A Unified Tight Frame Approach for Missing Data Recovery

Raymond H. Chan
Department of Mathematics
The Chinese University of Hong Kong

Rome-Moscow School of Numerical Math

September 19-October 24, 2010

Supported by Hong Kong Research Grant Council

In collaboration with

Tony Chan

(UCLA, NSF)

Lixin Shen

(Syracuse University)

Zuowei Shen, Xia Tao, Jian-Feng Cai
(National University of Singapore)

Gabi Steidl, Simon Setzer
(University of Mannheim)

Application: Inpainting

given image **y**

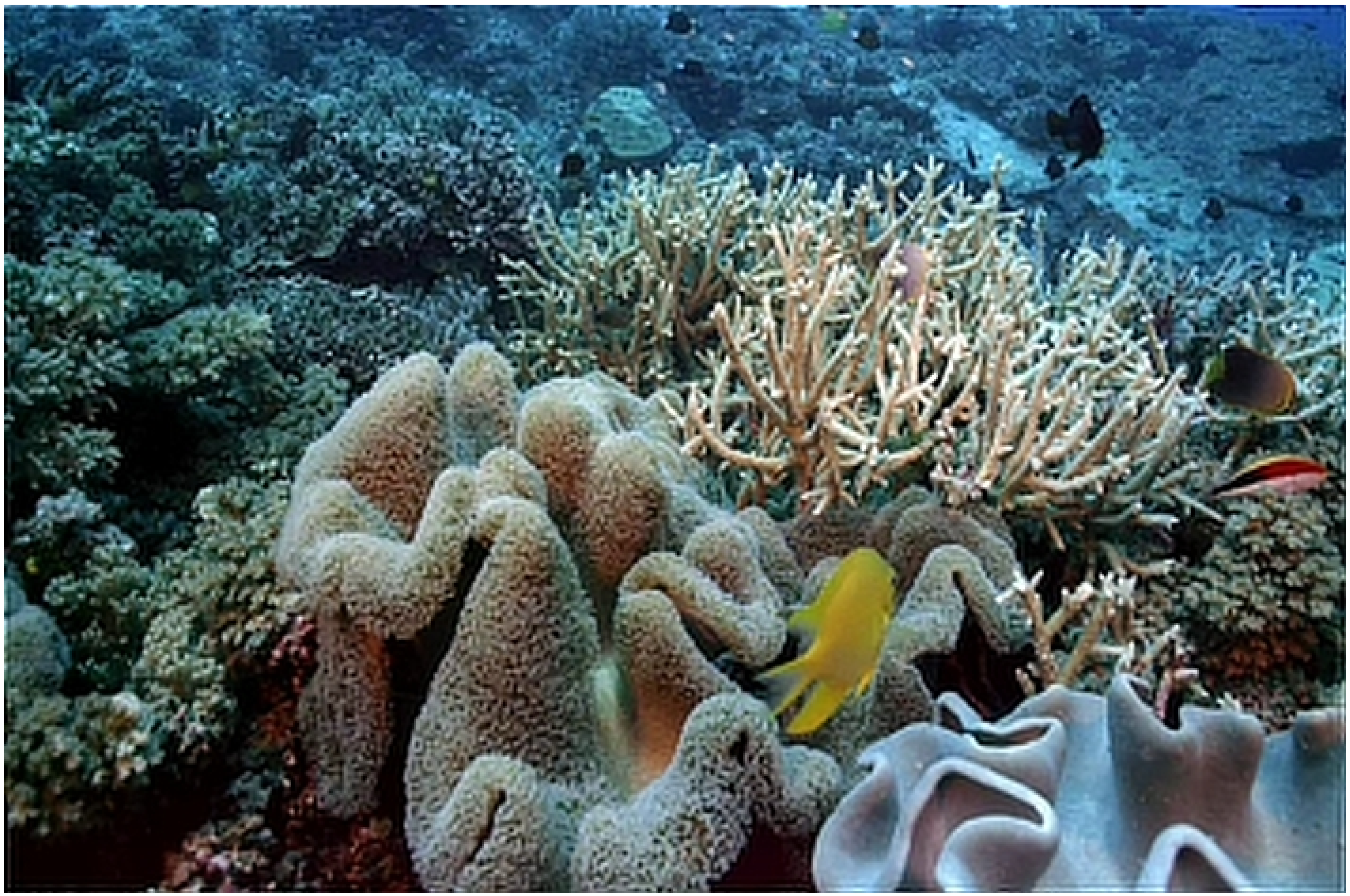
restored image **f**

Application: Noise Removal

given image y

restored image f

Application: High-Resolution Image Reconstruction

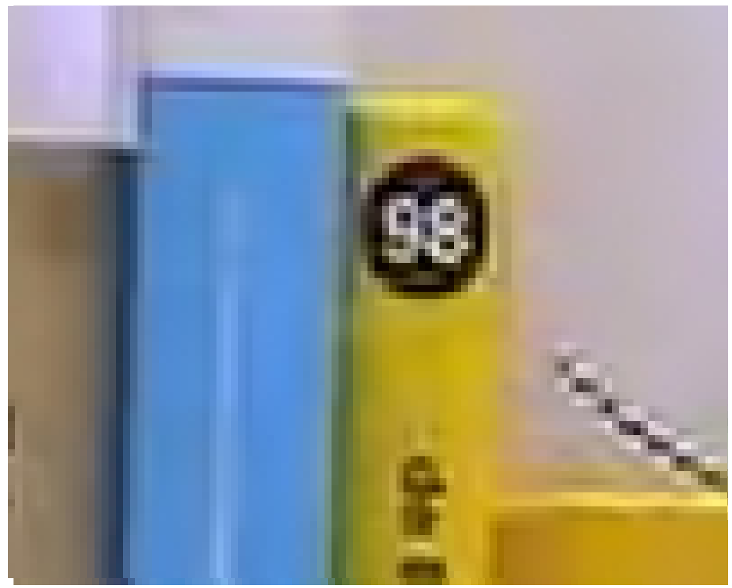


Bicubic interpolation

Application: Video Enhancement

A 352-by-288 video
from a video recorder

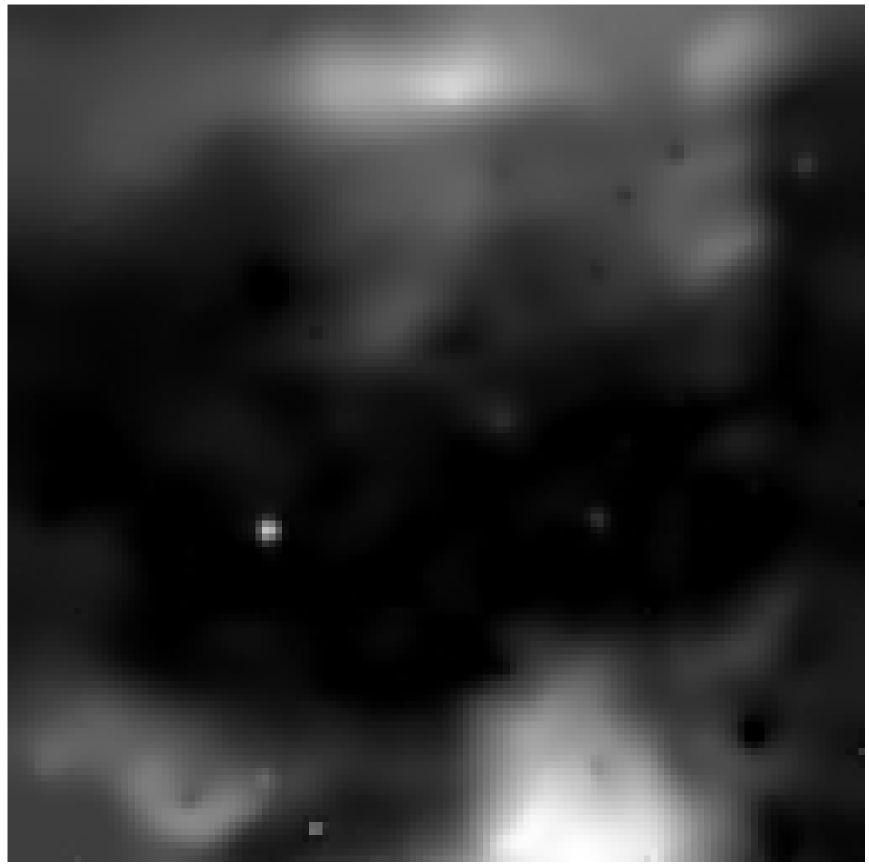
Application: Video Enhancement



Bilinear method

Tight-frame method

Application: Astronomical Infrared Imaging



Observed Image from United Kingdom Infrared Telescope **y**

Reconstruction image **f**

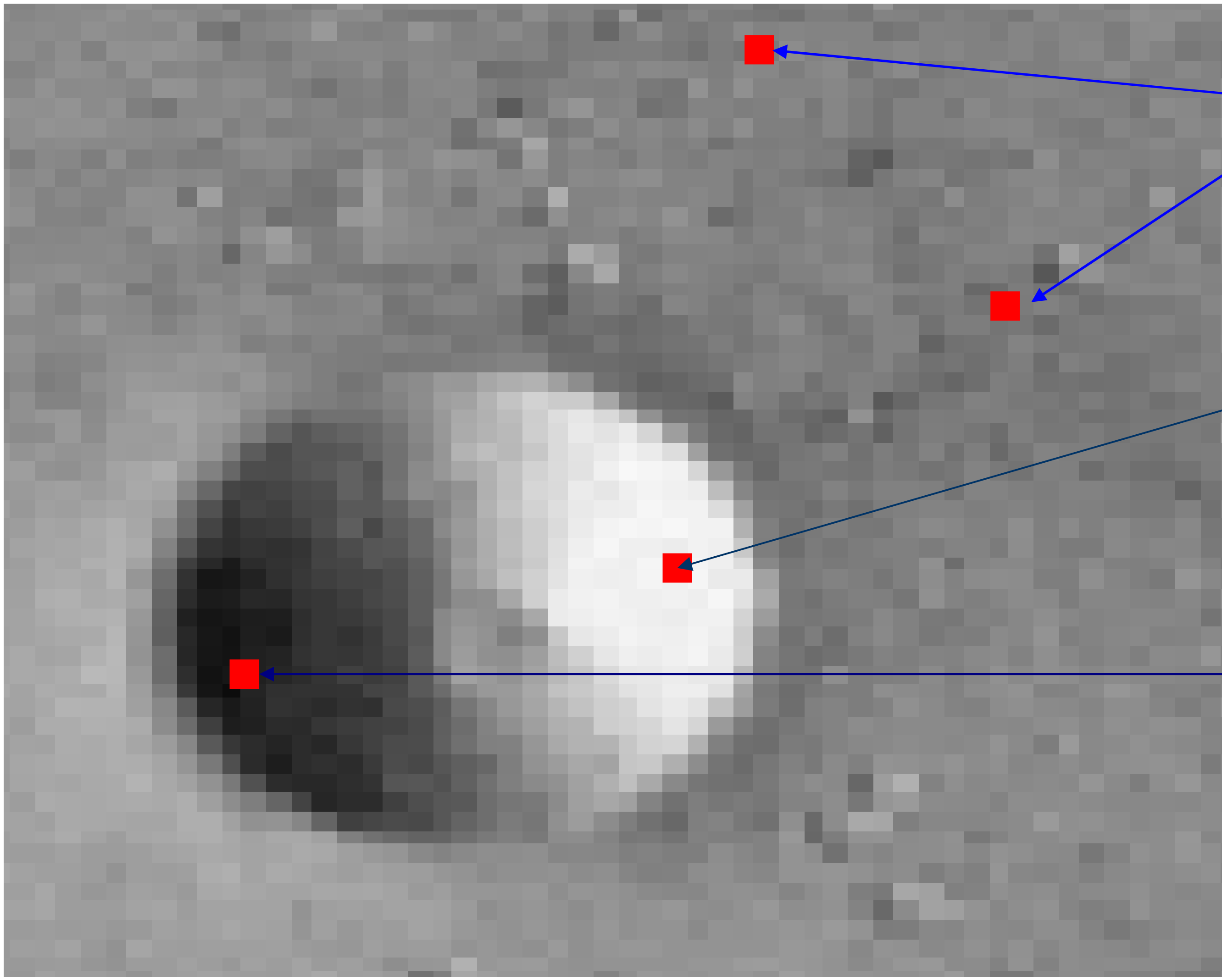
Outline

1. **Tight Frames**
2. **Inpainting**
3. **Impulse Noise Removal**
4. **High Resolution Image Reconstruction**
5. **Video Enhancement**
6. **Extensions**
7. **Convergence Analysis**
8. **Combining PDE and Tightframe**

Preamble

A very brief introduction to image denoising and image deblurring

What is (gray-scale) image?



pixels

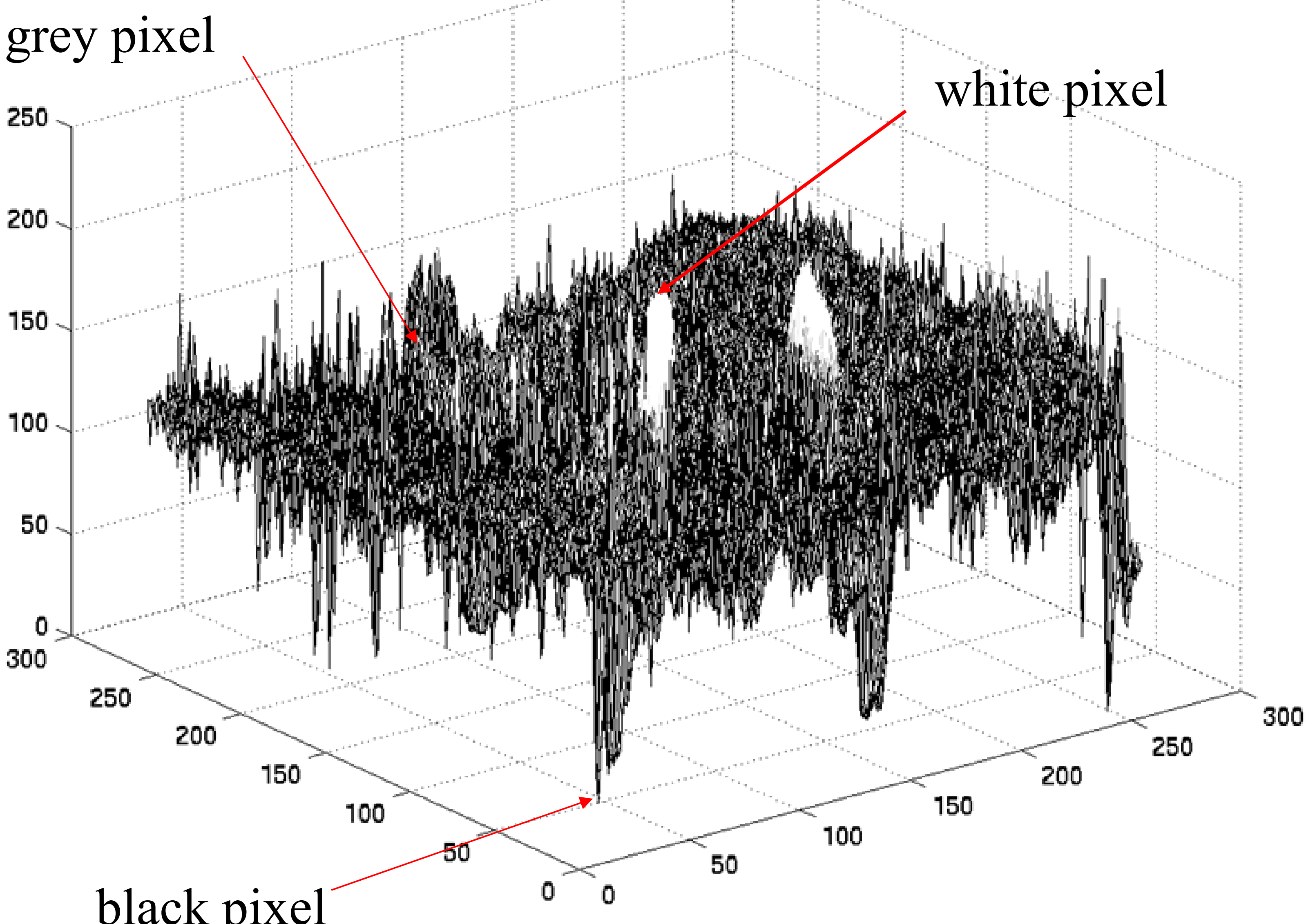
completely
white = 255

completely
black = 0

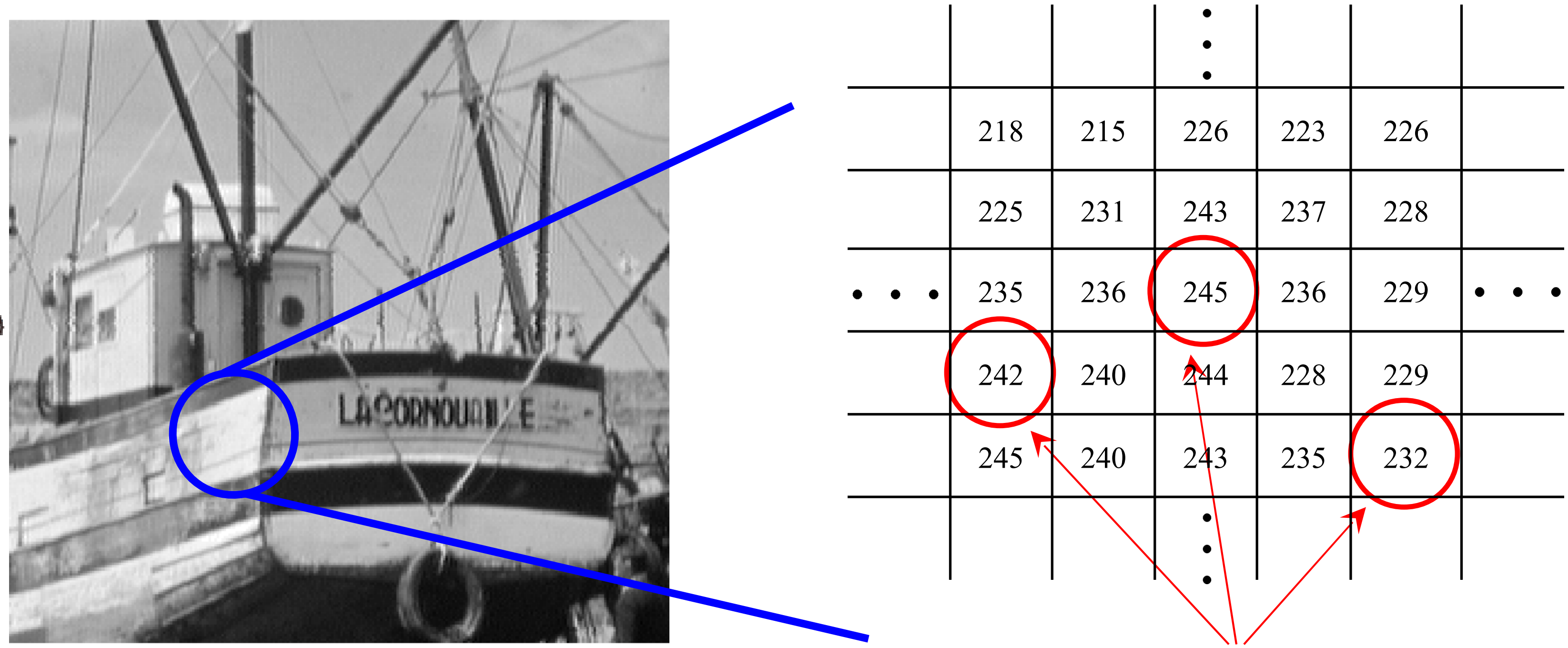
8-bit/pixel

The moon as seen by human

The moon as seen by the computer



What is (gray-scale) image?



