazione del rendering, discretizzata appositamente, nell'ipotesi che la scena si componga di sole superfici diffusive. Vediamo anzitutto come si possa ottenere una approssimazione lineare discreta dell'equazione integrale del rendering.

Sia N il numero di patches in cui suddividiamo la scena  $\Sigma$ . Indichiamo con  $b_i$  la radiosità della patch *i*-esima, segue dal Capitolo 1 che

$$b_{i} = \frac{1}{A_{i}} \int_{S_{i}} \int_{\Omega(\mathbf{x})} L(\mathbf{x}) \langle \Theta, \mathbf{n}(\mathbf{x}) \rangle \ d\omega(\Theta) dA(\mathbf{x}), \tag{1.1}$$

essendo  $A_i = |S_i|$  l'area della patch *i*-esima ed  $\mathbf{n}(\mathbf{x})$  la normale uscente al punto  $\mathbf{x}$ . Poiché ci interessiamo al caso di superfici puramente diffusive, la radianza  $L(\mathbf{x})$  è indipendente dalla direzione di uscita, precisamente

$$L(\mathbf{x}) = L_e(\mathbf{x}) + \int_{\Omega(\mathbf{x})} f_r(\mathbf{x}) L(\mathbf{x} \leftarrow \Psi) \langle \Psi, \mathbf{n}(\mathbf{x}) \rangle \ d\omega(\Psi).$$
(1.2)

La radianza entrante in  $\mathbf{x}$  è ancora funzione dell'angolo di incidenza ma, come già è stato osservato nel Capitolo 1, se  $\mathbf{y} = r(\mathbf{x}, -\Psi)$  è il punto visibile da  $\mathbf{x}$  in direzione  $-\Psi$ , allora  $L(\mathbf{x} \leftarrow \Psi) = L(\mathbf{y} \rightarrow -\Psi) =$  $L(\mathbf{y})$ , dunque l'equazione (1.2) può essere riformulata come un integrale sull'intera scena  $\Sigma$ , indipendente dalle direzioni

$$L(\mathbf{x}) = L_e(\mathbf{x}) + f_r(\mathbf{x}) \int_{\Sigma} L(\mathbf{y}) \mathcal{K}(\mathbf{x}, \mathbf{y}) \, dA(\mathbf{y})$$
(1.3)

dove indichiamo con  $\mathcal{K}(\mathbf{x}, \mathbf{y}) = G(\mathbf{x}, \mathbf{y})V(\mathbf{x}, \mathbf{y})$  il prodotto del fattore geometrico e di quello di visibilità . Sia  $b(\mathbf{x})$  la radiosità del punto  $\mathbf{x}$ , dalla relazione (Capitolo 1)  $b(\mathbf{x}) = \int_{\Omega(\mathbf{x})} L(\mathbf{x}) \langle \Theta, \mathbf{n}(\mathbf{x}) \rangle \ d\omega(\Theta) = \pi L(\mathbf{x})$  e da (1.1), segue che

$$b_{i} = \frac{1}{A_{i}} \int_{S_{i}} L(\mathbf{x}) \int_{\Omega(\mathbf{x})} \langle \Theta, \mathbf{n}(\mathbf{x}) \rangle \ d\omega(\Theta) dA(\mathbf{x}) = \frac{1}{A_{i}} \int_{S_{i}} b(\mathbf{x}) \ dA(\mathbf{x}).$$
(1.4)

Introducendo la condizione che la radiosità ed il fattore di riflessione bidirezionale siano costanti sulle singole patch, possiamo modificare l'equazione ottenuta come segue

$$b_{i} = \frac{1}{A_{i}} \int_{S_{i}} b^{(e)}(\mathbf{x}) \, dA(\mathbf{x}) + \int_{S_{i}} f_{r}(\mathbf{x}) \int_{\Sigma} \mathcal{K}(\mathbf{x}.\mathbf{y}) b(\mathbf{y}) \, dA(\mathbf{y}) dA(\mathbf{x})$$
  
$$= b_{i}^{(e)} + \sum_{j=1}^{N} \frac{1}{A_{i}} \int_{S_{i}} \int_{S_{j}} f_{r}(\mathbf{x}) \mathcal{K}(\mathbf{x},\mathbf{y}) b(\mathbf{y}) \, dA(\mathbf{y}) dA(\mathbf{x}) = b_{i}^{(e)} + \rho_{i} \sum_{j=1}^{N} F_{ij} b_{j}$$
  
(1.5)

essendo  $\rho_i = f_r(\mathbf{x})$  il fattore di riflessione costante per  $\mathbf{x} \in S_i$  ed  $F_{ij}$  il fattore di forma dalla patch i alla patch j

$$F_{ij} = \frac{1}{A_i} \int_{S_i} \int_{S_j} \mathcal{K}(\mathbf{x}, \mathbf{y}) \, dA(\mathbf{y}) dA(\mathbf{x}). \tag{1.6}$$

Le approssimazioni fatte ci hanno permesso di formulare l'equazione integrale della radiosità nella forma di un sistema lineare (1.5), ovvero di ricondurre il problema al calcolo del vettore **b** soluzione di

$$M\mathbf{b} = \mathbf{b}^{(e)}, \quad (M)_{ij} = \delta(i,j) - \rho_i F_{ij}.$$
 (1.7)

L'approccio classico alla realizzazione del rendering di una scena tramite radiosità si compone quindi di quattro macro-operazioni costitutive per l'algoritmo della radiosità classico:

1. Suddividere la scena in un numero finito N di patches, assegnare ad ogni patch un coefficiente di riflettività  $\rho_i$ ;

**2.** Calcolare  $F_{ij}$  per ogni i = 1, ..., N, j = 1, ..., N risolvendo (o approssimando numericamente) l'integrale quadridimensionale (1.6);

**3.** Risolvere il sistema lineare della radiosità (1.7) numericamente, per mezzo di metodi iterativi quali Jacobi o Gauss-Seidel;

4. Visualizzare i risultati ottenuti, ovvero, ad esempio, convertire in valori di colore il valore di radiosità  $b_i$  ottenuto, per ogni i = 1, ..., N.

Benché non sia del tutto ovvio si intuisce come siano i punti 2 e 3 a dominare il costo computazionale dell'algoritmo. In particolare, forse ancora meno ovvio, la complessità di 3 è spesso trascurabile rispetto a

## 1. RADIOSITÀ CLASSICA

quella richiesta da **2**. Proviamo a convincerci di questo fatto: Cominciamo dando una stima del numero di operazioni per **3** operando con il metodo di Jacobi, la stima della complessità operando con Gauss-Siedel è del tutto analoga. Sia D la matrice diagonale  $(D)_{ii} = (M)_{ii}$  ed A = D - M. Sia  $\Phi(\mathbf{z}) = D^{-1} (\mathbf{b}^{(e)} + A\mathbf{z})$ , il metodo di Jacobi consiste nel calcolare la successione di vettori  $\{\mathbf{b}_k\}_k$  definita da  $\mathbf{b}_{k+1} = \Phi(\mathbf{b}_k)$ , scelta comunque una congettura iniziale  $\mathbf{b}_0 \in \mathbb{R}^N$ . Indichiamo con  $\mathbf{r}_k = \mathbf{b} - \mathbf{b}_k$ , dall'osservare che  $\mathbf{b}$  è punto fisso per  $\Phi$  (ovvero  $\mathbf{b} = \Phi(\mathbf{b})$ ) segue che all'iterazione k-esima si avrà

$$\|\mathbf{r}_{k}\| = \|\Phi(\mathbf{r}_{k-1})\| \leq \|D^{-1}A\|^{k} \|\mathbf{r}_{0}\| \approx \rho(D^{-1}A)^{k} \|\mathbf{r}_{0}\|.$$

La dominanza diagonale di M assicura che esiste  $\gamma > 0$  tale che  $\rho(D^{-1}A) \leq \gamma < 1$ , dunque che  $\mathbf{b}_k \xrightarrow{k\uparrow\infty} \mathbf{b}$ . Il calcolo di ogni iterata  $\mathbf{b}_k$  necessita di una moltiplicazione matrice-vettore (il prodotto  $D^{-1}\mathbf{b}^{(e)}$  essendo calcolato a priori) dunque un numero di operazioni dell'ordine di  $\mathcal{O}(N^2)$ . Stabiliamo una precisione di  $\tau$  cifre significative, abbiamo bisogno di  $k(\tau, \gamma)$  iterazioni, con  $k(\tau, \gamma) \approx -(\tau + \log_{10} \|\mathbf{r}_0\|)(\log_{10} \gamma)^{-1}$  per poter ottenere  $\|\mathbf{r}_{k(\tau,\gamma)}\| \approx 10^{-\tau}$ . In definitiva completare **3** richiede approximativamente complessità  $k(\tau, \gamma)\mathcal{O}(N^2)$ .