pixel values
between 0 and 255

1000-by-1000 image = 1000-by-1000 matrix

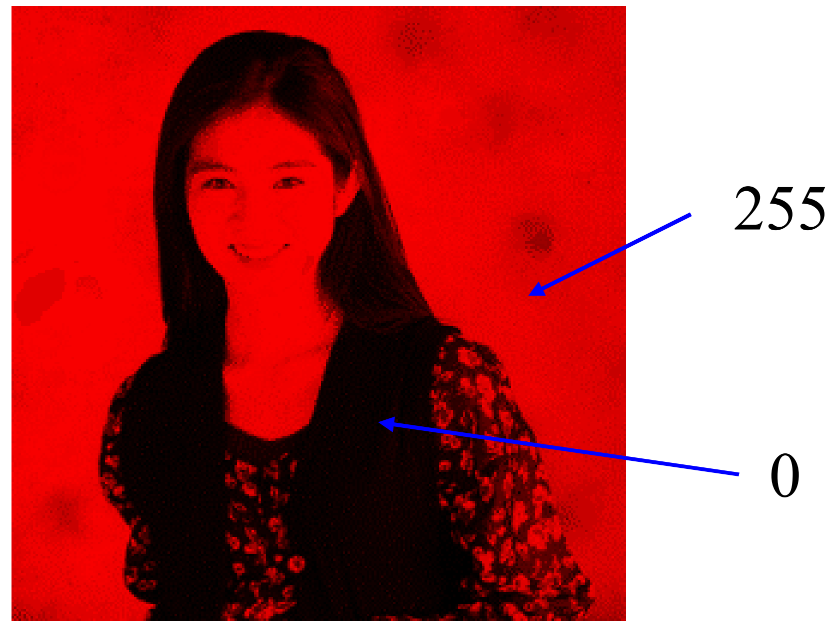
pixel-value = matrix-entry

Concatenate into a 1000^2-vector

Color Images

RGB (red, green and blue channels) 24-bit color:

color image



red channel

What is an image?

Images are piecewise-smooth functions with jumps at edges.

≅ constant

jumps

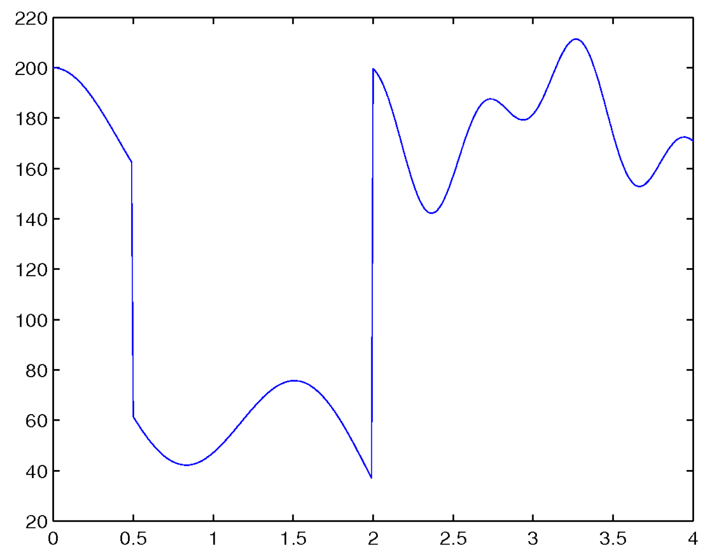
What is an image?

image = smooth parts + jumps

= low frequency components

+ high frequency components

where the high frequency components have big magnitudes.



Denoising:

observed image = **true image** + **noise**

y = **f** + **n**

Denoising:

We want:

1. Data fitting: $\mathbf{n} = \mathbf{y} - \mathbf{f}$ is small, i.e.

$$\min_{\mathbf{n}} \|\mathbf{n}\| = \min_{\mathbf{f}} \|\mathbf{y} - \mathbf{f}\|.$$

2. Regularity: $\mathbf{f}$ is piecewise smooth, i.e.

$$\min_{\mathbf{f}} \|\mathcal{D}\mathbf{f}\|,$$

for some differential operator $\mathcal{D}$.

Variational Method for Denoising

Restoration = Minimization of a cost functional F

$$\min_{\mathbf{f}} F_{\mathbf{y}}(\mathbf{f}) = \min_{\mathbf{f}} [\underbrace{\|\mathbf{f} - \mathbf{y}\|}_{\text{data fitting term}} + \underbrace{\beta \cdot \|\mathcal{D}\mathbf{f}\|}_{\text{regularization term}}],$$

i.e. $\mathbf{f}$ is close to $\mathbf{y}$ and yet $\mathbf{f}$ is smooth.

The number β, which balances the two terms, is called *regularization parameter*.

Variational Method for Denoising

Restoration = Minimization of a cost functional F

Total Variation of Rudin, Osher and Fatemi :

$$\min_{\mathbf{f}} F_{\mathbf{y}}(\mathbf{f}) = \min_{\mathbf{f}} [\underbrace{\frac{1}{2}\|\mathbf{f}-\mathbf{y}\|_{L^2}^2}_{\text{data fitting term}} + \underbrace{\beta \int |\nabla \mathbf{f}|}_{\text{regularization term}}].$$

i.e. $\mathbf{f}$ is close to $\mathbf{y}$ and yet the total variation norm of $\mathbf{f}$ is small.

Variational Method for Denoising

Euler-Lagrange equation:

$$\frac{\partial F_{\mathbf{y}}(\mathbf{f})}{\partial \mathbf{f}} = \mathbf{0} \Longrightarrow \mathbf{f} - \mathbf{y} + \beta \nabla \cdot \left(\frac{\nabla \mathbf{f}}{|\nabla \mathbf{f}|} \right) = \mathbf{0}.$$

Nonlinear PDE—difficult to solve.

Solve until steady state:

$$\frac{\partial \mathbf{f}}{\partial t} = \mathbf{f} - \mathbf{y} + \beta \nabla \cdot \left(\frac{\nabla \mathbf{f}}{|\nabla \mathbf{f}|} \right).$$

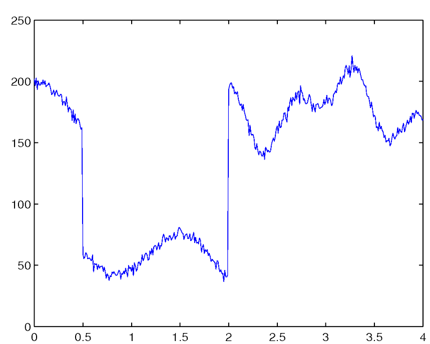
Wavelet Denoising

image = low frequency components

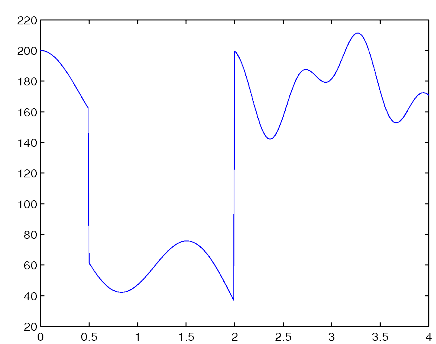
+ high frequency components

where the high frequency components have big magnitudes.

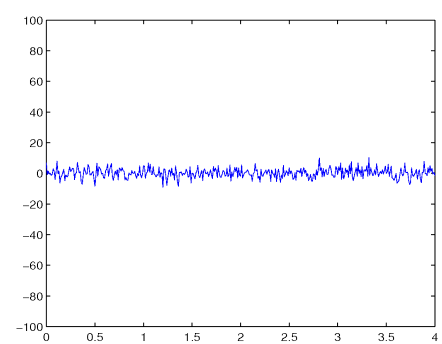
On the contrary, (Gaussian) noise are high frequency components with small magnitudes.



=



+



Wavelet Denoising

A wavelet transform is an orthogonal transform that consists of a low-pass filter H_L and one or more high-pass filters H_H.

More precisely, for any signal $\mathbf{f}$, $H_L\mathbf{f}$ and $H_H\mathbf{f}$ are the low- and high-refrequency parts of $\mathbf{f}$ respectively, and

$$H_L^t H_L + H_H^t H_H = I$$

the *perfect-reconstruction formula*.

Wavelet Denoising

$$
\begin{tikzcd}
 & H_L\mathbf{f} \arrow[r, "H_L^t"] & H_L^tH_L\mathbf{f} \arrow[dr] & \\
\mathbf{f} \arrow[ur, "H_L"] \arrow[dr, "H_H"'] & & & \mathbf{f} \\
 & H_H\mathbf{f} \arrow[r, "H_H^t"] & H_H^tH_H\mathbf{f} \arrow[ur] &
\end{tikzcd}
$$

For image $\mathbf{f}$, $H_L\mathbf{f}$ consists of the smooth parts of $\mathbf{f}$, and $H_H\mathbf{f}$ consists of the jumps of $\mathbf{f}$, and they are big.

For noise $\mathbf{n}$, both $H_L\mathbf{n}$ and $H_H\mathbf{n}$ will be small.

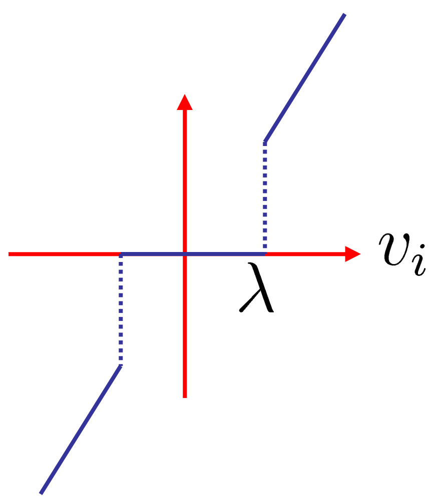
Thresholding

Given $\mathbf{v} = [v_1, \cdots, v_n]$, the hard-thresholding operator $\mathcal{T}_\lambda$ is defined as:

$$\mathcal{T}_\lambda(\mathbf{v}) = [t_{\lambda_1}(v_i), \cdots, t_{\lambda_n}(v_n)]$$

where

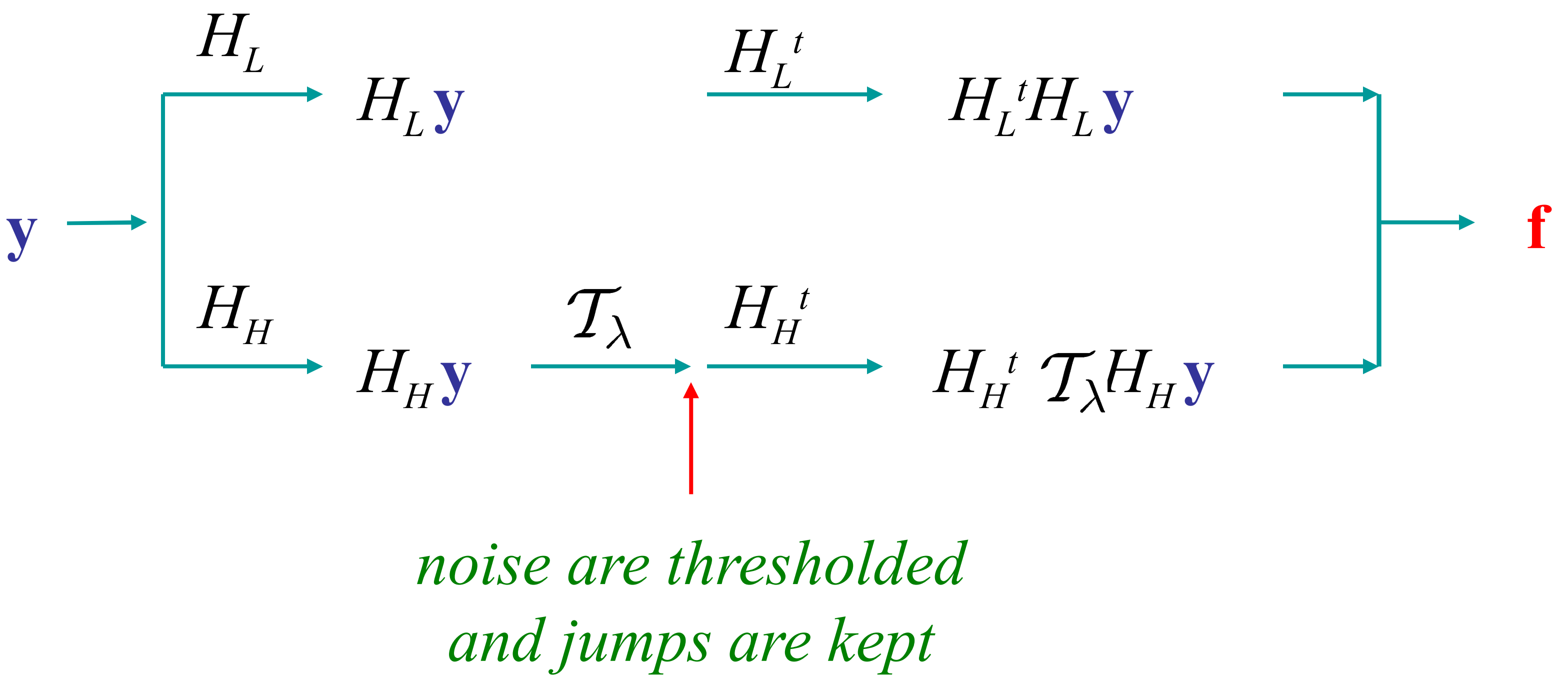
$$t_{\lambda_i}(v_i) \equiv \begin{cases} v_i, & \text{if } |v_i| > \lambda_i, \\ 0, & \text{if } |v_i| \leq \lambda_i. \end{cases}$$



25

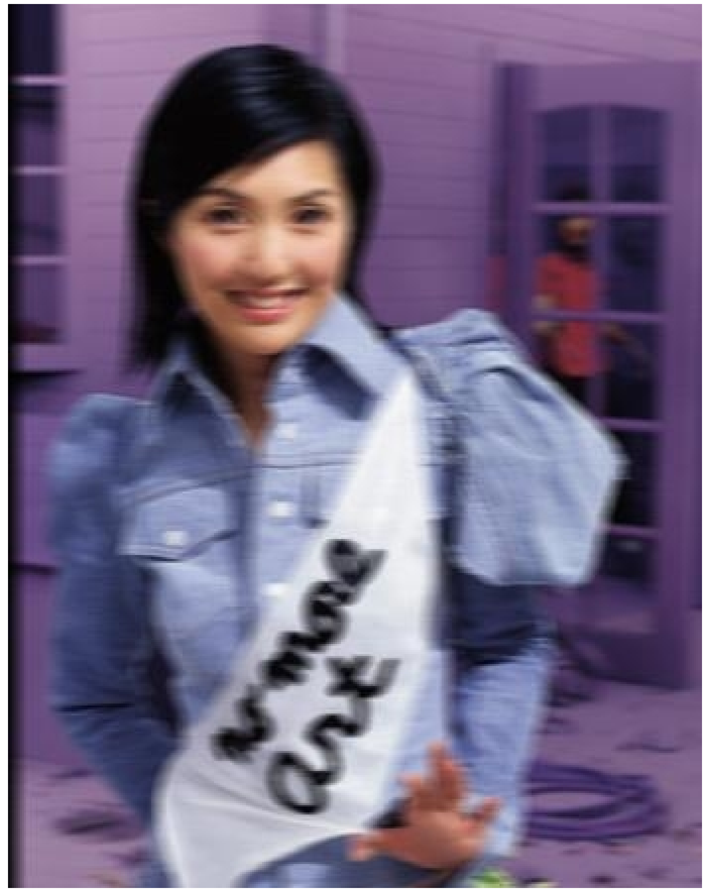
Wavelet Denoising

Given $\mathbf{y} = \mathbf{f} + \mathbf{n}$, we decompose $\mathbf{y}$ into low- and high-frequency parts and we threshold the high-frequency parts. Then we reconstruct $\mathbf{f}$.