Diamo ora una stima grossolana per la complessità del punto **2** Supponiamo di voler approssimare gli integrali per il calcolo degli  $F_{ij}$  con metodi di Monte Carlo (voler calcolare diversamente tali integrali certamente aumenterebbe la mole di operazioni richieste). Si intuisce da subito come la complessità che otterremo sarà strettamente dipendente dalla scelta della distribuzione di probabilità per i campioni dell'estimatore di Monte Carlo. Per ogni fattore di forma richiediamo una precisione di  $\tau$  cifre significative con probabilità  $1 - 10^{-\tau}$ . Indichiamo con  $S_{ij}^{(n)}$  l'estimatore di  $F_{ij}$  con n campioni aleatori. Supponiamo di considerare una distribuzione di probabilità congiunta (di quattro variabili aleatorie) p tale che

$$X = (X_1, X_2, X_3, X_4), \ \Psi_{ij}(X) = \frac{\mathcal{K}(X)}{A_i p(X)} \chi_{(S_i \times S_j)}(X)$$
$$\implies \operatorname{var} (\Psi_{ij}(X)) \leqslant E \left(\Psi_{ij}(X)^2\right) \approx 1 \quad (1.8)$$

Dalla disuguaglianza di Tschebyschev segue che

$$n(\tau) \approx 10^{3\tau} \Longrightarrow \mathbf{P}\left(\left|\mathcal{S}_{ij}^{(n(\tau))} - F_{ij}\right| < 10^{-\tau}\right) \approx 1 - 10^{-\tau} \tag{1.9}$$

Tale numero di campioni è necessario per ogni estimatore  $S_{ij}$ , tuttavia, come mostreremo fra breve, a meno di un fattore di scala  $\alpha_{ij} \in \mathbb{R}$  vale la relazione  $F_{ij} = \alpha_{ij}F_{ji}$ , dunque dal fattore di forma  $i \rightsquigarrow j$  possiamo facilmente ricavare il simmetrico  $j \rightsquigarrow i$ . Ne segue che il numero di estimatori complessivi di cui abbiamo bisogno è dell'ordine di  $\mathcal{O}(N)$ . Il costo complessivo del punto **2** è circa  $n(\tau)\mathcal{O}(N)$ .

Confrontiamo le stime ottenute: se ci accontentiamo di una precisione di  $\tau = 3$  cifre decimali e immaginiamo di suddividere la scena in  $N = \mathcal{O}(10^5)$  patches, completare **2** richiederebbe  $\mathcal{O}(N)n(\tau) \approx 10^{14}$ operazioni. Mentre **3**, nell'ipotesi (spesso verificata sperimentalmente) che  $\gamma \approx 1/2$ , richiede  $\mathcal{O}(N^2)k(\tau,\gamma) \approx \frac{10^{10}\cdot3}{\log_{10}(1/2)} \approx 10^{11}$ . Come ci eravamo proposti di mostrare, anche ad una stima molto grossolana, la complessità computazionale richiesta per **2** domina quella per **3**.

# 2. Fattori di forma e loro proprietà

Il calcolo dei fattori di forma è, dunque, un problema importante e complesso (dal punto di vista computazionale), di centrale importanza nell'algoritmo di radiosità .

PROPOSIZIONE 2.1. Siano  $F_{ij}$ , i = 1, ..., N, j = 1, ..., N come in (1.6), dove ricordiamo che

$$\mathcal{K}(\mathbf{x}, \mathbf{y}) = G(\mathbf{x}, \mathbf{y}) V(\mathbf{x}, \mathbf{y}) = \frac{\langle v(\mathbf{x}, \mathbf{y}), \mathbf{n}(\mathbf{x}) \rangle \langle -v(\mathbf{x}, \mathbf{y}), \mathbf{n}(\mathbf{y}) \rangle}{\pi \|\mathbf{x} - \mathbf{y}\|^2} V(\mathbf{x}, \mathbf{y})$$

essendo  $v(\mathbf{x}, \mathbf{y})$  il versore da  $\mathbf{x}$  in direzione di  $\mathbf{y}$ . Se  $\Sigma$  è una scena chiusa, allora valgono le seguenti proprietà :

$$F_{ij} \ge 0 \quad \forall (i,j) \in [1,N] \times [1,N]$$
(2.1)

$$\sum_{i=1}^{N} F_{ij} = 1 \quad \forall i \in [1, N]$$

$$(2.2)$$

$$A_i F_{ij} = A_j F_{ji} \quad \forall (i,j) \in [1,N] \times [1,N]$$

$$(2.3)$$

DIMOSTRAZIONE. La condizione di non-negatività (2.1) è una conseguenza diretta della definizione di  $\mathcal{K}$ , infatti certamente  $G(\mathbf{x}, \mathbf{y}) > 0$ a meno che  $\mathbf{x}$  ed  $\mathbf{y}$  non siano complanari, inoltre  $V(\mathbf{x}, \mathbf{y}) \in \{0, 1\}$ , dunque  $\mathcal{K}(\mathbf{x}, \mathbf{y}) \ge 0$ . Per provare (2.2) si osservi che dal confronto fra (1.2) e (1.3) (o più direttamente dalle considerazioni fatte nel Capitolo 1) segue che

$$\pi \int_{\Sigma} \mathcal{K}(\mathbf{x}, \mathbf{y}) \, dA(\mathbf{y}) = \int_{\Omega(\mathbf{x})} \langle \Psi, \mathbf{n}(\mathbf{x}) \rangle \, d\omega(\Psi) \tag{2.4}$$

pertanto si ha

$$\sum_{j=1}^{N} F_{ij} = \frac{1}{A_i} \int_{S_i} \sum_{j=1}^{N} \int_{S_j} \mathcal{K}(\mathbf{x}, \mathbf{y}) dA(\mathbf{y}) dA(\mathbf{x}) = \frac{1}{A_i} \int_{S_i} \int_{\Sigma} \mathcal{K}(\mathbf{x}, \mathbf{y}) dA(\mathbf{x})$$
$$= \frac{1}{A_i} \int_{S_i} \frac{1}{\pi} \int_{\Omega(\mathbf{x})} \langle \Psi, \mathbf{n}(\mathbf{x}) \rangle \ d\omega(\Psi) dA(\mathbf{x}) = \frac{1}{A_i} \int_{S_i} dA(\mathbf{x}) = 1.$$

Si osservi che questa proprietà è fisicamente giustificata dalla conservazione dell'energia o, analogamente, dell'angolo solido: l'energia per unità di area, emessa dalla patch *i*-esima che raggiunge tutte le altre patches della scena (essendo  $\Sigma$  chiusa) deve essere pari ad 1.

La proprietà (2.3) è conseguenza diretta dell'uguaglianza  $\mathcal{K}(\mathbf{x}, \mathbf{y}) = \mathcal{K}(\mathbf{y}, \mathbf{x}).$ 

Osserviamo che il sistema (1.7) della radiosità può essere formulato in maniera analoga per le potenze emesse, mantenendo una struttura analoga. Indichiamo con  $D(\mathbf{v})$  la matrice diagonale i cui elementi sono gli elementi del vettore  $\mathbf{v}$ , con  $A \in \mathbb{R}^N$  il vettore delle aree delle patches (ovvero  $(A)_i = A_i$ ), con  $\boldsymbol{\rho}$  il vettore dei fattori di riflettività (ovvero  $(\boldsymbol{\rho})_i = \rho_i = f_r(\mathbf{x})$  per  $\in S_i$ ) e con F la matrice dei fattori di forma  $(F)_{ij} = F_{ij}$ . Se  $w_i$  è la potenza emessa dalla patch *i*-esima, allora (come abbiamo visto nel Capitolo 1)  $w_i = A_i b_i$ , pertanto dalla proprietà (2.3) segue che

$$M\mathbf{b} = (I - D(\boldsymbol{\rho})F)\mathbf{b} = \mathbf{b}^{(e)} \Leftrightarrow D(A)(I - D(\boldsymbol{\rho})F)\mathbf{b} = D(A)\mathbf{b}^{(e)}$$
  
$$\Leftrightarrow (I - D(\boldsymbol{\rho})F^{T})D(A)\mathbf{b} = D(A)\mathbf{b}^{(e)} \Leftrightarrow (I - D(\boldsymbol{\rho})F^{T})\mathbf{w} = \mathbf{w}^{(e)}.$$

Dunque (1.5) è equivalente a

$$w_i = w_i^{(e)} + \sum_{j=1}^N \rho_i w_j F_{ji}, \quad i = 1, 2, \dots, N.$$
 (2.5)

Tale sistema di equazioni ha un interessante interpretazione fisica: la potenza  $w_i$  emessa dalla patch *i*-esima consiste della potenza  $w_i^{(e)}$ , che essa stessa genera, più la percentuale di potenza  $w_j$  che riceve riflessa dalle altre  $j = 1, \ldots, N$  patches; il fattore di forma  $F_{ij}$  indica la frazione di potenza emessa da *i* che raggiunge *j*. Pensare ad i fattori di forma in questi termini suggerisce un metodo per darne una semplice stima in termini probabilistici:

Stima dei fattori di forma con raggi locali. Consideriamo i due vettori aleatori  $(X_i, Y_i) = X(i)$ , uniformemente distribuito su  $S_i$ , e  $\Theta = (\Theta_1, \Theta_2)$  dipendente da X(i), con densità di distribuzione

$$p_{X(i)}(x_i, y_i) = \frac{1}{A_i}, \quad p_{\Theta|X(i)}(\theta_1, \theta_2) = \frac{1}{\pi} \langle (\theta_1, \theta_2), \mathbf{n}(x_i, y_i) \rangle, \quad (2.6)$$

indichiamo con  $\mathbf{x}_i = (x_i, y_i)$  e con  $\theta = (\theta_1, \theta_2)$ , la densità congiunta di  $(X(i), \Theta)$  risulta essere

$$p_{(X(i),\Theta)}(\mathbf{x}_i,\theta) = p_{X(i)}(\mathbf{x}_i) \cdot p_{\Theta|X(i)}(\theta) = \frac{\langle \theta, \mathbf{n}(\mathbf{x}_i) \rangle}{\pi A_i}.$$
 (2.7)

OSSERVAZIONE 2.2. La scelta della distribuzione di  $\Theta$  è ben posta, infatti sono soddisfatte le due proprietà di *non-negatività* e di *normalizzazione* della densità congiunta ottenuta, più precisamente

(i) 
$$p_{(X(i),\Theta)}(\mathbf{x}_{i},\theta) \ge 0, \quad \forall (\mathbf{x}_{i},\theta) \in S_{i} \times \Omega(\mathbf{x})$$
  
(ii)  $\|p_{(X(i),\Theta)}\|_{L^{1}(S_{i} \times \Omega(\mathbf{x}_{i}))} = \int_{S_{i} \times \Omega(\mathbf{x}_{i})} p_{(X(i),\Theta)}(\mathbf{x}_{i},\theta) \, d\omega(\theta) dA(\mathbf{x}_{i})$   
 $= \frac{1}{\pi} \int_{S_{i}} \frac{1}{A_{i}} \int_{\Omega(\mathbf{x}_{i})} \langle \theta, \mathbf{n}(\mathbf{x}_{i}) \rangle \, d\omega(\theta) dA(\mathbf{x}_{i}) = 1$ 

Vediamo come ricavare una formula per i fattori di forma  $F_{ij}$ , facendo uso delle due variabili aleatorie che abbiamo considerato. Definiamo  $n_i$  vettori aleatori  $Y_1^{(i)}, \ldots, Y_{n_i}^{(i)}$  indipendenti ed uniformemente distribuiti, con densità  $p_{Y_k^{(i)}} = p_{(X(i),\Theta)} = p$  per ogni  $k = 1, \ldots, n_i$ . Sia  $f_j(\mathbf{x}_i, \Psi) = \chi_j(\mathbf{x}_i, \Psi) p(\mathbf{x}_i, \Psi)$ , dove  $\chi_j(\mathbf{x}_i, \Psi)$  vale 1 se un raggio da  $\mathbf{x}_i \in S_i$  in direzione  $\Psi$  colpisce la patch j, vale zero altrimenti. Sia  $Y_k^{(i)} = (X_k^{(i)}, \Theta_k^{(i)})$ , consideriamo l'estimatore di Monte Carlo

$$\langle F_{ij} \rangle_{n_i} = \frac{1}{n_i} \sum_{k=1}^{n_i} \frac{f_j\left(Y_k^{(i)}\right)}{p\left(Y_k^{(i)}\right)} = \frac{1}{n_i} \sum_{k=1}^{n_i} \chi_j\left(Y_k^{(i)}\right).$$
 (2.8)

OSSERVAZIONE 2.3. Per ogni  $n_i$  risulta che  $E\left(\langle F_{ij}\rangle_{n_i}\right) = F_{ij}$ , per ogni  $i = 1, \ldots, N, \ j = 1, \ldots, N$  (l'estimatore è unbiased), inoltre  $\operatorname{var}\left(\langle F_{ij}\rangle_{n_i}\right) = \frac{F_{ij}}{n_i}(1 - F_{ij}).$ 

DIMOSTRAZIONE. Tenendo conto della relazione (2.4) si ha

$$E\left(\langle F_{ij}\rangle_{n_i}\right) = \frac{1}{n_i} \sum_{k=1}^{n_i} E\left(\chi_j(Y_k^{(i)})\right)$$
$$= \frac{1}{n_i} \sum_{k=1}^{n_i} \int_{S_i \times \Omega(\mathbf{x})} f_j(\mathbf{x}, \Psi) \, dA(\mathbf{x}) d\omega(\Psi)$$
$$= \frac{1}{\pi} \int_{S_i} \frac{1}{A_i} \int_{\Omega(\mathbf{x})} \chi_j(\mathbf{x}, \Psi) \, \langle \Psi, \mathbf{n}(\mathbf{x}) \rangle \, d\omega(\Psi) dA(\mathbf{x})$$
$$= \frac{1}{A_i} \int_{S_i} \int_{\Sigma} \chi_j(\mathbf{x}, \mathbf{y}) \mathcal{K}(\mathbf{x}, \mathbf{y}) \, dA(\mathbf{y}) dA(\mathbf{x}) = F_{ij}$$

dove  $\chi_j(\mathbf{x}, \mathbf{y})$  vale 1 solo se  $\mathbf{y} \in S_j$  (vale zero altrimenti), essendo  $\mathbf{y} = r(\mathbf{x}, -\Psi)$  il "dirimpettaio" di  $\mathbf{x} \in S_i$  in direzione  $-\Psi$ . Per quanto riguarda la varianza, da quanto osservato nel Capitolo 2, indicata con  $Y^{(i)} = (X^{(i)}, \Theta^{(i)})$  una qualsiasi delle  $n_i$  variabili  $Y_k^{(i)}$ , segue che

$$\operatorname{var}\left(\langle F_{ij} \rangle_{n_{i}}\right) = \frac{1}{n_{i}} \operatorname{var}\left(\chi_{j}(Y^{(i)})\right)$$
$$= \frac{1}{n_{i}} \int_{S_{i} \times \Omega(\mathbf{x})} \left(\chi_{j}(\mathbf{x}, \Psi) - F_{ij}\right)^{2} p\left(\mathbf{x}, \Psi\right) \, d\omega(\Psi) dA(\mathbf{x})$$
$$= \frac{1}{n_{i}} \int_{S_{i} \times \Omega(\mathbf{x})} \left(\chi_{j}^{2} \cdot p + F_{ij}^{2} \cdot p - 2\chi_{j} \cdot p\right) \left(\mathbf{x}, \Psi\right) \, d\omega(\Psi) dA(\mathbf{x})$$
$$= \frac{F_{ij}(1 - F_{ij})}{n_{i}}.$$

Pertanto possiamo interpretare il fattore di forma  $F_{ij}$  come la probabilità che un raggio tracciato da un punto casuale  $\mathbf{x}_i \in S_i$  nella direzione aleatoria  $\Theta$ , raggiunga la patch  $S_j$ , e possiamo stimare tale probabilità con metodi di Monte Carlo, commettendo un errore aleatorio tanto minore quanti più campioni  $n_i$  scegliamo di usare.