Deblurring

image due to motion to the right.



Motion blur

true image

Observed image **y** obtained from true image **f** as follows:

y=

$$\mathbf{y}(i) = \mathbf{f}(i) + \mathbf{f}(i+1) + \mathbf{f}(i+2) + \cdots + \mathbf{f}(i+k)$$

Thus

$$\mathbf{y}(i) = \sum_t \mathbf{f}(i-t).$$

More generally, we have

$$\mathbf{y}(i) = \sum_t b(t)\mathbf{f}(i-t).$$

In matrix terminology, this is a matrix equation:

blurring matrix

$$\boxed{\mathcal{B}\mathbf{f} = \mathbf{y}}$$

To obtain the true image **f**:

$$\boxed{\mathbf{f} = \mathcal{B}^{-1}\mathbf{y}}$$

For linear blur, $\mathcal{B}$ is block diagonal Toeplitz matrix:

$$\mathcal{B} = \begin{bmatrix} L & & & & & \\ & L & & & \mathbf{0} & \\ & & L & & & \\ & & & L & & \\ & \mathbf{0} & & & \mathrm{e} & \\ & & & & & L \end{bmatrix}_{4{,}000{,}000 \times 4{,}000{,}000}$$

for picture with 2000-by-2000 pixels

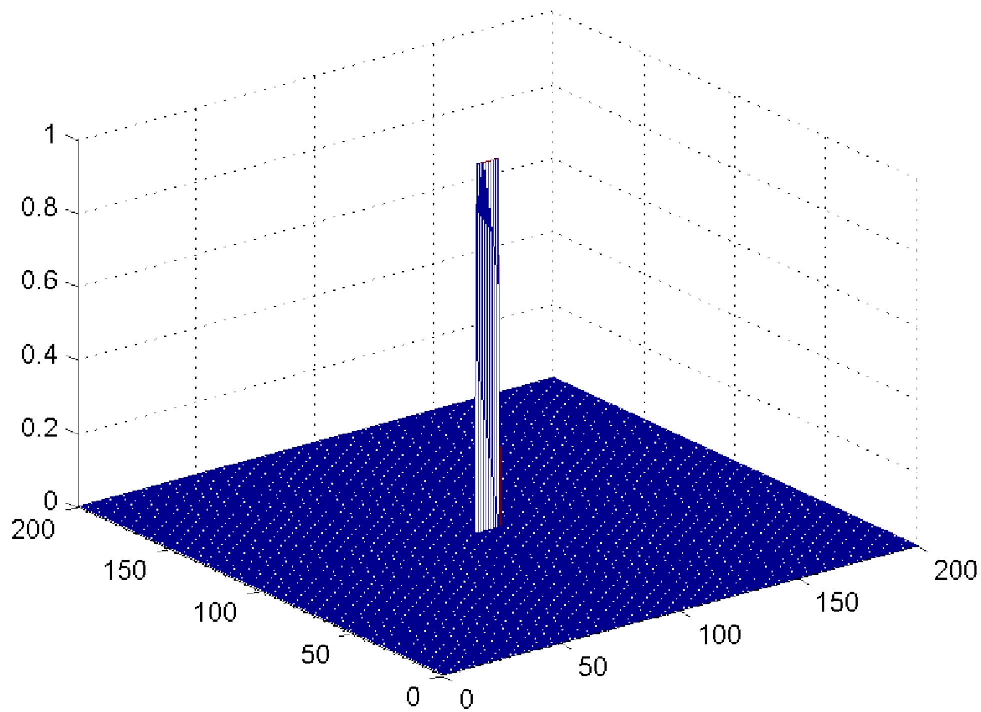
with L a lower-Δ Toeplitz matrix:

$$L = \begin{bmatrix} \mathbf{1} & & & & & & & \\ \mathbf{1} & \mathbf{1} & & & & & & \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & & & & \mathbf{0} & \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & & & & \\ & \cdot & \cdot & \cdot & \cdot & & & \\ & & \cdot & \cdot & \cdot & \cdot & & \\ & \mathbf{0} & & \cdot & \cdot & \cdot & \cdot & \\ & & & & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \end{bmatrix}_{2000 \times 2000}$$

Point-spread Function

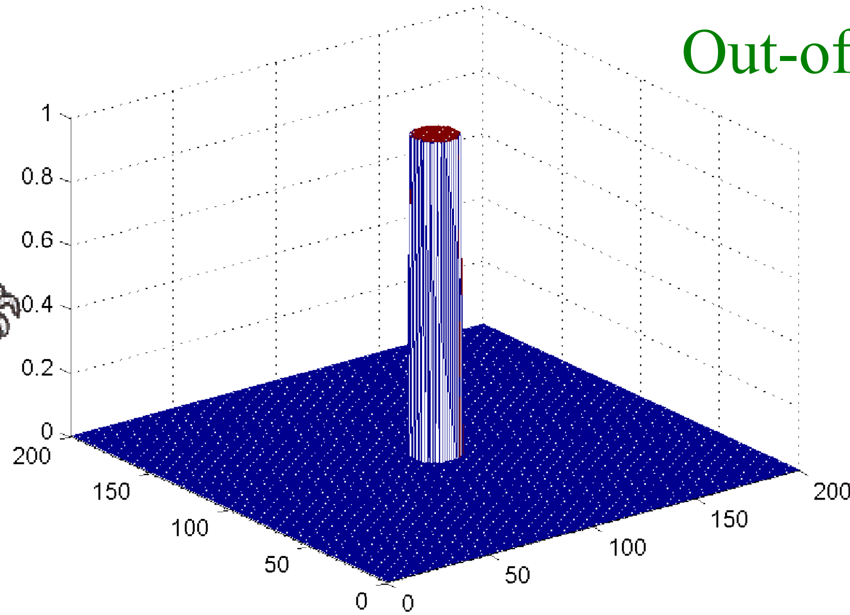
For general blur, $\mathcal{B}$ is block-Toeplitz-Toeplitz-block, hence determined by the middle row.

Reshape the middle row into an n-by-n matrix and display it as a 2D function or an image:

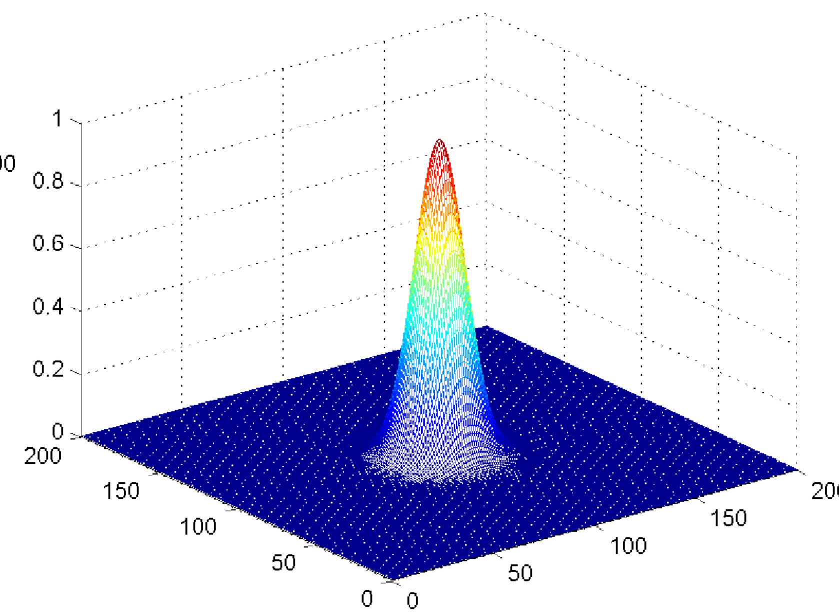


Point-spread function for motion blur

The point-spread function tells how every pixel in the image is blurred.



Out-of-focus blur



Gaussian blur

Given the point-spread function, we can form the BTTB $\mathcal{B}$ blurring matrix accordingly.

Blur Model

observed image = blurred image + noise

$$\mathbf{y} = \mathcal{B}\mathbf{f} + \mathbf{n}$$

Variational Method for Deblurring

Restoration = Minimization of a cost functional F

$$\min_{\mathbf{f}} F_{\mathbf{y}}(\mathbf{f}) = \min_{\mathbf{f}}[\ \underbrace{\|\mathcal{B}\mathbf{f} - \mathbf{y}\|}_{\text{data fitting term}} \ + \ \underbrace{\beta \cdot \|\mathcal{D}\mathbf{f}\|}_{\text{regularization term}}\ ],$$

i.e. $\mathcal{B}\mathbf{f}$ is close to $\mathbf{y}$ and yet $\mathbf{f}$ is smooth.

From observed $\mathbf{y}$ to reconstructed $\mathbf{f}$:

Variational Approach

$$\min_{\mathbf{f}} F_{\mathbf{y}}(\mathbf{f})$$

or

$$\frac{\partial F_{\mathbf{y}}(\mathbf{f})}{\partial \mathbf{f}} = \mathbf{0}$$

Tight-Frame Approach

$\mathcal{A}\mathbf{z}$ and $\mathcal{A}^t\mathbf{z}$,

and

thresholding

$$\boxed{\text{where } \mathcal{A}^t\mathcal{A} = \mathcal{I}}$$

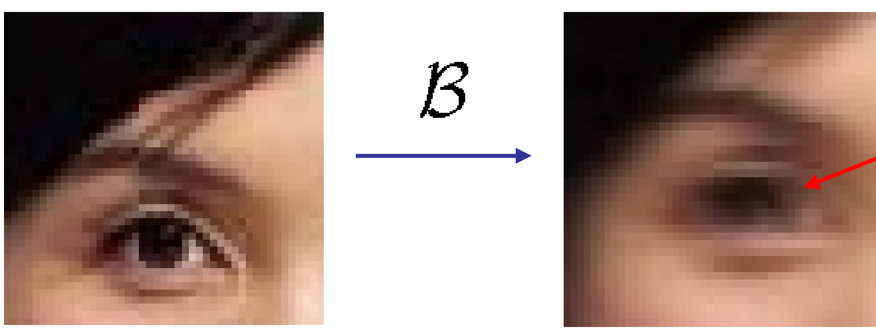
Blurs are Low-Pass Filters

Blurring has the form:

$$\mathbf{y}(i) = \sum_t b(t)\mathbf{f}(i-t).$$

where $b(t) \geq 0$. Thus blurring is a weighted averaging.

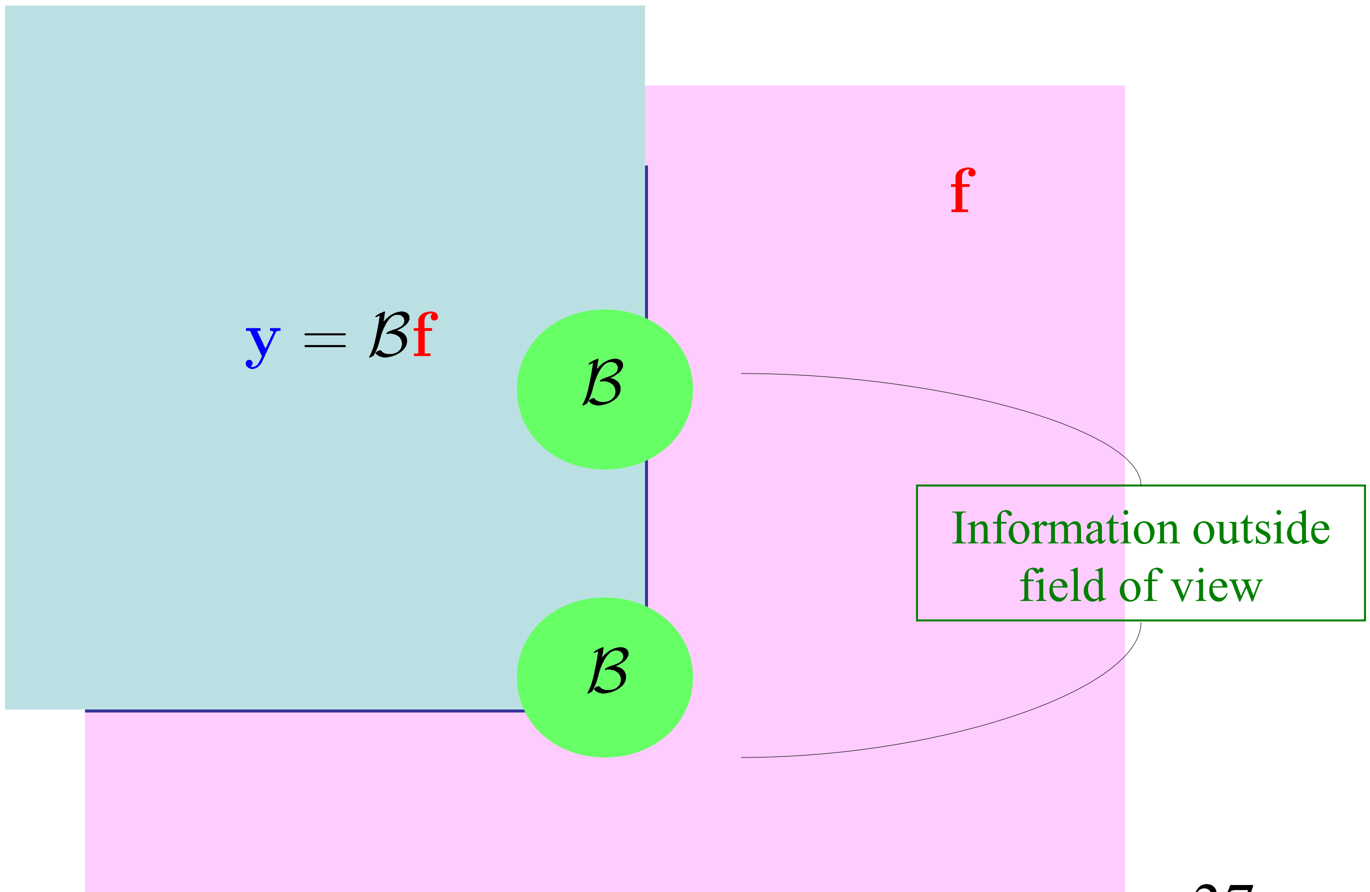
High frequencies in $\mathbf{f}$, e.g. edges, got averaged out or smoothed, and $\mathcal{B}\mathbf{f}$ consists mainly of low frequencies.



edges are smoothed

Boundary Conditions

The blurred image $\mathbf{y} = \mathcal{B}\mathbf{f}$ involves information of the true image $\mathbf{f}$ outside the field of view.



Boundary Conditions

Periodic Boundary Condition (Gonzalez and Woods, 93):

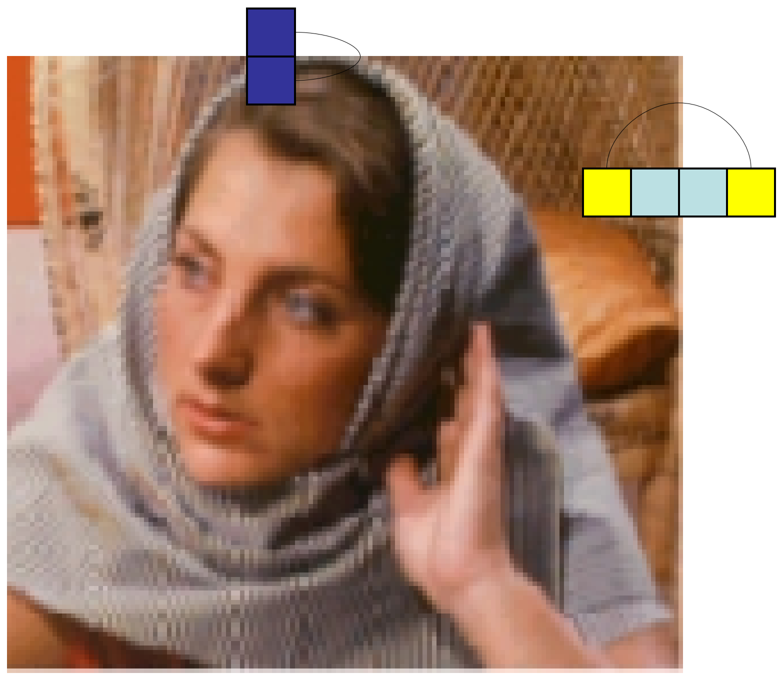
Assume data are periodic near the boundary.

Dirichlet (Zero) BC (Boo and Bose, IJIST 97):

Assume data zeros outside boundary

Neumann Boundary Condition (Ng, C. & Tang (SISC 00)):

Assume data are reflective near boundary.



Anti-reflexive Boundary Condition (Serra, SISC (2003))

Assume data are negated and reflected near boundary.