Tale metodo, che prende il nome di *local lines sampling*, permette di ridurre drasticamente la complessità computazionale della *macrooperazione* **2** nell'algoritmo di radiosità , migliorando la stima grossolana con ne avevamo dato: Essendo  $0 \leq F_{ij} < 1$  per ogni *i* e *j*, dalla disuguaglianza di Tchebycheff segue che

$$\mathbf{P}\left(\left|\langle F_{ij}\rangle_{n_{i}} - F_{ij}\right| < 10^{-\tau}\right) \ge 1 - \frac{10^{2\tau}F_{ij}(1 - F_{ij})}{n_{i}} \ge 1 - \frac{10^{2\tau}}{4n_{i}},$$

per ogni  $\tau > 0$ , dunque certamente per ogni fattore di forma, scegliendo  $n_i = n_i(\tau) \ge \frac{10^{3\tau}}{4}$ , risulta

$$\mathbf{P}\left(\left|\langle F_{ij}\rangle_{n_i} - F_{ij}\right| < 10^{-\tau}\right) \ge 1 - 10^{-\tau},$$

il che migliora la stima precedente di un fattore  $\frac{1}{4}$ . D'altra parte dalle proprietà (2.1) e (2.2) ci aspettiamo che di frequente  $F_{ij}$  sia molto minore di 1, ad esempio che mediamente si abbia  $F_{ij} \approx \frac{1}{N}$ . Da ciò segue che il numero di campioni necessari, in media, per stimare con probabilità  $1 - 10^{-\tau}$  un fattore di forma  $F_{ij}$ , con precisione  $10^{-\tau}$ , è approssimativamente  $n_i(\tau) \approx \frac{10^{3\tau}}{N}$ , essendo  $F_{ij}(1 - F_{ij}) \approx 1/N$ .

Stima dei fattori di forma con raggi globali. L'algoritmo precedente viene implementato tramite il tracciamento di  $n_i$  raggi che partono in direzioni  $\Theta$  aleatorie da  $\mathbf{x}_i \in S_i$ , per ogni  $i = 1, \ldots, N$  e misurando quanti di questi colpiscono la patch *j*-esima. Esistono, purtuttavia, alcune varianti di tale metodo che permettono di stimare  $F_{ij}$ tracciando un certo numero *n* di raggi globali per tutta la scena, uniformemente distribuiti. Ad esempio si può procedere racchiudendo la scena (di cui a priori conosciamo la geometria) entro una calotta sferica e sparare dei raggi partendo da un punto a caso sulla superficie della sfera. Si può dimostrare che se

- n è il numero complessivo di raggi tracciati,

-  $A_T$  è l'area complessiva delle patches (ovvero  $A_T = \sum_{i=1}^N A_i$ ),

- n(i) è il numero di raggi che raggiungono la patch i,

- n(i, j) è il numero di raggi che attraversando la patch i raggiungono la patch j, allora risulta

$$n(i) \approx n \frac{A_i}{A_T}, \quad F_{ij} \approx \frac{n(i,j)}{n(i)}$$

Il vantaggio principale di questo procedimento sul procedimento di tracciamento locale è dovuto alla possibilitè di sfruttare la *coerenza* nella geometria della scena per generare più efficentemente le linee globali, tuttavia questo tipo di approccio non permette di scegliere il numero di raggi in maniera adattiva, modificandolo di volta in volta a seconda della patch considerata.

# 3. Radiosità stocastica

Abbiamo osservato come ridurre il costo computazionale dovuto ai fattori di forma, dell'algoritmo di radiosità , che stiamo studiando. Come abbiamo sottolineato più volte il calcolo di tali fattori è molto oneroso e, pertanto, sono stati introdotti metodi probabilistici basati su tecniche di Monte Carlo per darne delle stime con una sostanziale riduzione del costo computazionale. Tuttavia, una difficoltà ulteriore cui fin ora abbiamo sottaciuto è quella dovuta alla quantità di memoria necessaria per poter risolvere il sistema (1.7). Basta poco a convincersi che questo è un problema non-irrilevante, nel caso in cui tutti i fattori di forma fossero necessari per definire la matrice M e per risolvere il sistema lineare della radiosità . In questa sezione vedremo come sia possibile, sulla base di quanto visto fin ora, produrre dei metodi per risolvere (1.7) senza dover in verità calcolare **nessun** fattore di forma, sfruttando alcune tecniche di teoria della probabilità . I metodi che vedremo nel dettaglio sono metodi di rilassamento stocastico e metodi basati su catene di Markov discrete.

3.1. Metodi di rilassamento stocastici. Rientrano in questa classe quei metodi per la risoluzione numerica di sistemi lineari che, basati su schemi iterativi di punto fisso (come Jacobi o Gauss-Seidel), riducono il costo computazionale delle iterazioni (che nel caso classico è  $\mathcal{O}(n^2)$ ) stimando con tecniche di Monte Carlo i prodotti matrice-vettore richiesti dal metodo.

Richiamiamo la struttura del sistema lineare cui siamo interessati:

$$M\mathbf{b} = \mathbf{b}^{(e)}, \quad M = I - D(\boldsymbol{\rho})F$$

essendo  $\boldsymbol{\rho}$  il vettore dei coefficienti di riflettività ed F la matrice dei fattori di forma. Limitiamoci a considerare lo schema iterativo di Jacobi. Come è noto, se D è la matrice diagonale  $(D)_{ii} = (M)_{ii}$  ed A = D - M, il metodo di Jacobi consiste nel definire  $\Phi(\mathbf{u}) = D^{-1}(\mathbf{b}^{(e)} + A\mathbf{u})$  e costruire la successione di vettori  $\mathbf{b}_{k+1} = \Phi(\mathbf{b}_k), k = 0, 1, 2, \ldots$  Osserviamo, in particolare, che D = I e che le iterazioni del metodo assumono la forma

$$\mathbf{b}_0 \in \mathbb{R}^N$$
,  $\mathbf{b}_{k+1} = \Phi_F(\mathbf{b}_k) = \mathbf{b}^{(e)} + D(\boldsymbol{\rho})F\mathbf{b}_k$   $k = 0, 1, 2, \dots$  (3.1)

Abbiamo già osservato che il sistema lineare della radiosità è equivalente al sistema delle potenze emesse (2.5), pertanto possiamo riscrivere le iterazioni di Jacobi nella forma

$$\mathbf{w}_0 \in \mathbb{R}^N, \quad \mathbf{w}_{k+1} = \Phi_{F^T}(\mathbf{w}_k) = \mathbf{w}^{(e)} + D(\boldsymbol{\rho})F^T\mathbf{w}_k \quad k = 0, 1, 2, \dots$$
(3.2)

Scegliamo  $\mathbf{w}_0 = \mathbf{w}^{(e)}$ , ne segue che

$$\mathbf{w}_{k+1} = (\Phi_{F^T} \circ \cdots \circ \Phi_{F^T})(\mathbf{w}^{(e)}) = \Phi_{F^T}^k(\mathbf{w}^{(e)}) = \sum_{\nu=0}^k \left( D(\boldsymbol{\rho}) F^T \right)^{\nu} \mathbf{w}^{(e)}$$

Allora possiamo costruire uno schema iterativo equivalente, considerando la potenza incrementale delle patches, ovvero

$$\Delta \mathbf{w}_0 = \mathbf{w}^{(e)}, \quad \Delta \mathbf{w}_{k+1} = D(\boldsymbol{\rho}) F^T \Delta \mathbf{w}_k, \quad k = 0, 1, 2, \dots$$
(3.3)

osservando che, così facendo, il vettore delle potenze calcolato con Jacobi al passo *m*-esimo coincide esattamente con la somma dei primi *m* incrementi  $\Delta \mathbf{w}_j$ , ovvero  $\mathbf{w}_m = \sum_{j=0}^m \Delta \mathbf{w}_j$ .

Dal punto di vista del metodo iterativo puramente deterministico non vi sono differenze (in termini di complessità per iterazione) fra gli schemi (3.1), (3.2) e (3.3), mentre questa ultima formulazione è ottimale se si usano tecniche di Monte Carlo per stimare le approssimanti kesime. Abbiamo osservato, nel metodo delle *linee locali*, che il fattore di forma  $F_{ij}$  può essere interpretato come la probabilità che un raggio tracciato in un direzione casuale  $\Theta$  da un punto a caso della patch i, atterri sulla patch j. Scegliamo con probabilità  $p_k(j)$  la patch j al passo k-esimo, dove

$$p_k(j) = \frac{(\Delta \mathbf{w}_k)_j}{\sum_{i=1}^N (\Delta \mathbf{w}_k)_i} = \frac{\Delta w_j^{(k)}}{\Delta w_T^{(k)}}$$
(3.4)

e sia  $F_{ij}$  la distribuzione di probabilità condizionale di colpire j dato che è stata scelta la patch i. La probabilità complessiva di selezionare una coppia di patch (i, j) in tale modo è

$$p_k(i,j) = p_k(i) \cdot F_{ij} = \frac{F_{ij}\Delta w_i^{(k)}}{\Delta w_T^{(k)}}.$$
 (3.5)

È chiaro che l'idea degli estimatori di Monte Carlo che abbiamo sviluppato per approssimare un integrale, può essere estesa alle sommatorie, nel modo seguente: consideriamo l'insieme limitato  $J \subset \mathbb{Z}^d$ , supponiamo di voler approssimare la somma S di una certa sequenza  $u: J \longrightarrow \mathbb{R}$ 

$$S = \sum_{\underline{i} \in J} u(\underline{i}), \quad \underline{i} = (i^{(1)}, \dots, i^{(d)})$$

Scegliamo una distribuzione discreta di probabilità  $\pi$  ed un insieme di variabili aleatorie  $\{\underline{i}_{\ell}\}$  i.i.d. con distribuzione  $\pi$ . Si osserva, in modo analogo al caso continuo, che per ogni h > 0, la seguente

$$\langle S \rangle_h = \frac{1}{h} \sum_{\ell=1}^h \frac{u(\underline{i}_\ell)}{\pi(\underline{i}_\ell)}, \quad \underline{i}_\ell \in J, \, \forall \ell \, : \, 1 \leqslant \ell \leqslant h$$

definisce un estimatore *unbiased* per S, ovvero tale che  $E(\langle S \rangle_h) = S$ .

Riscriviamo (3.3) scalarmente, con la notazione introdotta in (3.4), introducendo un secondo indice nella sommatoria, per motivi che chiariremo a breve

$$\Delta w_i^{(k+1)} = \sum_{j,r=1}^N \Delta w_j^{(k)} F_{jr} \rho_r \delta(r,i) \quad i = 1, \dots, N$$
 (3.6)

#### 3. RADIOSITÀ STOCASTICA

essendo  $\delta$  la delta di Kronecker. Vogliamo produrre un estimatore di Monte Carlo per (3.6) facendo uso della distribuzione di probabilità pche abbiamo introdotto in (3.5) e ragionando con *local lines*. Selezioniamo a caso  $m_i^{(k)}$  patches  $\{(i_{\nu}, j_{\nu})\}_{\nu}$  con distribuzione p, come abbiamo appena richiamato, l'estimatore per (3.6) assume la forma

$$\left\langle \Delta w_i^{(k+1)} \right\rangle_{m_i^{(k)}} = \frac{1}{m_i^{(k)}} \sum_{\nu=1}^{m_i^{(k)}} \frac{\Delta w_{j_\nu}^{(k)} F_{j_\nu r_\nu} \rho_{r_\nu} \delta(r_\nu, i)}{p_k(j_\nu, r_\nu)} = \rho_i \frac{n_i^{(k)}}{m_i^{(k)}} \Delta w_T^{(k)}$$
(3.7)

dove è stato posto  $n_i^{(k)} = \sum_{\nu=1}^{m_i^{(k)}} \delta(r_{\nu}, i)$ , la frazione delle  $m_i^{(k)}$  patches selezionate che coincide con *i*.

Dunque possiamo riscrivere (3.3) con una formulazione "approssimata", in termini scalari ed in modo che risulti sensibilmente più agevole dal punto di vista computazionale, non essendo **mai** richiesto il calcolo (ne lo storage) dei vari fattori di forma:

$$\begin{cases} \Delta w_i^{(0)} = w_i^{(e)} \\ \Delta w_i^{(k+1)} = \left\langle \Delta w_i^{(k+1)} \right\rangle_{m_i^{(k)}} = \rho_i \frac{n_i^{(k)}}{m_i^{(k)}} \left( \sum_{i=1}^N \Delta w_i^{(k)} \right) , \quad i = 1, \dots, N \\ w_i^{(k)} = \sum_{j=0}^k \Delta w_i^{(j)} \end{cases}$$
(3.8)

Concludendo, facciamo notare che un metodo analogo può essere implementato facendo uso di linee globali, si pone  $p_k(i, j) = \frac{A_i}{A_T}F_{ij}$ , assegnando ad ogni coppia (i, j) la probabilità con la quale una stessa linea globale interseca una coppia di patch i, j mutuamente visibili. Infine va sottolineato che lo schema (3.8) può essere ulteriormente esemplificato scegliendo il numero di *campioni* indipendentemente da ie da k (ovvero si sceglie  $m_i^{(k)} = m$  per ogni  $i \in k$ ), calcolando contemporaneamente tutte le approssimanti del vettore  $\Delta \mathbf{w}_k$ , tuttavia è comunque necessario, elemento per elemento, valutare la quantità  $n_i$ .

**3.2. Catene di Markov discrete.** La seconda classe di metodi per la risoluzione di (1.7), si basa sul concetto di *passeggiata aleatoria su uno spazio degli stati discreto* o, analogamente, su quello di *catena di Markov discreta, finita ed omogenea.* Ci limiteremo a presentare le idee alla base di tali metodi, che non miglioreranno i risultati ottenuti nel caso di "Jacobi stocastico", ma sostanzialmente ne eguaglieranno le prestazioni.

3.2.1. *Definizioni e fondamenti*. Introduciamo preliminarmente il concetto di Catena di Markov finita, discreta ed omogenea, presentandone le proprietà principali. Sia  $\{X_1, \ldots, X_n, \ldots\} = \mathcal{CM}$  un insieme numerabile di variabili aleatorie discrete che assumo un numero finito di valori  $\mathcal{S} = \{1, 2, \ldots, n\}$ (detto spazio degli stati). Diremo che  $\mathcal{CM}$  è omogeneo se ogni variabile aleatoria  $X_j \in \mathcal{CM}$  dipende solo dalla variabile  $X_{j-1} \in \mathcal{S} \setminus \{X_j\}$  mentre è indipendente dalle altre n-2 variabili  $X_i \in \mathcal{CM} \setminus \{X_j, X_{j-1}\}$ . Un insieme di variabili aleatorie  $\mathcal{CM}$  con tali proprietà e tale che la probabilità con cui  $X_{j+1}$  dipende da  $X_j$  è la stessa per ogni  $j = 1, 2, \ldots$ , si dice Catena di Markov discreta, omogenea, con un numero finito di stati (o semplicemente finita).

DEFINIZIONE 3.1 (Probabilità di transizione). Data una catena di Markov finita, discreta ed omogenea, si chiama probabilità di transizione ad un passo da i a j, la probabilità

$$p_{ij} = \mathbf{P}(X_{n+1} = j \mid X_n = i, \dots, X_0 = i_0) = \mathbf{P}(X_{n+1} = j \mid X_n = i),$$

analogamente si dice probabilità di transizione admpassi da i a j la probabilità

$$p_{ij}^{(m)} = \mathbf{P}(X_m = j \mid X_0 = i).$$

Poiché ci interesseremo solo di catene di Markov finite, discrete ed omogenee, sottintederemo che ogni catena nel seguito considerata abbia tali proprietà .

DEFINIZIONE 3.2 (Matrice di transizione). Sia  $\mathcal{CM} = \{X_1, X_2, X_3, ...\}$ una catena di Markov con spazio degli stati  $\mathcal{S} = \{1, 2, ..., n\}$  e probabilità di transizione da *i* a *j* ad un passo  $p_{ij}$ . La matrice  $P \in \mathbb{R}^{n \times n}$ tale che

$$P = \begin{bmatrix} p_{11} & \dots & p_{1n} \\ \vdots & \ddots & \vdots \\ p_{n1} & \dots & p_{nn} \end{bmatrix}$$

si dice matrice di transizione della catena.

OSSERVAZIONE 3.3. Ogni matrice di transizione P di una  $\mathcal{CM}$  è tale che

- (2)  $(P)_{ij} \ge 0$ , per ogni  $i, j \in \mathcal{S}$ .
- (3)  $P\mathbf{e} = \mathbf{e}$ , essendo  $\mathbf{e}^T = (1, \dots, 1)$ .

DIMOSTRAZIONE. La proprietà (2) è ovvia. Mostriamo la proprietà (1) che implica la (3).

<sup>(1)</sup> P è stocastica per righe;

$$\sum_{j=1}^{n} p_{ij} = \sum_{j=1}^{n} \mathbf{P} \left( X_{n+1} = j \mid X_n = i \right) = \mathbf{P} \left( \bigcup_{j \in \mathcal{S}} \left\{ X_{n+1} = j \right\} \mid X_n = i \right)$$
$$= \mathbf{P} \left( X_{n+1} \in \mathcal{S} \mid X_n = i \right) = 1$$

Vale la seguente formula per la probabilità di transizione admpassi, di fondamentale importanza

PROPOSIZIONE 3.4 (Chapman-Kolmogorov).

$$p_{ij}^{(m)} = \sum_{\nu=1}^{n} p_{i\nu}^{(m-1)} p_{\nu j}$$

da cui segue che la probabilità di transizione da i a j ad m passi è data dall'elemento (i, j) della potenza m-esima della matrice di transizione P, più precisamente  $p_{ij}^{(m)} = (P^m)_{ij}$ .