Outline

1. Tight Frames
2. Inpainting
3. Impulse Noise Removal
4. High Resolution Image Reconstruction
5. Video Enhancement
6. Extensions
7. Convergence Analysis
8. Combining PDE and Tightframe

Tight Frames

- Definition (Duffin and Schaeffer, Trans. AMS, 1952): Let $\mathcal{X} \subset L^2(\mathbb{R})$ be countable. $\mathcal{X}$ is a tight frame for $L^2(\mathbb{R})$ if

$$\sum_{g \in \mathcal{X}} |\langle f, g \rangle|^2 = \|f\|^2, \quad \forall f \in L^2(\mathbb{R})$$

- This is equivalent to

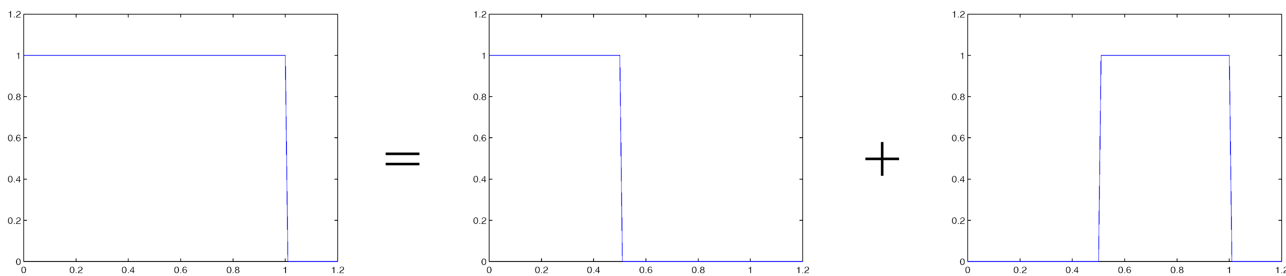
$$f = \sum_{g \in \mathcal{X}} \langle f, g \rangle g, \quad \forall f \in L^2(\mathbb{R})$$

- An orthonormal basis is a tight frame

Construction of Haar Wavelet

Define $\phi(x) = 1$ for $x \in [0,1]$, and 0 otherwise. Then we have the refinement equation:

$$\phi(x) = 1 \cdot \phi(2x) + 1 \cdot \phi(2x-1).$$



The transfer function is

$$h_0(\omega) = \frac{1}{2} + \frac{1}{2}e^{i\omega} = e^{-i\omega/2}\cos(\omega/2)$$

Construction of Haar Wavelet

If we define

$$h_1(\omega) = \frac{1}{2} + \frac{-1}{2} e^{i\omega} = e^{-i\omega/2} \sin(\omega/2)$$

We have

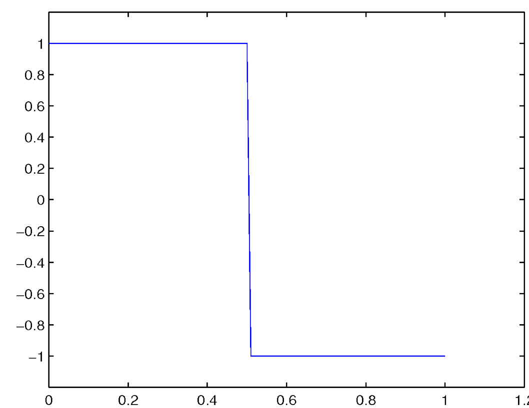
$$\sum_{i=0}^{1} h_i(\omega)\overline{h_i(\omega)} = 1 \quad \text{and} \quad \sum_{i=0}^{1} h_i(\omega)\overline{h_i(\omega+\pi)} = 0.$$

which gives the *perfect reconstruction property* of Haar wavelet.

Construction of Haar Wavelet

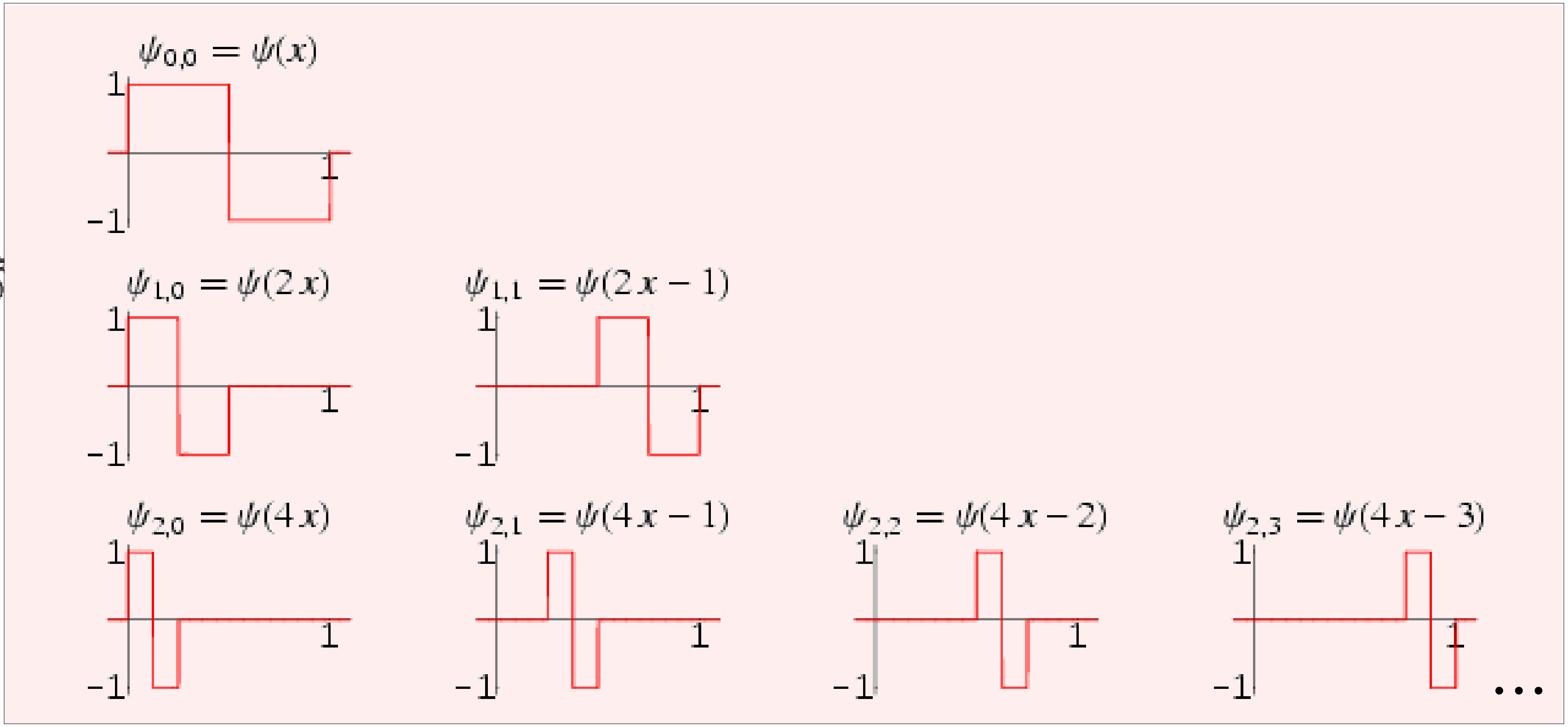
Accordingly, define

$$\begin{aligned}\psi(x) &= 1 \cdot \phi(2x) + (-1) \cdot \phi(2x-1) \\ &= \begin{cases} 1 & 0 \le x < \frac{1}{2} \\ -1 & \frac{1}{2} < x \le 1 \end{cases}\end{aligned}$$



The set $\mathcal{X} = \{\psi_{jk} \mid \psi_{jk}(x) = 2^{j/2}\psi(2^j x - k)\}$ is a wavelet system and hence a tight frame.

Haar Function

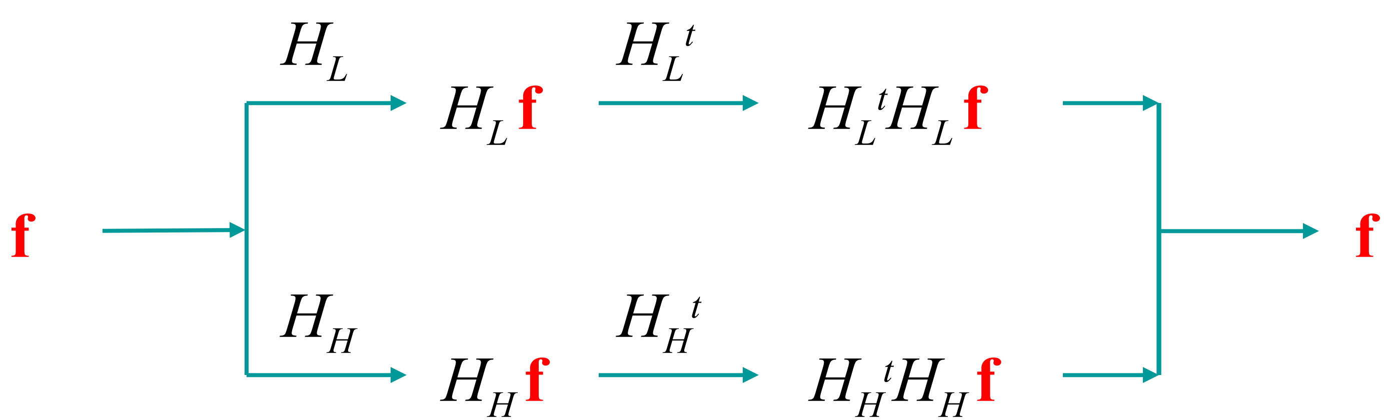


The collection of all these functions is the Haar wavelet system and an orthonormal basis for $L_2[0,1]$.

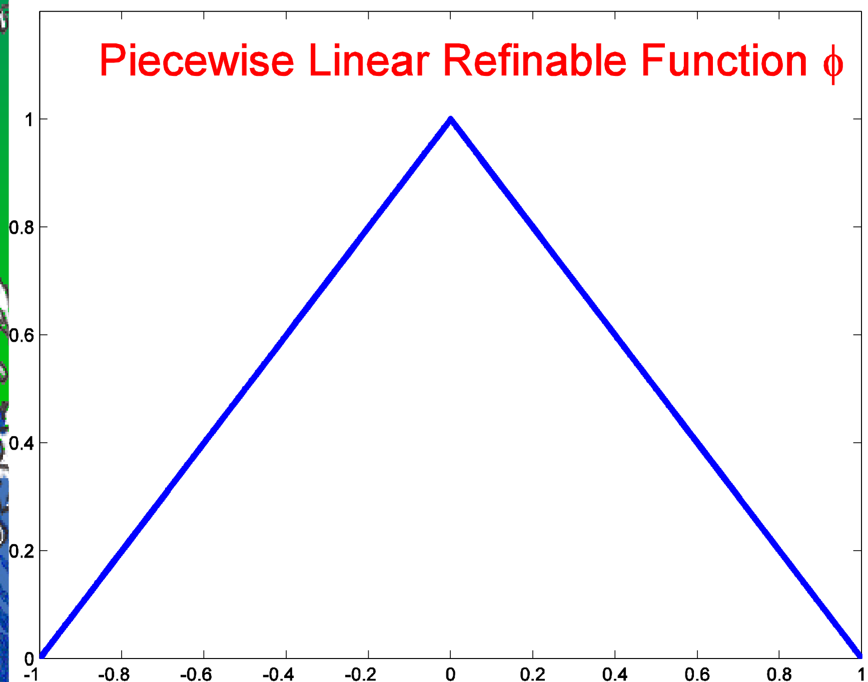
Haar Wavelet Filters

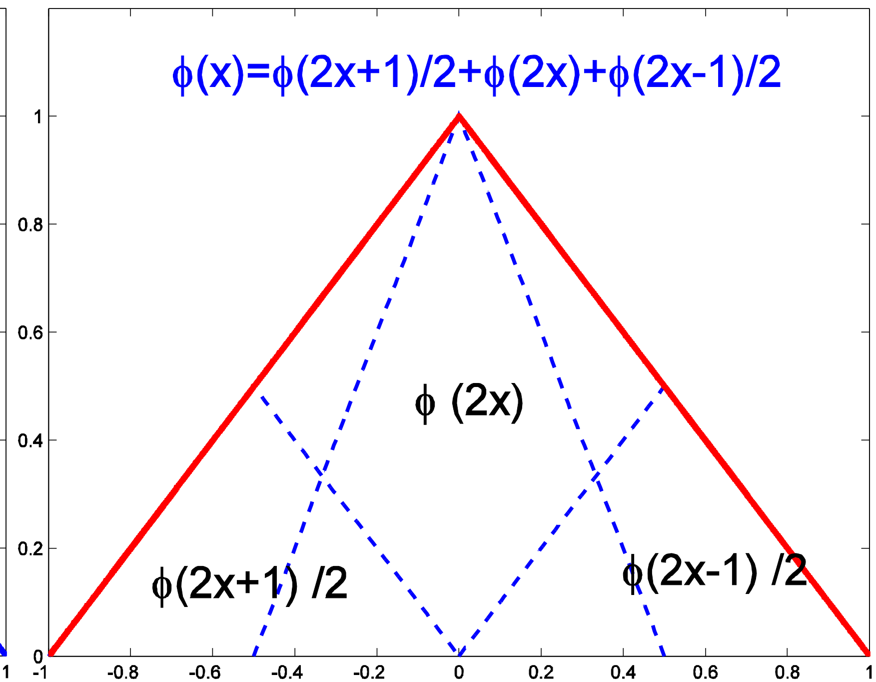
Haar's filters are: $\boxed{\frac{1}{2}[1, 1] \text{ and } \frac{1}{2}[1, -1]}$

If we use $\frac{1}{2}[1, 1]$ and $\frac{1}{2}[1, -1]$ to construct the low-pass filter H_L and the high-pass filters H_H respectively, then $H_L^t H_L + H_H^t H_H = I$, the *perfect-reconstruction formula*.



Piecewise Linear Tight Frame





$$\phi(x) = \frac{1}{2}\phi(2x+1) + 1 \cdot \phi(2x) + \frac{1}{2}\phi(2x-1)$$

$$h_0(\omega) = \frac{1}{4}e^{i\omega} + \frac{1}{2} + \frac{1}{4}e^{-i\omega}$$

Piecewise Linear Tight Frame

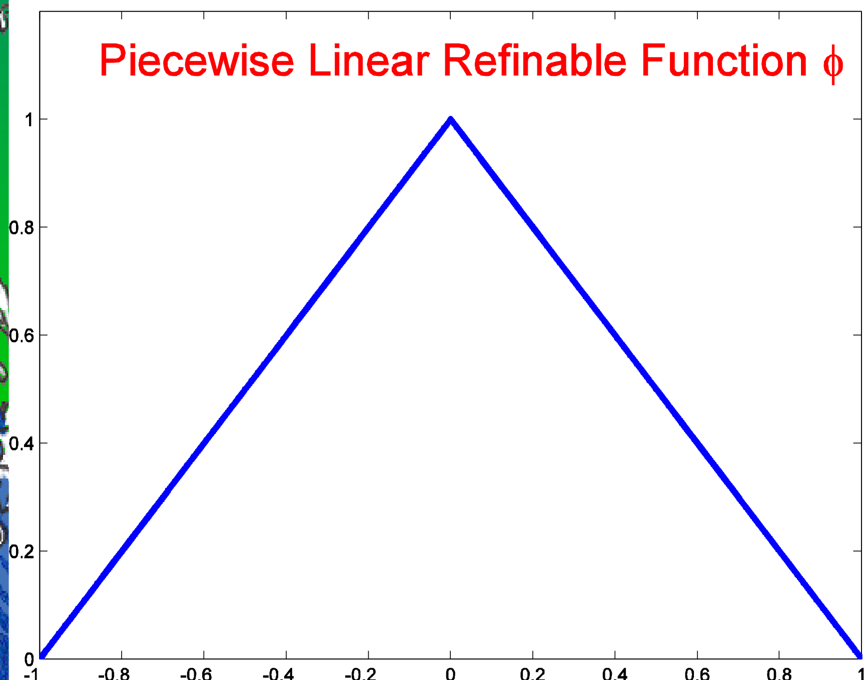
Let

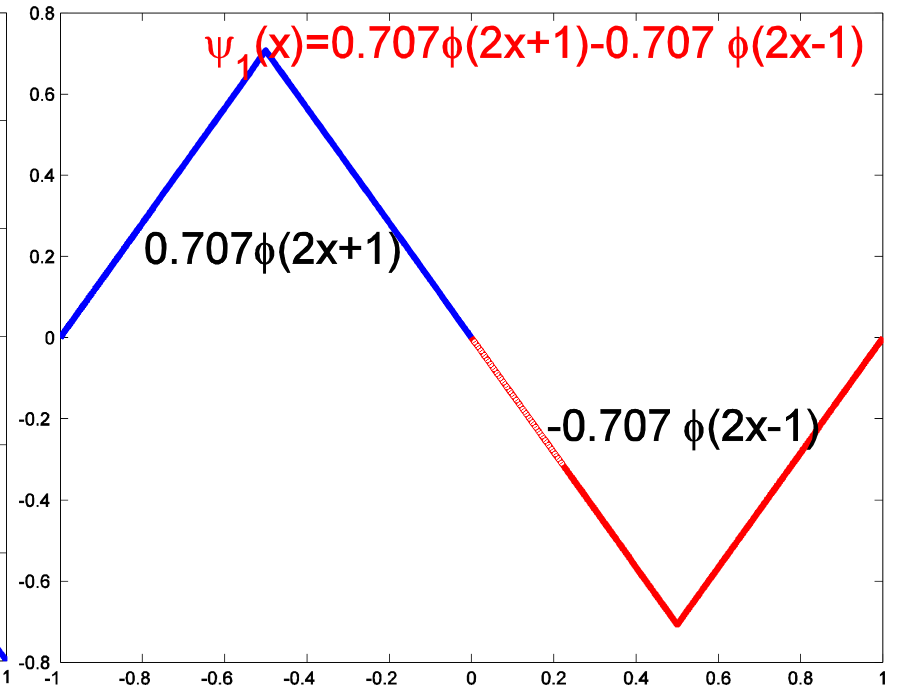
$$\begin{aligned} h_1(\omega) &= \frac{\sqrt{2}}{4}e^{i\omega} - \frac{\sqrt{2}}{4}e^{-i\omega} \\ h_2(\omega) &= -\frac{1}{4}e^{i\omega} + \frac{1}{2} - \frac{1}{4}e^{-i\omega} \end{aligned}$$