DIMOSTRAZIONE. Dalla proprietà  $\mathbf{P}(A \mid B) = \mathbf{P}(A \cap B)/\mathbf{P}(B)$  segue che

$$p_{ij}^{(m)} = \mathbf{P} \left( X_m = j \mid X_0 = i \right) = \frac{\mathbf{P} \left( X_m = j, X_0 = i \right)}{\mathbf{P} \left( X_0 = i \right)}$$
$$= \sum_{\nu=1}^n \frac{\mathbf{P} \left( X_m = j, X_{m-1} = \nu, X_0 = i \right)}{\mathbf{P} \left( X_{n-1} = \nu, X_0 = i \right)} \cdot \frac{\mathbf{P} \left( X_{n-1} = \nu, X_0 = i \right)}{\mathbf{P} \left( X_0 = i \right)}$$
$$= \sum_{\nu=1}^n \mathbf{P} \left( X_n = j \mid X_{m-1} = \nu, X_0 = i \right) \mathbf{P} \left( X_{m-1} = \nu \mid X_0 = i \right)$$
$$= \sum_{\nu=1}^n p_{i\nu}^{(m-1)} p_{\nu j}$$

DEFINIZIONE 3.5. Siano  $i, j \in \mathcal{S}$  due indici nello spazio degli stati della catena  $\mathcal{S}$ , diremo che *i comunica* con *j* se esiste un m > 0 tale che  $p_{ij}^{(m)} > 0$ . Sia  $\mathcal{C} \subseteq \mathcal{S}$  un sottoinsieme dello spazio degli stati.  $\mathcal{C}$  si dice una classe chiusa di  $\mathcal{S}$  se per ogni  $i, j \in \mathcal{S}$  con  $i \in \mathcal{C}$  e  $j \in \mathcal{C}^{\complement}$  si ha  $p_{ij}^{(m)} = 0$  per ogni m > 0. Una classe chiusa è dunque un sottoinsieme di stati che "non comunicano mai con l'esterno".

Pertanto se per un certo m si verifica  $X_m \in \mathcal{C}$ , con  $\mathcal{C}$  classe chiusa di  $\mathcal{S}$ , allora  $X_q \in \mathcal{C}$  per ogni  $q \ge m$ . Data la classe chiusa  $\mathcal{C}$ , introduciamo

la seguente variabile aleatoria

$$\tau_i(\mathcal{C}) = \inf_{n \in \mathbb{N}} \left\{ X_n \in \mathcal{C} \mid X_0 = i, \, i \notin \mathcal{C} \right\}$$
(3.9)

ovvero il primo "istante" in cui la catena visita  $\mathcal{C}$  essendo partita da  $i \notin \mathcal{C}$ , detta tempo di prima visita della classe chiusa  $\mathcal{C}$ . Vale la seguente

PROPOSIZIONE 3.6. Sia C una classe chiusa di S e  $T \subset S$  il sottoinsieme degli stati di S che non sono in nessuna classe chiusa (detti transitori). Sia  $\zeta_i(C) = E(\tau_i(C))$ , allora  $\zeta_i(C)$  è soluzione della seguente

$$\zeta_i(\mathcal{C}) = 1 + \sum_{j \in T} p_{ij} \zeta_j(\mathcal{C}).$$
(3.10)

Si osservi che l'insieme  $\nu_i(h)$  delle volte che la catena visita lo stato *i*-esimo

$$\nu_i(h) = \{m \in [1, h] : X_m = i\}$$

è anch'essa una variabile aleatoria. Indichiamo con  $\pi$  il vettore delle distribuzioni iniziali di probabilità , ovvero il vettore tale che

- 1.  $\pi_i \ge 0$  per ogni  $i \in \mathcal{S}$ ,
- 2.  $\|\boldsymbol{\pi}\|_1 = 1$ ,
- 3.  $\pi_i = \mathbf{P}(X_0 = i).$

Se  $\gamma_i(h)$  è il numero medio di volte che la catena visita i,  $\gamma_i(h) = E(\nu_i(h))$ , allora  $\gamma_i(h)$  è soluzione dell'equazione

$$\gamma_i(h) = h\pi_i + \sum_{j \in \mathcal{S}} (P^T)_{ij} \gamma_j(h) = h\pi_i + \sum_{j \in \mathcal{S}} p_{ji} \gamma_j(h)$$
(3.11)

per ogni i = 1, ..., n. Studiamo, infine, la situazione in cui si volesse sapere con che probabilità partendo dallo stato *i*-esimo, la catena, dopo un certo numero di passi, vi faccia ritorno. Siamo dunque interessati alla distribuzione di probabilità

$$\chi_i = \mathbf{P} \left( \exists \, m > 0 : X_m = i \mid X_0 = i \right). \tag{3.12}$$

che per ogni  $i \in \mathcal{S}, \chi_i$  soddisfa l'equazione

$$\chi_i = \pi_i + \sum_{j \in \mathcal{S}} (P^T)_{ij} \, \chi_j = \pi_i + \sum_{j \in \mathcal{S}} p_{ji} \, \chi_j, \quad i = 1, \dots, n.$$
(3.13)

Vediamo come sia possibile applicare i concetti qui introdotti per il problema della radiosità (1.7).

**3.3.** Metodi di *Shooting* per la radiosità. Abbiamo già osservato in (3.2) che se il vettore delle radiosità della scena, **b**, è soluzione di (1.7) allora il vettore delle potenze **w** è punto fisso della mappa  $\Phi_{F^T} : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ , ovvero risulta

$$\mathbf{w} = \Phi_{F^T}(\mathbf{w}) = \mathbf{w}^{(e)} + D(\boldsymbol{\rho})F^T\mathbf{w}.$$

Indichiamo con  $\mathbf{z} = \frac{\mathbf{w}}{\|\mathbf{w}^{(e)}\|_1}$  e con  $\mathbf{z}^{(e)} = \frac{\mathbf{w}^{(e)}}{\|\mathbf{w}^{(e)}\|_1}$ , osserviamo che, di conseguenza,  $\|\mathbf{z}^{(e)}\|_1 = 1$  e che  $\mathbf{z}$  è soluzione del sistema

$$\mathbf{z} = \mathbf{z}^{(e)} + (FD(\boldsymbol{\rho}))^T \mathbf{z} \Longrightarrow z_i = z_i^{(e)} + \rho_i \sum_{j=1}^n F_{ji} z_j.$$
(3.14)

Pertanto ogni singola componente  $z_i$  di  $\mathbf{z}$  è soluzione di un equazione del tipo (3.13) e può essere pensata come la distribuzione di probabilità (3.12) associata ad una catena di Markov. Se generiamo con probabilità  $z_i^{(e)}$  dei raggi che partono dalla sorgente i e, facendo uso del metodo delle *local lines*, delle probabilità di transizione  $p_{ij} = \rho_j F_{ij}$ , possono essere utilizzati i seguenti tre metodi di stima per le potenze emesse  $w_i, i = 1, ..., n$ :

(1) (Survival) Facendo uso della probabilità di transizione  $p_{ij}$  viene fatto un test di "accettazione" per ogni altra patch della scena usando per ogni j = 1, ..., N,  $\rho_j$  come probabilità di sporavvivenza. Vengono simulate h variabili aleatorie, costruendole in questo modo, e si registra il numero  $\gamma_i$  di volte che viene visitata la patch *i*-esima. Allora si può stimare  $w_i$  facendo uso della seguente

$$w_i \approx \frac{\gamma_i}{h} \cdot \left\| \mathbf{w}^{(e)} \right\|_1$$

(2) (*Collition*) Il metodo survival implica che vengano sparati dei raggi anche se poi non sempre questi saranno usati per il calcolo di **w**, perché potrebbero non superare il test di sopravvivenza (per esempio succede di frequente se  $\rho_j$  è molto piccolo). Il metodo collision prevede di contare sempre ogni patch che viene colpita da un raggio (senza il test di accettazione), si ottiene così la stima seguente per  $w_i$ 

$$w_i \approx \rho_i \frac{\tilde{\gamma}_i}{h} \cdot \left\| \mathbf{w}^{(e)} \right\|_1$$

dove  $\tilde{\gamma}_i$  indica il numero totale di raggi che colpiscono *i*, il cui valore atteso è  $\tilde{\gamma}_i \rho_i \approx \gamma_i$ .