Then

$$\sum_{i=0}^{2} h_i(\omega)\overline{h_i(\omega)} = 1 \quad \text{and} \quad \sum_{i=0}^{2} h_i(\omega)\overline{h_i(\omega+\pi)} = 0.$$

Piecewise Linear Tight Frame

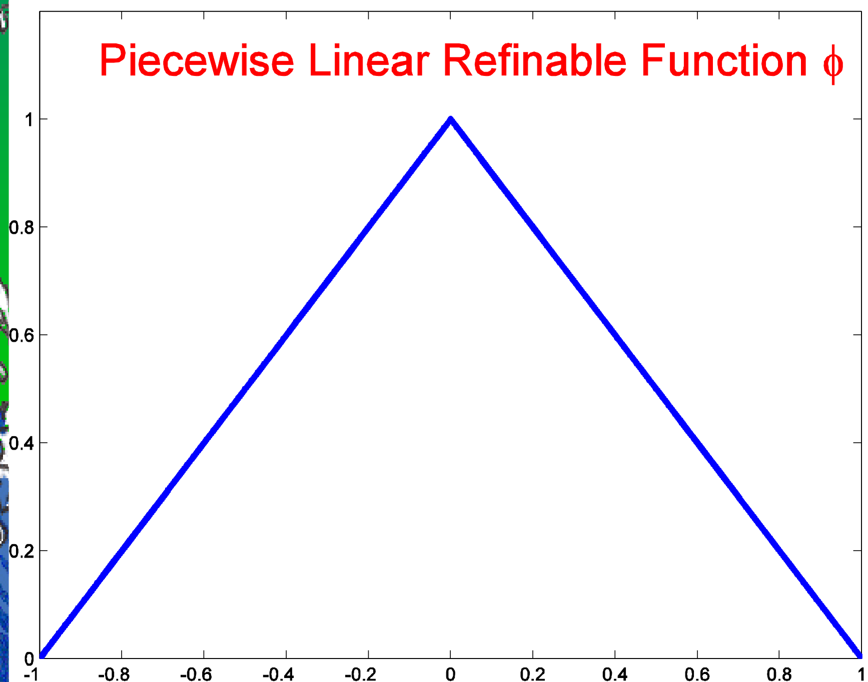


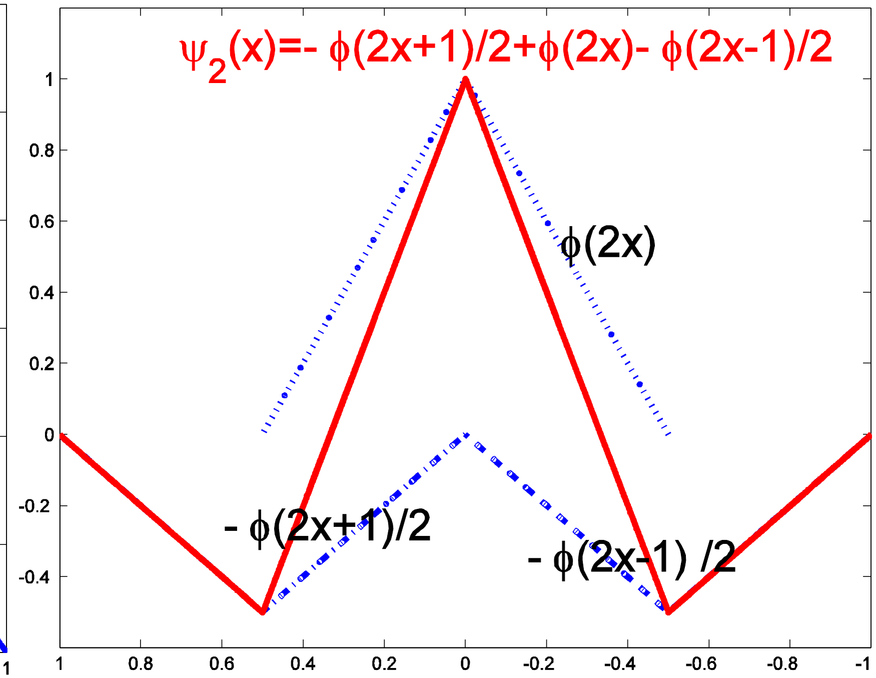


$$h_1(\omega) = \frac{\sqrt{2}}{4}e^{i\omega} - \frac{\sqrt{2}}{4}e^{-i\omega}$$

$$\psi_1(x) = \frac{1}{\sqrt{2}}\phi(2x+1) - \frac{1}{\sqrt{2}}\phi(2x-1)$$

Piecewise Linear Tight Frame





$$h_2(\omega) = -\frac{1}{4}e^{i\omega} + \frac{1}{2} - \frac{1}{4}e^{-i\omega}$$

$$\psi_2(x) = -\frac{1}{2}\phi(2x+1) + 1 \cdot \phi(2x) - \frac{1}{2}\phi(2x-1)$$

Piecewise Linear Tight Frame

The system obtained by dilation and translation:

$$\mathcal{X} = \{2^{k/2}\psi_i(2^k \cdot - j) : k,\, j \in \mathbb{Z}; i = 1, 2\}$$

is the *piecewise linear tight framelet system*.

Given h_0, is it easy to find h_1 and h_2 such that

$$\sum_{i=0}^{2} h_i(\omega)\overline{h_i(\omega)} = 1 \quad \text{and} \quad \sum_{i=0}^{2} h_i(\omega)\overline{h_i(\omega+\pi)} = 0?$$

Unitary Extension Principle

Theorem (Ron-Shen, 97): Let $\phi \in L^2(\mathbb{R})$ be a refinable function whose refinement equation is

$$\widehat{\phi}(2\cdot) = h_0(\cdot)\widehat{\phi}(\cdot),$$

(h_0 is called *refinement mask* or *low-pass filter*.)

Let h_i, $i = 1, \ldots, m$ be highpass filters satisfying

$$\sum_{i=0}^{m} h_i(\omega)\overline{h_i(\omega)} = 1 \quad \text{and} \quad \sum_{i=0}^{m} h_i(\omega)\overline{h_i(\omega+\pi)} = 0.$$

($\{h_i\}_{i=1}^{m}$ are called *framelet masks*.)

Unitary Extension Principle

Define $\Psi := \{\psi_1, \ldots, \psi_m\}$ with

$$\widehat{\psi}_i(2\cdot) = h_i(\cdot)\widehat{\phi}(\cdot).$$

Then $\mathcal{X}(\Psi)$ is a *tight frame* of $L^2(\mathbb{R})$ and $\{\psi_i\}_{i=1}^m$ are called *framelets*.

- Easy to find $\{h_i\}_{i=1}^m$ if

$$|h_0(\cdot)|^2 + |h_0(\cdot + \pi)|^2 \leq 1$$

- Explicit formula for h_i for B-spline tightframes

1D Piecewise Linear Tight Frame

1. Start with linear B-spline ϕ (the hat function).

2. Its Fourier transform is $\widehat{\phi}(\omega) = \frac{\sin^2(\omega/2)}{(\omega/2)^2}$.

3. Define framelets

$$\widehat{\psi}_i(\omega) = \widehat{h}_i(\omega/2)\widehat{\phi}(\omega/2)$$

with framelet masks

$$\widehat{h}_i(\omega) = \sqrt{\binom{2}{i}} \sin^i(\omega/4)\cos^{2-i}(\omega/4),$$

for $1 \leq i \leq 2$.

1D Piecewise Linear Tight Frame

4. The system

$$\mathcal{X} = \{2^{k/2}\psi_i(2^k \cdot -j) : k,\, j \in \mathbb{Z}; i = 1, 2\}$$

is the *piecewise linear tight framelet system*.

5. The filters are:

$$\begin{aligned} h_0 &= \frac{1}{4}[1, 2, 1], \\ h_1 &= \frac{\sqrt{2}}{4}[1, 0, -1], \\ h_2 &= \frac{1}{4}[-1, 2, -1]. \end{aligned}$$

Matrix Representation

To apply a filter onto a signal is equivalent to pre-multiply the signal vector by a Toeplitz matrix.

E.g. $h_0 = \frac{1}{4}[1, 2, 1]$ corresponds to

$$h_0 \longleftrightarrow H_0 = \frac{1}{4}\begin{bmatrix} 2 & 1 & & & 0 \\ 1 & 2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 2 & 1 \\ 0 & & & 1 & 2 \end{bmatrix}.$$

Matrix Representation

Usually, one uses reflexive boundary condition to minimize boundary artifacts—resulting a Toeplitz-like matrix.

E.g. $h_0 = \frac{1}{4}[1, 2, 1]$ corresponds to

$$h_0 \longleftrightarrow H_0 = \frac{1}{4} \begin{bmatrix} 3 & 1 & & & 0 \\ 1 & 2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 2 & 1 \\ 0 & & & 1 & 3 \end{bmatrix}.$$

C. & Jin, *Iterative Toeplitz Solvers*, SIAM (2007)

Analysis and Synthesis Operators

The tight-frame transform is obtained by:

$$\mathcal{A} = \begin{bmatrix} H_0 \\ H_1 \\ H_2 \end{bmatrix} \quad \leftrightarrow \quad \text{analysis operator}$$

$$\mathcal{A}^* = \begin{bmatrix} {H_0}^* & {H_1}^* & {H_2}^* \end{bmatrix} \quad \leftrightarrow \quad \text{synthesis operator}$$

$$\mathcal{A}\mathbf{f} = \begin{bmatrix} H_0 \\ H_1 \\ H_2 \end{bmatrix} \mathbf{f} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} \quad \leftrightarrow \quad \text{framelet coefficients}$$

Important Observation

$\boxed{\textit{Tight Frames = Redundant Bases}}$

If

$$\mathcal{A} = \text{analysis operator}$$

and

$$\mathcal{A}^* = \text{synthesis operator},$$

then

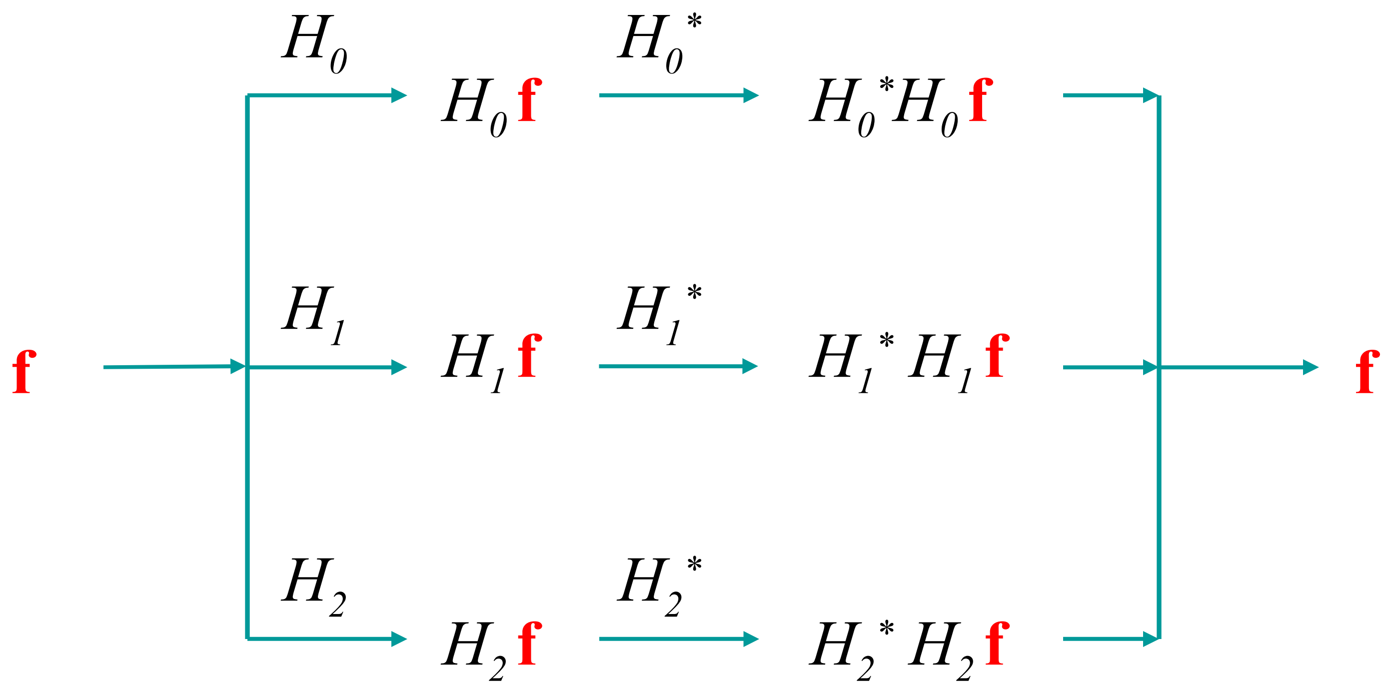
$$\mathcal{A}^*\mathcal{A} = \mathcal{I},$$

but

$$\mathcal{A}\mathcal{A}^* \neq \mathcal{I}.$$

Perfect Reconstruction Formula

$$\mathcal{A}^*\mathcal{A} = \mathcal{I} \Leftrightarrow H_0^* H_0 + H_1^* H_1 + H_2^* H_2 = I$$



$\boxed{\text{No need for } \mathcal{A}\mathcal{A}^* = \mathcal{I}}$

Multi-level Decomposition without Down-sampling

For piecewise linear tight frame:

$h_0 = [\frac{1}{4}, \frac{1}{2}, \frac{1}{4}]$, $h_1 = [-\frac{\sqrt{2}}{4}, 0, \frac{\sqrt{2}}{4}]$, $h_2 = [-\frac{1}{4}, \frac{1}{2}, -\frac{1}{4}]$.

Define h_0 at level ℓ is

$$h_0^{(\ell)} = \left[\frac{1}{4}, \underbrace{0, \cdots, 0}_{2^{(\ell-1)}-1}, \frac{1}{2}, \underbrace{0, \cdots, 0}_{2^{(\ell-1)}-1}, \frac{1}{4}\right].$$

The masks $h_1^{(\ell)}$ and $h_2^{(\ell)}$ can be given similarly.

Multi-level Decomposition without Down-sampling

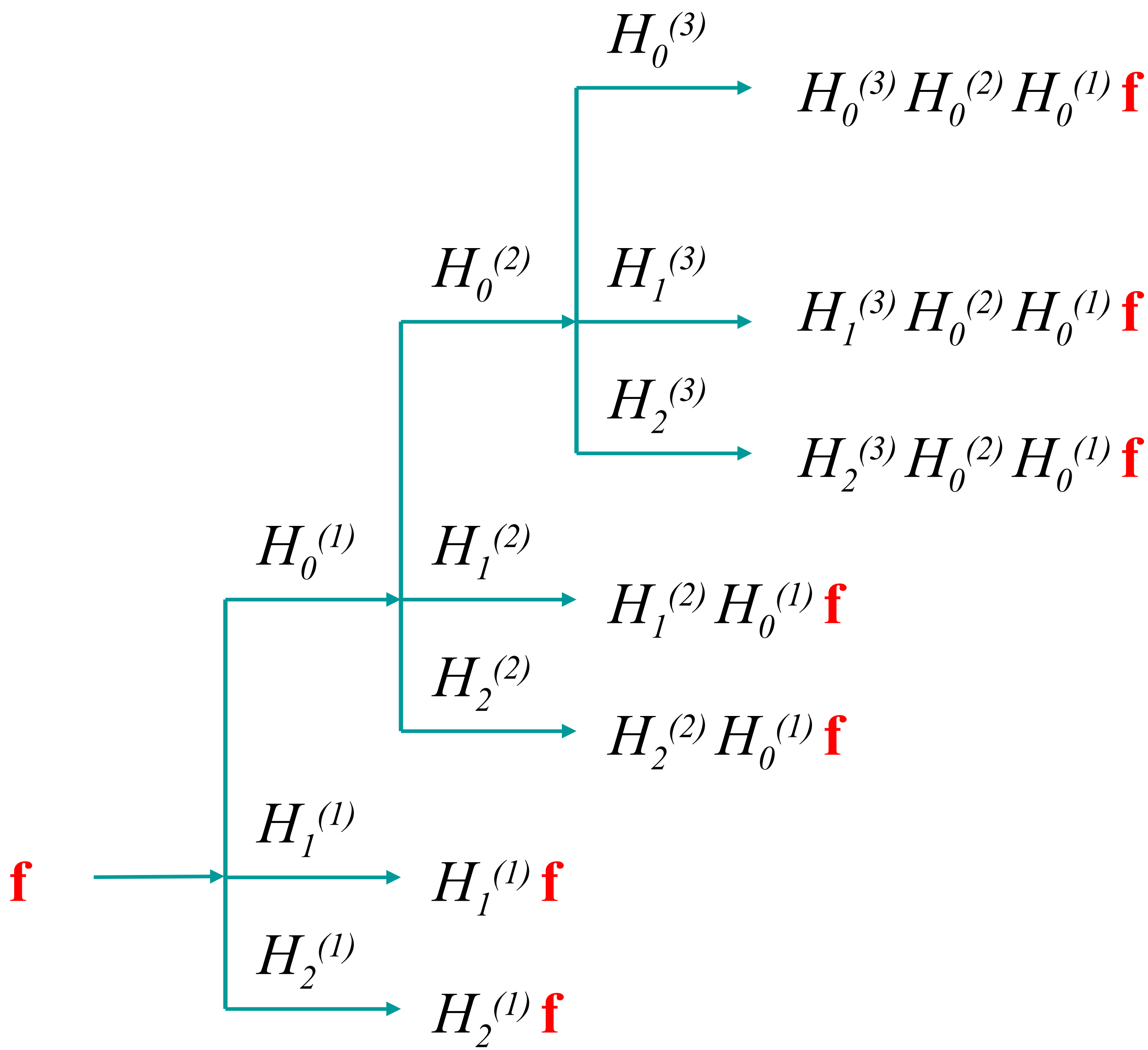
Let $H_i^{(\ell)}$ be the matrix corresponding to $h_i^{(\ell)}$. Then

$$\mathcal{A} = \begin{bmatrix} \prod_{\ell=0}^{L-1} H_0^{(L-\ell)} \\ \hline H_1^{(L)} \prod_{\ell=1}^{L-1} H_0^{(L-\ell)} \\ H_2^{(L)} \prod_{\ell=1}^{L-1} H_0^{(L-\ell)} \\ \vdots \\ H_1^{(1)} \\ H_2^{(1)} \end{bmatrix} \equiv \begin{bmatrix} \mathcal{A}_L \\ \hline \\ \mathcal{A}_H \\ \\ \end{bmatrix}.$$

and

$$\mathcal{A}^* \mathcal{A} = \mathcal{A}_L^* \mathcal{A}_L + \mathcal{A}_H^* \mathcal{A}_H = I.$$

Multi-level Decomposition



2D Tight Frame

We use tensor product to produce a tight framelet system in $\mathcal{L}^2(\mathbb{R}^2)$:

- framelet masks:

$$\widehat{h}_{i,j}(\omega_1, \omega_2) = \widehat{h}_i(\omega_1)\widehat{h}_j(\omega_2)$$

- filters:

$$h_{ij} = h_i^t h_j$$

for $i, j = 0, 1, 2$.

2D Piecewise Linear Framelets

$$\frac{1}{16}\begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix} \quad \frac{\sqrt{2}}{16}\begin{bmatrix}1&0&-1\\2&0&-2\\1&0&-1\end{bmatrix} \quad \frac{1}{16}\begin{bmatrix}-1&2&-1\\-2&4&-2\\-1&2&-1\end{bmatrix}$$

$$\frac{\sqrt{2}}{16}\begin{bmatrix}1&2&1\\0&0&0\\-1&-2&-1\end{bmatrix} \quad \frac{1}{8}\begin{bmatrix}1&0&-1\\0&0&0\\-1&0&1\end{bmatrix} \quad \frac{\sqrt{2}}{16}\begin{bmatrix}-1&2&-1\\0&0&0\\1&-2&1\end{bmatrix}$$

$$\frac{1}{16}\begin{bmatrix}-1&-2&-1\\2&4&2\\-1&-2&-1\end{bmatrix} \quad \frac{\sqrt{2}}{16}\begin{bmatrix}-1&0&1\\2&0&-2\\-1&0&1\end{bmatrix} \quad \frac{1}{16}\begin{bmatrix}1&-2&1\\-2&4&-2\\1&-2&1\end{bmatrix}$$

$$\begin{matrix}H_{0,0} & H_{0,1} & H_{0,2}\\ H_{1,0} & H_{1,1} & H_{1,2}\\ H_{2,0} & H_{2,1} & H_{2,2}\end{matrix}$$

1D Piecewise Cubic Tight Frame

1. Start with cubic B-spline $\widehat{\phi}(\omega) = \frac{\sin^4(\omega/2)}{(\omega/2)^4}$.
2. Define framelets

$$\widehat{\psi}_i(\omega) = \widehat{h}_i(\omega/2)\widehat{\phi}(\omega/2)$$

with framelet masks

$$\widehat{h}_i(\omega) = \sqrt{\binom{4}{i}} \sin^i(\omega/4) \cos^{4-i}(\omega/4),$$

for $1 \le i \le 4$.

1D Piecewise Cubic Tight Frame

3. The system

$$\mathcal{X} = \{2^{k/2}\psi_i(2^k \cdot -j) : k, j \in \mathbb{Z}; i = 1, \ldots, 4\}$$

is the *piecewise cubic tight framelet system*.

4. The filters are: $h_0 = \frac{1}{16}[1, 4, 6, 4, 1]$,

$$h_1 = \frac{1}{8}[1, 2, 0, -2, -1], \qquad h_2 = \frac{\sqrt{6}}{16}[-1, 0, 2, 0, -1],$$

$$h_3 = \frac{1}{8}[-1, 2, 0, -2, 1], \qquad h_4 = \frac{1}{16}[1, -4, 6, -4, 1].$$

5. Again we still have $\mathcal{A}^*\mathcal{A} = \mathcal{I}$.

Tight Frames

- Tight frames — redundant bases
  - preserve the unitary property of the analysis and synthesis operators
  - sacrifice orthogonality and linear independence to get more flexibility
- Robust signal representation—errors in signals can be reduced when represented by a redundant system
- Discrete Fourier transform frames applied successfully to many fields

Spline Framelet Systems

- Spline framelet systems: piecewise linear or cubic tight frames
- Can be constructed from the unitary extension principle of Ron and Shen (JFA, 97)
- Either symmetric or anti-symmetric
- Have small supports for a given smoothness order—good time frequency localization
- Our algorithm works with other tight frames

Outline

1. Tight Frames
2. Inpainting
3. Impulse Noise Removal
4. High Resolution Image Reconstruction
5. Video Enhancement
6. Extensions
7. Convergence Analysis
8. Combining PDE and Tightframes

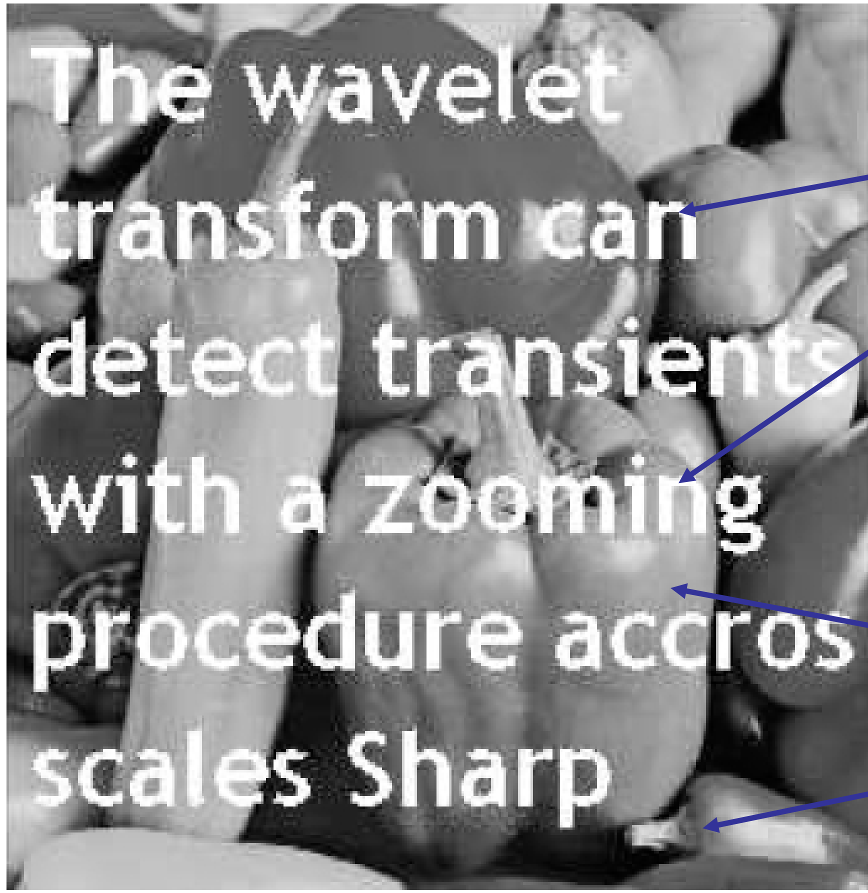
Inpainting

Image Inpainting: filling-in missing pixels based on information in the observed region

Applications: film restoration, text or scratch removal, and digital zooming

Notations

$\Omega = \mathcal{N} \cup \Lambda$



noise set $\mathcal{N}$

Λ *data set*

Fill in data in $\mathcal{N}$ with given data in Λ.

Variational Method

Restoration = Minimization of a cost functional F

Total Variation of Rudin, Osher and Fatemi :

$$\min_{\mathbf{f}} F_{\mathbf{y}}(\mathbf{f}) = \min_{\mathbf{f}}[\ \underbrace{\frac{1}{2}\|\mathbf{f}-\mathbf{y}\|_{L^2}^2}_{\text{data fitting term}} + \underbrace{\beta\int|\nabla\mathbf{f}|}_{\text{regularization term}}\ ].$$

Euler-Lagrange equation gives rise to PDE:

$$\mathbf{f}-\mathbf{y}+\beta\nabla\cdot\left(\frac{\nabla\mathbf{f}}{|\nabla\mathbf{f}|}\right)=\mathbf{0}.$$

Existing Schemes

CS TV Inpainting Model (T. Chan and J. Shen, 2001)

$$F_{\mathbf{y}}(\mathbf{f}) = \frac{\lambda}{2}\int_{\Lambda}(\mathbf{f}-\mathbf{y})^2 + \int_{\Omega}|\nabla \mathbf{f}|$$

The steepest descent equation for the energy is

$$\frac{\partial \mathbf{f}}{\partial t} = \nabla \cdot \left[\frac{\nabla \mathbf{f}}{|\nabla \mathbf{f}|}\right] + \lambda \cdot \chi_{\Lambda}(\mathbf{f}-\mathbf{y}),$$

a diffusion-reaction type of nonlinear PDE.

Existing Schemes

BSCB Model (M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, 2000):

$$\frac{\partial \mathbf{f}}{\partial t} = \nabla^{\perp}\mathbf{f} \cdot \nabla L(\mathbf{f})$$

where $\nabla^{\perp}$ is the 90-degree-rotated copy of the gradient, and $L(\mathbf{f})$ is an operator that evaluates the degree of smoothness.

Propagation of smoothness along the isophotes

Existing Schemes

Curvature-Driven Diffusion Model (T. Chan and J. Shen, 2001):

$$\begin{cases} \dfrac{\partial \mathbf{f}}{\partial t} = \nabla \cdot \left[\frac{d(\kappa)}{|\nabla \mathbf{f}|} \nabla \mathbf{f} \right], & \text{in } \mathcal{N} \\ \mathbf{f} = \mathbf{y}, & \text{in } \Lambda \end{cases}$$

where

$$d(s) = s^p, \quad s > 0, p \geq 1$$

and

$$\kappa = \nabla \cdot \left[\frac{\nabla \mathbf{f}}{|\nabla \mathbf{f}|} \right].$$

Existing Schemes

- Simultaneous Structure and Texture Image Inpainting, (M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, 2003)
- Simultaneous Cartoon and Texture Image Inpainting using Morphological Component Analysis, (Elad, Starck, Querre, and Donoho, 2005)

Basic Idea of Our Framelet Method

Repeat three basic steps in our iteration:

1. Transform current approximation f into the framelet domain via the analysis operator $\mathcal{A}$ to obtain the framelet coefficients $c = \{c_{ij}\}$.

2. Propagate information from Λ into $\mathcal{N}$ by *thresholding* each c_{ij} to obtain $\tilde{c} = \{\tilde{c}_{ij}\}$.

3. Obtain new approximation $\tilde{f}$ on $\mathcal{N}$ by applying the synthesis operator $\mathcal{A}^*$ on $\tilde{c}$.

Explanation

In variational method, there are two terms to minimize:

- data fitting term
- regularization term

$$\min_{\mathbf{f}} F_{\mathbf{y}}(\mathbf{f}) = \min_{\mathbf{f}} [\ \underbrace{\frac{1}{2}\|\mathbf{f}-\mathbf{y}\|_{L^2}^2}_{\text{data fitting term}} + \underbrace{\beta \int |\nabla \mathbf{f}|}_{\text{regularization term}}\ ].$$

Hand-waving Explanation

- In Step 3, the new image is

$$\tilde{f} = \mathcal{A}^* \tilde{c} = \sum_{g \in \mathcal{X}} \tilde{c}_g g.$$

- At the same time, we have

$$\tilde{f} = \mathcal{I} \tilde{f} = \mathcal{A}^* (\mathcal{A} \tilde{f})$$

where

$$\mathcal{A} \tilde{f} = \mathcal{A}(\mathcal{A}^* \tilde{c}) \neq \mathcal{I} \tilde{c} = \tilde{c}.$$

Hand-waving Explanation

- We have two representations of $\tilde{f}$:

$$\tilde{f} = \mathcal{A}^* \tilde{c} = \mathcal{A}^*(\mathcal{A}\tilde{f}).$$

- Frame theory states that $\mathcal{A}\tilde{f}$ has the minimum ℓ_2 norm among all sequences $\{c_g\}_{g \in \mathcal{X}}$ such that

$$\tilde{f} = \sum_{g \in \mathcal{X}} c_g g.$$

- Our process regularizes the new $\tilde{f}$ and gives a representation with minimum ℓ_2.

Tight Frame Algorithm

For $r = 0, 1, \ldots,$ until convergence:

1. Compute $c^{(r)} = \mathcal{A} f^{(r)}$.
2. Threshold $c^{(r)}$ by threshold λ to get $\tilde{c}^{(r)}$.
3. Reconstruct $f^{(r+1)} = \mathcal{A}^* \tilde{c}^{(r)}$.
4. Data fitting: set

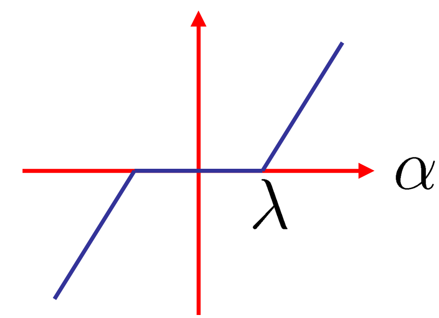
$$[f^{(r+1)}]_p = \begin{cases} [f^{(r+1)}]_p, & p \in \mathcal{N}, \\ [y]_p\,, & p \in \Lambda. \end{cases}$$

Tight Frame Algorithm

The tight frame algorithm is

$$f^{(r+1)} = (\mathcal{I} - \mathcal{P}_\Lambda)\mathcal{A}^*\mathcal{T}_\lambda\mathcal{A}f^{(r)} + \mathcal{P}_\Lambda y$$

where

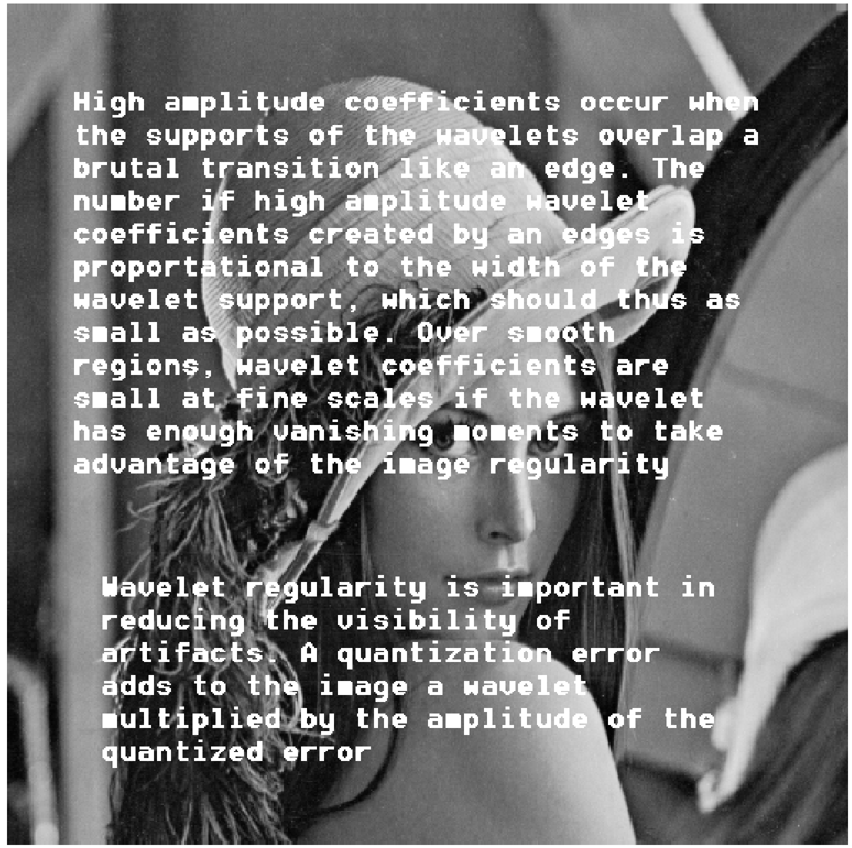


- $\mathcal{P}_\Lambda$: projection onto Λ
- $\mathcal{T}_\lambda$: soft-thresholding operator

$$t_{\lambda_i}(\alpha_i) \equiv \begin{cases} \mathrm{sgn}(\alpha_i)(|\alpha_i| - \lambda_i), & \text{if } |\alpha_i| > \lambda_i, \\ 0, & \text{if } |\alpha_i| \le \lambda_i. \end{cases}$$