(3) (Absorption) Quest'ultimo metodo tiene conto solo dei raggi che vengono assorbiti da i (quelli che con survival sono scartati), risulta di conseguenza

$$w_i \approx \frac{\bar{\gamma}_i}{h} \cdot \frac{\rho_i}{1 - \rho_i} \cdot \left\| \mathbf{w}^{(e)} \right\|_1$$

dove  $\bar{\gamma}_i$  è il numero di raggi assorbiti da *i*, tale che  $\tilde{\gamma}_i = \gamma_i + \bar{\gamma}_i$ ,  $(1 - \rho_i)\tilde{\gamma}_i \approx \bar{\gamma}_i$ 

#### 3. RADIOSITÀ STOCASTICA



FIGURA 1. Simulazione del trasporto di un fotone. Selezionata una posizione iniziale, seguiamo il raggio fino a quando non viene assorbito.

3.3.1. Rilassamento stocastico o Random walk? Si può osservare che entrambe le tre tipologie poc'anzi presentate (Survival, Collision, Absorption) danno luogo ad una varianza approssimativamente pari a

$$\frac{\operatorname{var}(\mathbf{b}_{\mathbf{rw}})}{h_{\mathbf{rw}}} \approx \frac{\rho_k}{h_{\mathbf{rw}} A_k} \left\| \mathbf{w}^{(e)} \right\|_1 \left( \mathbf{b}_k - \mathbf{b}_k^{(e)} \right).$$
(3.15)

Dove  $\mathbf{b_{rw}}$  è la radiosità approssimata con tali metodi ed  $h_{\mathbf{rw}}$  è il numero di variabili aleatorie che vengono simulate con metodi di *local lines* (o, analogamente, con linee globali) per emulare il processo di Markov. Una stima analoga si può dare per i metodi di rilassamento stocastico (in particolare per quello di Jacobi, presentato nei dettagli nella sezione 3.1). Facendo uso della formula (3.7) risulta che la varianza della radiosità stimata stocasticamente è

$$\frac{\operatorname{var}(\mathbf{b}_{\mathbf{js}})}{m_{\mathbf{js}}} \approx \frac{\rho_k}{m_{\mathbf{js}} A_k} \|\mathbf{w}\|_1 \left(\mathbf{b}_k - \mathbf{b}_k^{(e)}\right).$$
(3.16)

Anche in questo caso intendiamo con  $\operatorname{var}(\mathbf{b}_{\mathbf{js}})$  la varianza della radiosità ottenuta e con  $m_{\mathbf{js}}$  il numero di campioni usati (indipendentemente dalle patches) per l'estimatore (3.7). Si può mostrare che per simulare un processo di Markov con  $h_{\mathbf{rw}}$  variabili sono necessari, mediamente,  $m = h_{\mathbf{rw}} \frac{\|\mathbf{w}\|_1}{\|\mathbf{w}^{(e)}\|_1}$  raggi. Pertanto se sostituiamo tale espressione in (3.16), ovvero poniamo  $m_{\mathbf{js}} = m$ , si ottiene

$$\frac{\operatorname{var}(\mathbf{b}_{\mathbf{js}})}{m_{\mathbf{is}}} \approx \frac{\operatorname{var}(\mathbf{b}_{\mathbf{rw}})}{h_{\mathbf{rw}}}$$
(3.17)

ovvero che, per lo stesso numero di raggi, **i metodi di stima con** catene di Markov discrete e con rilassamento stocastico (di Jacobi, in particolare), sono approssimativamente efficienti in egual misura.

## 4. Metodi di stima con densità di fotoni

L'idea di questi metodi è quella di risolvere stocasticamente (tramite processi di Markov con spazio degli stati continuo, anziché discreto) l'equazione integrale della radiosità (1.3), piuttosto che il sistema lineare che la approssima (1.5). Il beneficio maggiore che apporta questo tipo di approccio è la possibilità di tener conto della luce emessa non diffusivamente. Ricondurremo il problema del calcolo della radiosità ad un problema di *statistica* del tutto equivalente, presentando alcuni possibili approcci per la sua risoluzione. Teniamo, infine, a sottolineare che tali metodi acquistano maggiore importanza in un ottica generale, in quanto aprono la strada per ulteriori, più recenti, tipi di rappresentazione dell'illuminazione della scena (differenti dall'idea della radiosità ), più sofisiticati ed allo stesso tempo più realistici (come ad esempio la mappa fotonica).

Cominciamo descrivendo un procedimento per la simulazione stocastica della traiettoria dei fotoni, in accordo con le leggi fisiche cui soggiacciono. Sia  $X_0$  una variabile aleatoria con cui indichiamo il punto della scena scelto inizialmente. Costruiamo tale variabile con densità proporzionale alla radiosità emessa

$$p_{X_0}(\mathbf{x}) = \frac{b^{(e)}(\mathbf{x})}{\|\mathbf{w}^{(e)}\|_1},\tag{4.1}$$

detta birth density. Successivamente si sceglie una direzione aleatoria  $\Theta_0$ , dando rilevanza proporzionale al coseno dell'angolo con la normale in  $X_0$ . Dunque si pone la densità di  $\Theta_0 \mid X_0$  pari ad

$$p_{(\Theta_0|X_0)}(\boldsymbol{\theta}, \mathbf{x}) = \frac{1}{\pi} \langle \boldsymbol{\theta}, \mathbf{n}(\mathbf{x}) \rangle.$$
(4.2)

Se con  $X_1$  indichiamo la variabile aleatoria che descrive un secondo punto della scena che viene raggiunto da un raggio sparato da  $X_0$  in direzione  $\Theta_0$ , allora  $(X_1 | X_0, \Theta_0)$  dipende da come la superficie su cui il raggio atterra è orientata (rispetto alla direzione  $\Theta_0$ ), quanto il punto dista da  $X_0$  e dalla mutua visibilità tra i due. Pertanto

$$p_{(X_1|X_0,\Theta_0)}(\mathbf{x}_1,\mathbf{x}_0,\boldsymbol{\theta}) = \frac{\langle -\boldsymbol{\theta}, \mathbf{n}(\mathbf{x}_1) \rangle}{\|\mathbf{x}_0 - \mathbf{x}_1\|_2^2} V(\mathbf{x}_0,\mathbf{x}_1), \qquad (4.3)$$

dove  $V(\mathbf{x}, \mathbf{y})$  è il fattore di visibilità fra  $\mathbf{x}$  ed  $\mathbf{y}$  (si noti che sia  $\boldsymbol{\theta}$  che  $\mathbf{x}$  sono vettori di due componenti). Osserviamo che, facendo uso della proprietà  $\mathbf{P}(A, B) = \mathbf{P}(A \mid B)\mathbf{P}(B)$ , si ottiene la formula

$$\mathbf{P}(C \mid A, B)\mathbf{P}(A, B) = \mathbf{P}(C, B, A) = \mathbf{P}(C, B)\mathbf{P}(A)$$

da cui, dividendo per  $\mathbf{P}(A, B)$ ,

$$\mathbf{P}(C \mid A, B) = \frac{\mathbf{P}(C, B)}{\mathbf{P}(B \mid A)}$$

Pertanto la densità della variabile aleatoria  $(X_1 \mid \Theta_0)$  risulta essere

$$p_{(X_1|\Theta_0)}(\mathbf{x}_1, \mathbf{x}_0) = p_{(X_1|X_0, \Theta_0)}(\mathbf{x}_1, \mathbf{x}_0, \boldsymbol{\theta}) p_{(\Theta_0|X_0)}(\boldsymbol{\theta}, \mathbf{x}) = \mathcal{K}(\mathbf{x}_0, \mathbf{x}_1),$$
(4.4)

dove ricordiamo che

$$\mathcal{K}(\mathbf{x}, \mathbf{y}) = G(\mathbf{x}, \mathbf{y}) V(\mathbf{x}, \mathbf{y}) = \frac{\langle v(\mathbf{x}, \mathbf{y}), \mathbf{n}(\mathbf{x}) \rangle \langle -v(\mathbf{x}, \mathbf{y}), \mathbf{n}(\mathbf{y}) \rangle}{\pi \|\mathbf{x} - \mathbf{y}\|_2^2} V(\mathbf{x}, \mathbf{y}),$$

essendo  $v(\mathbf{x}, \mathbf{y})$  il versore della direzione da  $\mathbf{x}$  a  $\mathbf{y}$ . Successivamente viene fatto un test di "sopravvivenza" sul punto  $X_1$  così ottenuto. Bisogna decidere se il raggio viene assorbito o riflesso. Indichiamo con  $\sigma(X_1)$  la probabilità che avvenga una riflessione, scegliamo tale densità pari alla frazione (albedo) di luce  $\rho(X_1, -\Theta_0)$  che, provenendo dalla direzione di  $X_0$ , viene riflessa in  $X_1$ . Se la superficie di  $X_1$  è diffusiva l'albedo coincide con il coefficiente di riflettività  $f_r(X_1) = \rho(X_1)$ , pertanto la probabilità di transizione da  $X_0$  ad  $X_1$  è

$$p_{(X_1|X_0)} = p_{(X_1|\Theta_0)} \cdot \rho(X_1) = \mathcal{K}(\mathbf{x}_0, \mathbf{x}_1)\rho(\mathbf{x}_1).$$
(4.5)