85

Numerical Test 1: 512-by-512 Lena



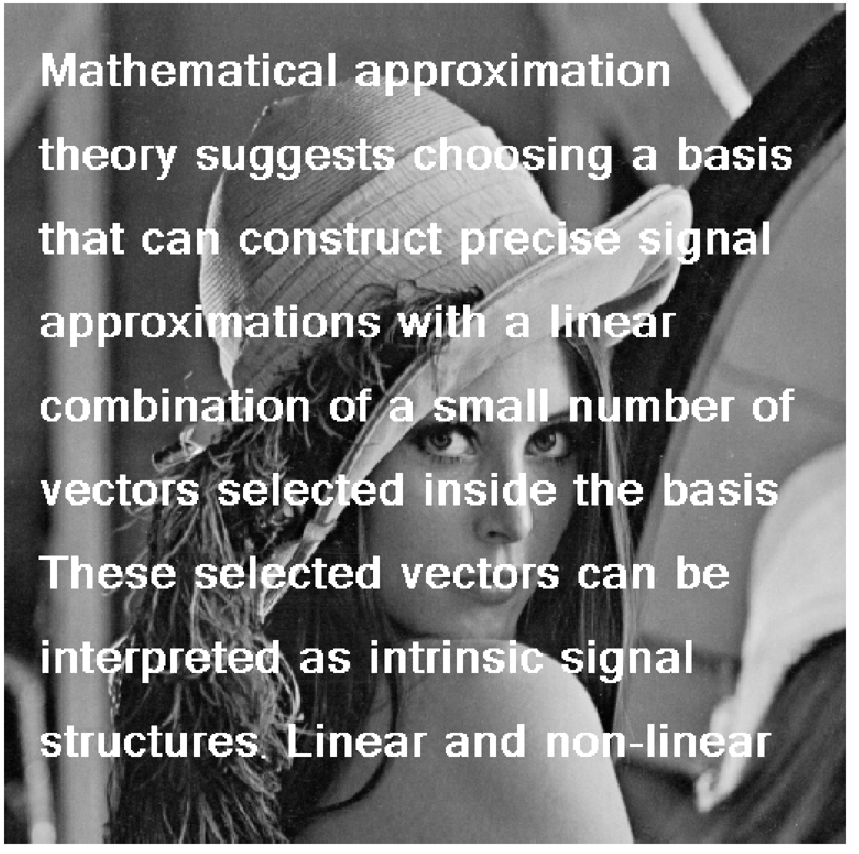
Compared with *curvature-based* program provided by T. Chan and J. Shen (SIAMAM, 02)

Numerical Results 1

Chan-Shen Model
PSNR = 34.78dB
CPU time = 7,862s

Our Algorithm
PSNR = 37.60dB (2.82dB)
CPU time = 521s (15 times)

Numerical Test 2: 512-by-512 Lena



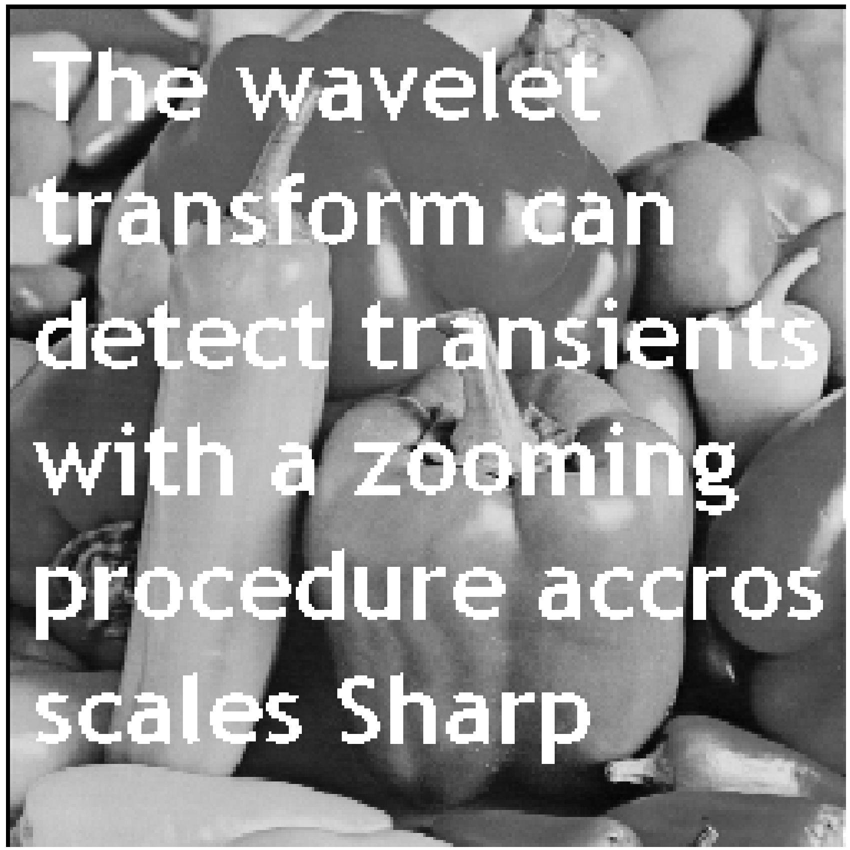
Lena image with bigger text

Numerical Results 2

Chan-Shen Model
PSNR = 34.38dB
CPU time = 7,883s

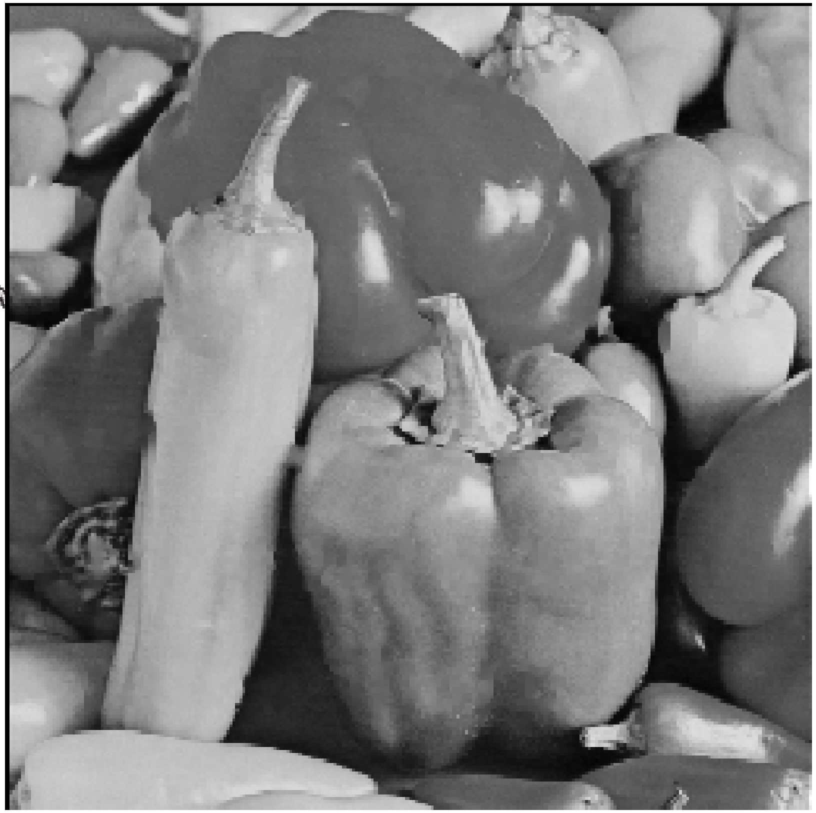
Our Algorithm
PSNR = 36.40dB (2.02dB)
CPU time = 509s (16 times)

Numerical Test 3: 256-by-256 Pepper



Text with even bigger font

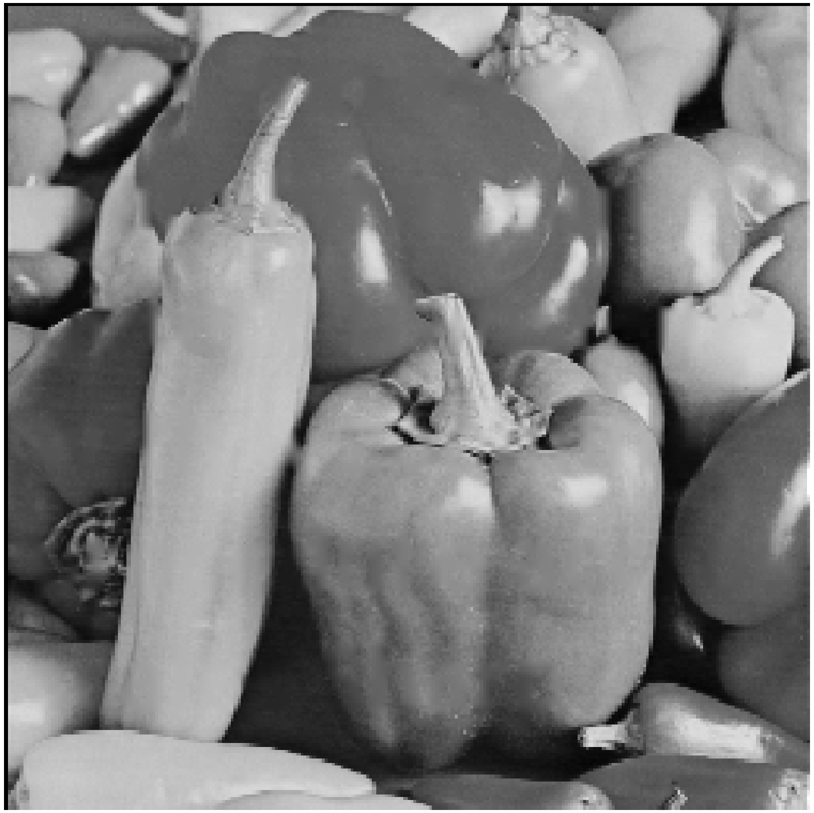
Numerical Results 3



Chan-Shen Model

PSNR = 32.91dB

CPU time = 1,682s



Our Algorithm

PSNR = 34.83dB (1.92dB)

CPU time = 162s (10 times)

Results Up-close

Text Zoomed

Results Up-close

Chan-Shen Model

Our Algorithm

Advantages of Tight Frame Algorithm

- Built-in regularization effect, and exact data fitting

- Framelet coefficients from Λ affect the missing framelet coefficients in a smooth way.

- 2 to 3 dB better and 10-15 times faster than T. Chan and J. Shen's curvature-driven method.

- Convergence proved by convex analysis (Cai, Chan, Shen, ACHA (2008)).

An Equivalent Formulation

The tight frame method

$$f^{(r+1)} = (\mathcal{I} - \mathcal{P}_\Lambda)\mathcal{A}^*\mathcal{T}_\lambda\mathcal{A}f^{(r)} + \mathcal{P}_\Lambda y$$

equivalent to Forward-Backward Splitting:

$$f^{(r+1)} = \mathrm{prox}_{F_1}(f^{(r)} - \nabla F_2(f^{(r)})),$$

for the minimization problem

$$\min_f \{F_1(f) + F_2(f)\},$$

where prox is Moreaus proximity operator.

Minimization Functional

The functional are:

In image domain, f minimizes:

$$\min_{\mathcal{P}_\Lambda f=\mathcal{P}_\Lambda y}\{\min_c\{\tfrac{1}{2}\|\mathcal{A}f - c\|_2^2 + \|\mathrm{diag}(\lambda)c\|_1\}\}.$$

In frequency domain, c minimizes:

$$\min_c\{\tfrac{1}{2}\|\mathcal{P}_\Lambda(\mathcal{A}^*c)-\mathcal{P}_\Lambda y\|_2^2+\tfrac{1}{2}\|(I-\mathcal{A}\mathcal{A}^*)c\|_2^2+\|\mathrm{diag}(\lambda)c\|_1\}.$$

(Cai, C., Shen, ACHA (2008))

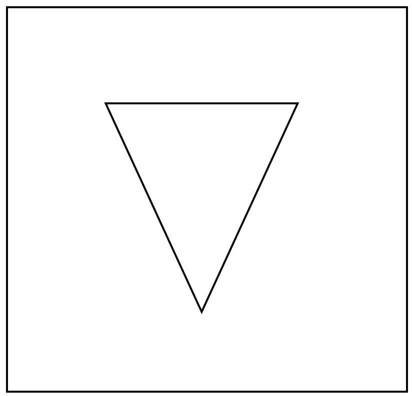
Outline

1. **Tight Frames**
2. **Inpainting**
3. **Impulse Noise Removal**
4. **High Resolution Image Reconstruction**
5. **Video Enhancement**
6. **Extension**
7. **Convergence Analysis**
8. **Combining PDE and Framelets**

Goal

Impulse noise removal

Impulse Noise Model



Original Image (a triangle)

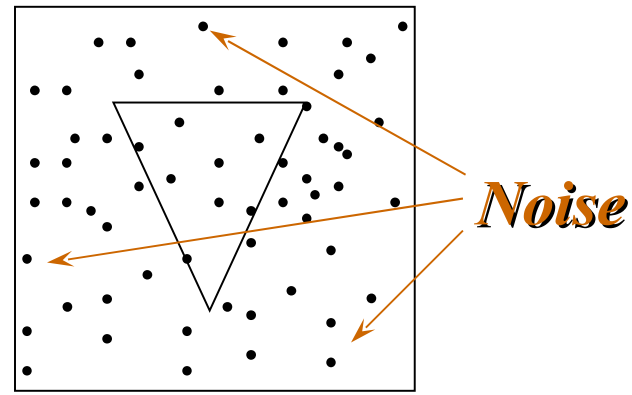


Image corrupted by *Impulse Noise*

Only a number of pixels are corrupted

Impulse Noise Model

Impulse Noise are caused by

- **Malfunctioning pixels** in camera sensors
- **Faulty memory locations** in hardware
- **Transmission** in a noisy channel

Two types of Impulse Noise

I. Salt-and-Pepper Noise

II. Uniformly-Distributed Random Noise

Salt-and-Pepper Noise

$\mathbf{f} = (f_{i,j})$: true image with $f_{i,j} \in [0, 255]$.
$\mathbf{y} = (y_{i,j})$: observed noisy image.

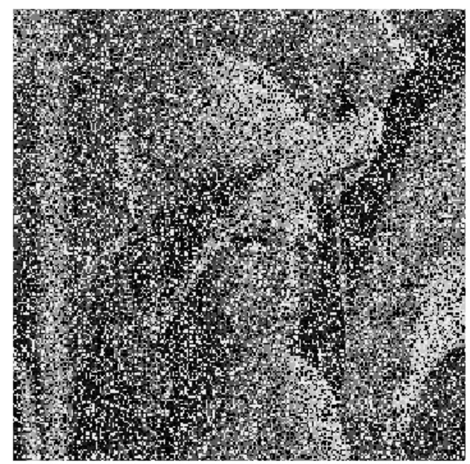
$$y_{i,j} = \begin{cases} 0 & \text{with probability } r/2\%, \\ 255 & \text{with probability } r/2\%, \\ f_{i,j} & \text{with probability } 1 - r\%. \end{cases}$$

Noise level $= r\%$.

Noise-free Image

At 10% Noise

At 30% Noise



At 50% Noise

Random-Valued Impulse Noise

$\mathbf{f} = (f_{i,j})$: true image with $f_{i,j} \in [0, 255]$.
$\mathbf{y} = (y_{i,j})$: observed noisy image.

$$y_{i,j} = \begin{cases} n_{i,j} & \text{with probability } r, \\ f_{i,j} & \text{with probability } 1 - r, \end{cases}$$

where $n_{i,j}$ is randomly distributed in [0,255].

Denoising Schemes

Median Filter

Noisy Image

$y_{i-1,j-1}$	$y_{i-1,j}$	$y_{i-1,j+1}$
$y_{i,j-1}$	$y_{i,j}$	$y_{i,j+1}$
$y_{i+1,j-1}$	$y_{i+1,j}$	$y_{i+1,j+1}$

Restored Image

$y_{i-1,j-1}$	$y_{i-1,j}$	$y_{i-1,j+1}$
$y_{i,j-1}$	y_{i_5}	$y_{i,j+1}$
$y_{i+1,j-1}$	$y_{i+1,j}$	$y_{i+1,j+1}$

Sort

Recovered

$$y_{i_1} \le y_{i_2} \le y_{i_3} \le y_{i_4} \le y_{i_5} \le y_{i_6} \le y_{i_7} \le y_{i_8} \le y_{i_9}$$

Median

30% Salt-and-Pepper Noise

Median filter

Median-type Filters

- Drawback of Median Filter: Every pixel is modified, hence fuzziness and blurring
- Extensions of Median Filters (Median-type Filters):
  - Adaptive Median Filter (*IEEE TIP 1995*)
  - Adaptive Center Weighted Median Filter (2001)
  - Multi-state Median Filters (2001)
  - Filter based on homogeneity info (2003)
  - …
  - Detection statistics (Dong, C., Xu, *IEEE TIP 2007*)

Adaptive Median Filter

Noisy Image

$y_{i-1,j-1}$	$y_{i-1,j}$	$y_{i-1,j+1}$
$y_{i,j-1}$	$y_{i,j}$	$y_{i,j+1}$
$y_{i+1,j-1}$	$y_{i+1,j}$	$y_{i+1,j+1}$

Sort

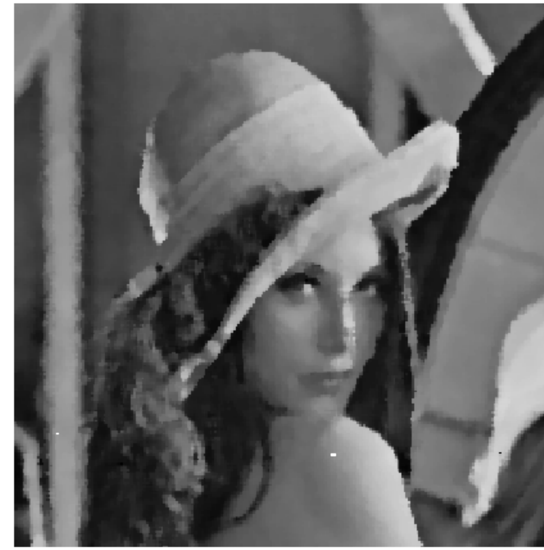
$$y_{i_1} \le y_{i_2} \le y_{i_3} \le y_{i_4} \le y_{i_5} \le y_{i_6} \le y_{i_7} \le y_{i_8} \le y_{i_9}$$

Median

If **Median** $= y_{i_1}$ or y_{i_9}, then increase window size.