Si procede iterando questo schema per alcuni altri punti della scena. Consideriamo il valore atteso  $\chi_X$  di raggi che si addensano in una zona in prossimità di X, al seguito di tale simulazione, per unità di area. Tale densità consiste di due contributi: la densità di probabilità  $S_X$  di raggi che sono inizialmente fatti partire da X e la densità di raggi che raggiungono nuovamente X dopo aver visitato altri punti Y della scena Tale valore soddisfa l'acuazione integrale

. Tale valore soddisfa l'equazione integrale

$$\chi_X(\mathbf{x}) = S_X(\mathbf{x}) + \int_{\Sigma} \chi_Y(\mathbf{y}) p_{(X|Y)}(\mathbf{x}, \mathbf{y}) dA(\mathbf{y}),$$

in particolare per lo schema diffusivo che abbiamo considerato, risulta

$$\chi_X(\mathbf{x}) = \frac{b^{(e)}(\mathbf{x})}{\|\mathbf{w}^{(e)}\|_1} + \int_{\Sigma} \chi_Y(\mathbf{y}) \mathcal{K}(\mathbf{y}, \mathbf{x}) \rho(\mathbf{x}) dA(\mathbf{y}).$$

pertanto da (1.3) segue che

$$\chi_X(\mathbf{x}) = \frac{b(\mathbf{x})}{\|\mathbf{w}^{(e)}\|_1}.$$
(4.6)

Abbiamo ottenuto, dunque, che il numero medio di raggi che per unità di area colpiscono la superficie in prossimità di X è direttamente proporzionale alla radiosità  $b(\mathbf{x})$  in tale punto. Il problema della radiosità è stato ricondotto al problema di stimare la densità di raggi che colpiscono un dato punto x della scena.

Il problema di stimare una tale densità è stato ampiamente studiato in statistica. Ci proponiamo di concludere presentando l'idea alla base di alcuni dei principali metodi di stima statistica per tale quantità .

Il metodo dell'istogramma. Questo è il metodo, probabilmente, più usato per la stima di densità . L'idea alla base è piuttosto semplice (e questo è il suo pregio principale e ciò che lo rende molto popolare): la scena viene suddivisa in N elementi  $E_1, \ldots, E_N$  di dimensione ridotta  $A_i$  (nel nostro caso le patches) e vengono contati il numero  $n_i$  di raggi generati, che cade in ogni elemento, per  $i = 1, \ldots, N$ . Il rapporto  $\frac{n_i}{A_i}$ risulta essere un approssimazione della densità di raggi che colpisce ogni elemento, dunque ogni punto  $\mathbf{x} \in E_i$ .

Il metodo delle basi ortogonali. Tale metodo si basa, come il precedente, nel suddividere la scena in N elementi  $\{E_i\}_{i=1,...,N}$ . L'idea è quella di approssimare il valore della radiosità  $b_i(\mathbf{x})$  della patch *i*esima, in un sottospazio di Hilbert finito dimensionale di  $L^2$ , in modo da generalizzare il metodo precedente, ottenendo un approssimazione migliore. Consideriamo il sottospazio

$$\mathbb{V} = \operatorname{Span} \left\{ \phi_1, \phi_2, \dots, \phi_\nu \right\} \subset L^2(\Sigma)$$
(4.7)

essendo  $\{\phi_1, \phi_2, \ldots, \phi_\nu\}$  un insieme di funzioni linearmente indipendenti di  $L^2$ . Data  $f \in L^2(\Sigma)$ , indichiamo con  $f_{\mathbb{V}}$  la funzione di  $\mathbb{V}$  che



FIGURA 2. Le immagini mostrano la stessa scena *renderizzata* con densità di raggi per patch pari a 100, 500, 1000, 10000, rispettivamente.

meglio approssima f, ovvero tale che

$$||f - f_{\mathbb{V}}||_{L^{2}(\Sigma)} = \min_{v \in \mathbb{V}} ||f - v||_{L^{2}(\Sigma)}.$$

Indichiamo con  $\boldsymbol{\alpha}_f$  il vettore dei coefficienti di  $f_{\mathbb{V}}$ , ovvero

$$f_{\mathbb{V}}(\mathbf{x}) = \sum_{i=1}^{\nu} \left( \boldsymbol{\alpha}_f \right)_i \phi_i(\mathbf{x})$$

Sia  $\langle \cdot, \cdot \rangle$  il prodotto scalare  $L^2(\Sigma)$ , dal noto teorema di proiezione di Hilbert segue che  $f_{\mathbb{V}}$  è l'unica funzione di  $\mathbb{V}$  per cui risulti

$$\langle f_{\mathbb{V}} - f, v \rangle = 0 \quad \forall v \in \mathbb{V} \iff \sum_{j=1}^{\nu} (\boldsymbol{\alpha}_f)_j \langle \phi_i, \phi_j \rangle = \langle f, \phi_i \rangle \quad \forall 1 \leqslant i \leqslant \nu$$

ovvero si possono caratterizzare i coefficienti di  $f_{\mathbb{V}}$  come la soluzione di un sistema lineare di  $\nu$  equazioni in  $\nu$  incognite. Si osserva facilmente che se  $\{\phi_i\}_{i=1}^{\nu}$  è una base ortogonale per  $\mathbb{V}$ , allora si può dare la seguente

100

formula esplicita per l'approssimante  $f_{\mathbb{V}}$ 

$$f_{\mathbb{V}}(\mathbf{x}) = \sum_{j=1}^{\nu} \frac{\langle f, \phi_j \rangle}{\langle \phi_j, \phi_j \rangle} \phi_j(\mathbf{x}).$$
(4.8)

L'idea del metodo che qui discutiamo è quella di scegliere opportuni sottospazi  $\mathbb{V}$ , considerare opportune basi ortogonali per tali sottospazi ed approssimare la radiosità su ogni elemento considerando  $b_{\mathbb{V}}(\mathbf{x})$  al posto di  $b(\mathbf{x})$ . La scelta tipica per il sottospazio  $\mathbb{V}$  è quella dei polinomi di grado minore o uguale a k > 0, che indichiamo con  $\mathbb{P}_k$ , oppure, molto in voga di recente, quella delle B-Spline (quasi sempre con nodi equi spaziati), di grado massimo k > 0, che indichiamo con  $\mathbb{S}_k$ . Si noti, ad esempio, che se si sceglie come base del sottospazio  $\mathbb{V} = \mathbb{S}_1$  l'insieme delle funzioni caratteristiche dei domini  $E_i$  (ovvero  $\phi_i = \chi_{E_i}$ ), allora si riottiene il metodo dell'istogramma.

Il metodo di Nearest Neighbor. Questo metodo prende spunto da un idea di Jensen e dall'algoritmo di Photon Mapping che egli ha sviluppato. L'idea è simile a quella del metodo dell'istogramma, tuttavia si lavora "cambiando punto di vista": invece di fissare un certo elemento  $E_i$  e contare il numero di raggi  $n_i$  che lo colpiscono, si fissa un numero n e si cerca un elemento che abbia ricevuto un tale numero di raggi. Se un tale numero n viene trovato su un elemento  $E_i$  di area piccola, allora la densità  $n/A(E_i)$  sarà elevata, e viceversa. Questo tipo di approccio permette di limitare la quantità di memoria da occupare, infatti necessita solo di tener memoria dei punti colpiti dai raggi.

# Bibliografia per la Parte III (Illuminazione Globale)

- T. Whitted, An Improved Illumination Model for Shaded Display, Communications ACM, 23 (1980), 343-349.
- [2] R. Cook, T. Porter, L. Carpenter, *Distributed Ray Tracing*, Computer Graphics, 18 (1984), 137-145.
- [3] J.T. Kajiya, The Rendering Equation, Computer Graphics 18 (1986), 143-150.
- [4] E.Veach, L.J. Guibas, Bidirectional Estimators for Light Transport, in Fifth Eurographics Workshop on Rendering, Darmstadt 1994, 147-162. 1
- 1