30% Salt-and-Pepper Noise

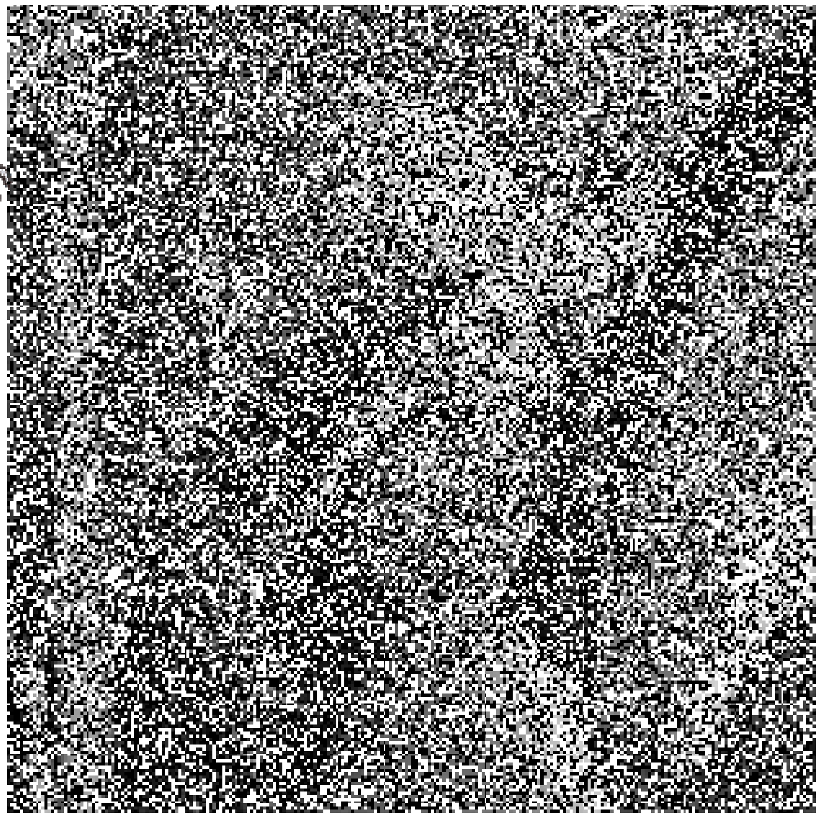
Median Filter



Adaptive Median Filter

But …at 70% Salt-and-Pepper Noise

Adaptive Median Filter



Replacement of noise by median cannot preserve edges

Characteristics of Median-type Filters

Two Steps

1. ***Noise Detection*** (e.g., thresholding)
2. ***Noise Replacement*** (by **Median** or its variants)

Advantages

1. ***Fast***
2. ***Accurate Detection***

Variational Method

Restoration = Minimization of a cost functional F

Total Variation of Rudin, Osher and Fatemi :

$$\min_{\mathbf{f}} F_{\mathbf{y}}(\mathbf{f}) = \min_{\mathbf{f}} [\ \underbrace{\frac{1}{2}\|\mathbf{f} - \mathbf{y}\|_{L^2}^2}_{\text{data fitting term}} + \underbrace{\beta \int |\nabla \mathbf{f}|}_{\text{regularization term}}\ ].$$

Euler-Lagrange equation gives rise to PDE:

$$\mathbf{f} - \mathbf{y} + \beta \nabla \cdot \left(\frac{\nabla \mathbf{f}}{|\nabla \mathbf{f}|} \right) = \mathbf{0}.$$

Edge-preserving for images corrupted by Gaussian noise

l_1 Fitting Term for Impulse Noise:

(Nikolova, *J. Math. Imaging & Vision*, (2004))

$$F(\mathbf{f}) = \sum_{i,j} \Big[\overbrace{|f_{i,j} - y_{i,j}|}^{\ell_1\text{-norm data fitting term}} + \beta \overbrace{\sum_{(m,n)\in\mathcal{V}_{i,j}} \varphi_\alpha(f_{i,j} - f_{m,n})}^{\text{edge-preserving regularization term}} \Big]$$

- ***Non-smooth*** data-fitting term (smooth data left unchanged)
- ***Edge-preserving*** potential function:

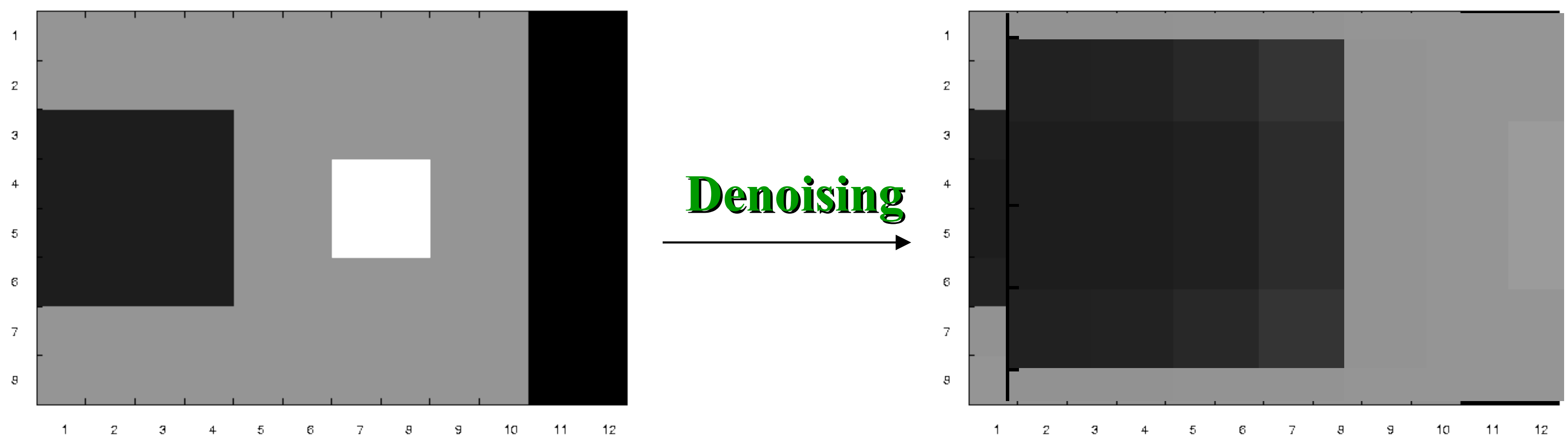
$$\varphi_\alpha(t) = \begin{cases} |t|, & \text{total variation} \\ |t|^\alpha, & 1 < \alpha \le 2, \\ \sqrt{\alpha + t^2}, & \alpha > 0. \end{cases}$$

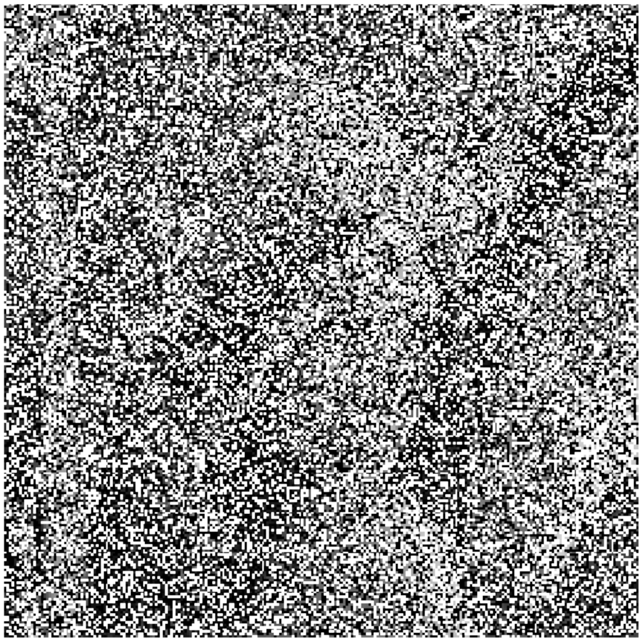
Advantages

- *Insensitive to the amplitude of noise* (l_1-norm), and
- *Preserve edges* when denoising (edge-preserving potential function).

Drawback

- *Continuous method* --- cannot handle noise patches well.
- Some *uncorrupted pixels* at the edges will be *distorted*.





Variational Method

70% ***Salt*-and-*Pepper*** Noise

Two-Phase Method:

Median-type Filter + ***Variational Method***

(Chan, Ho, and Nikolova, *IEEE TIP (2005)*)

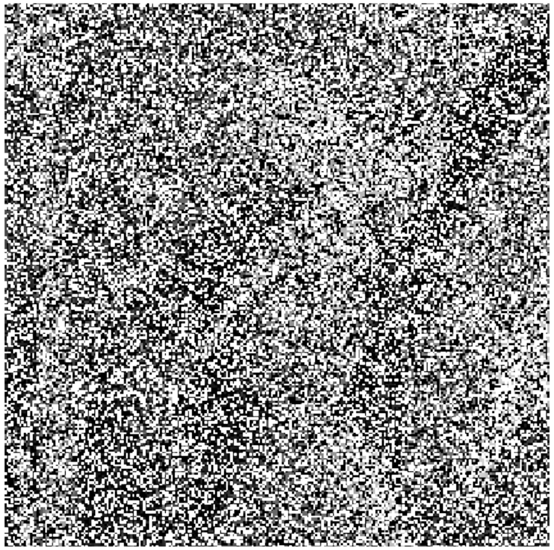
Phase 1: Detect noise candidate set $\mathcal{N}$ by ***Adaptive Median Filter***

Phase 2: Restore pixels in $\mathcal{N}$ by ***Variational Method***

$$\begin{cases} \min_{\mathbf{f}} \sum_{i,j} \left[\ |f_{i,j} - y_{i,j}| + \beta \sum_{(m,n)\in\mathcal{V}_{i,j}} \varphi_\alpha(f_{i,j} - f_{m,n}) \right], \\ \text{subject to } f_{i,j} = y_{i,j} \text{ if } (i,j) \notin \mathcal{N} \end{cases}$$

$$\begin{cases} \min\limits_{\mathbf{f}} \sum\limits_{i,j} \left[|f_{i,j} - y_{i,j}| + \beta \sum\limits_{(m,n)\in\mathcal{V}_{i,j}} \varphi_\alpha(f_{i,j} - f_{m,n}) \right] \\ \text{subject to } f_{i,j} = y_{i,j} \text{ if } (i,j) \notin \mathcal{N} \end{cases}$$

- β: regularization parameter,
  balance the *data-fitting* and *regularity* of $\mathbf{x}$
- $\mathcal{V}_{i,j}$: the four neighbours of (i,j)

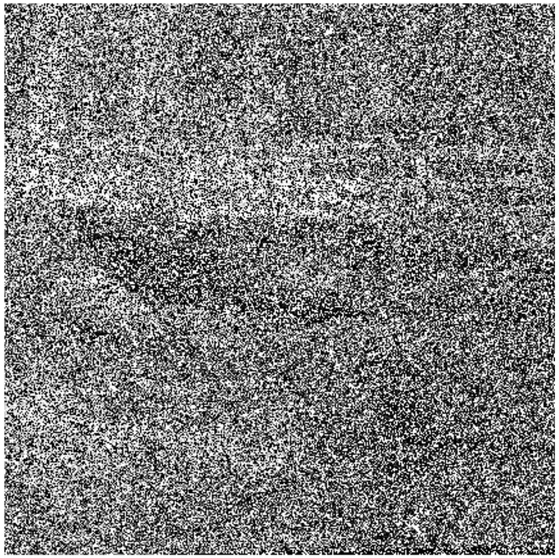


70% *Salt*-*and*-*Pepper* Noise

Adaptive Median Filter

Variational Method

Our Method

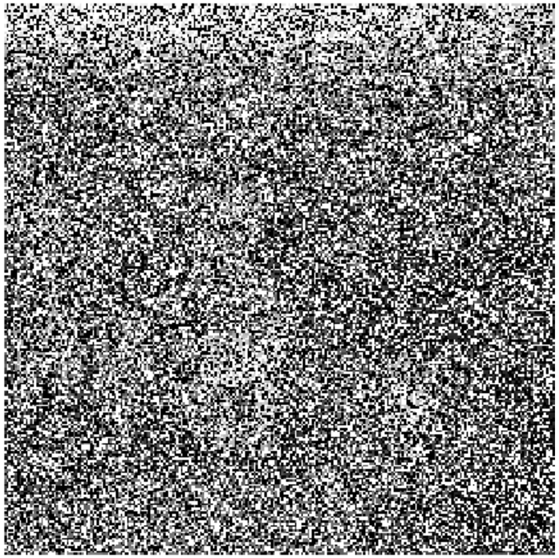


70% *Salt-and-Pepper* Noise

Adaptive Median Filter

Variational Method

Our Method

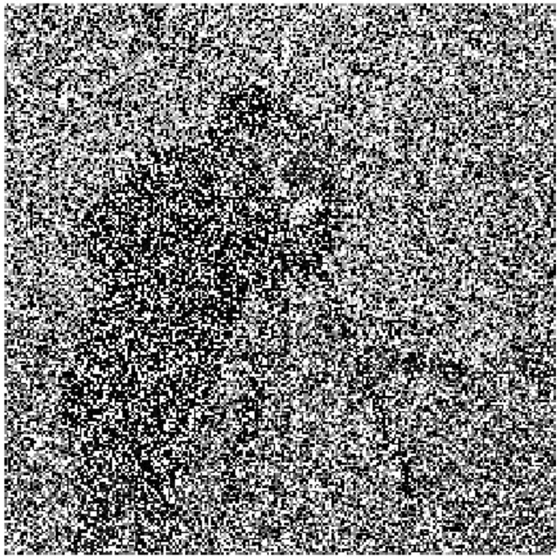


70% *Salt*-*and*-*Pepper* Noise

Adaptive Median Filter

Variational Method

Our Method



70% *Salt*-*and*-*Pepper* Noise

Adaptive Median Filter

Variational Method

Our Method

Performance in PSNR: 70% *Salt-and-Pepper* Noise

Salt-and-Pepper	*Lena*	*Bridge*	*Goldhill*	*Cameraman*
Noisy	6.71	6.78	6.93	6.63
Variational Method	24.64	21.11	23.54	20.69
Adaptive Median Filter	25.73	21.76	21.46	21.38
Our Method	29.26	25.00	26.94	24.91

PSNR increases by 1 dB, error decreases by 10%

Two-Phase Method using Framelets:

Median-type Filter + ***Tight Frame Method***

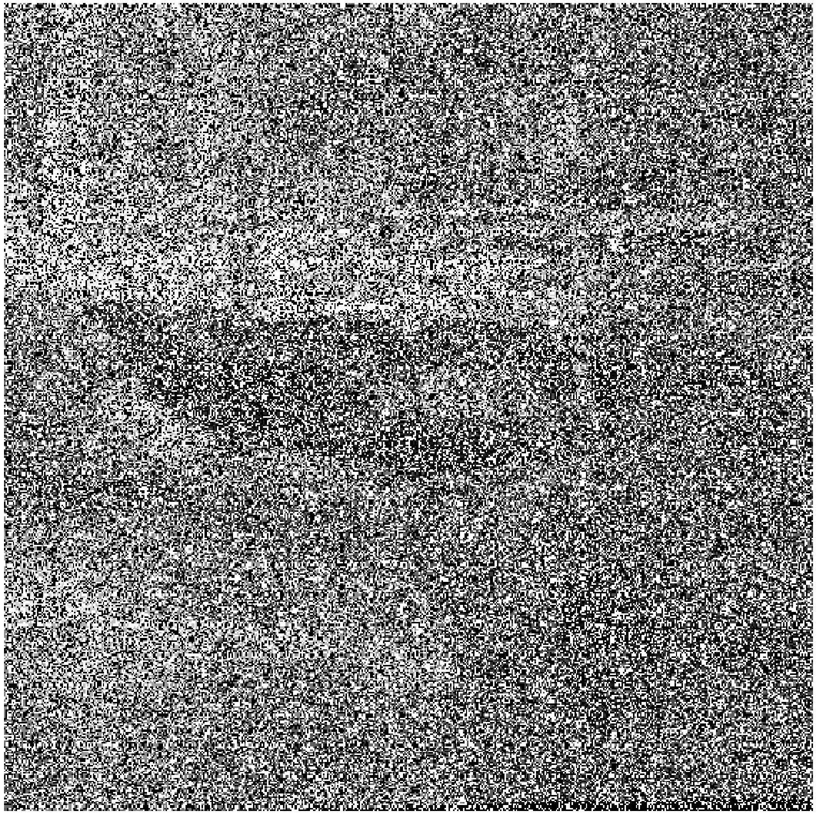
Phase 1: ***Detect*** noise candidate set $\mathcal{N}$ by ***Adaptive Median Filter***

Phase 2: ***Restore*** pixels in $\mathcal{N}$ by ***Tight Frame Method.***

(Do an inpainting on $\mathcal{N}$.)

70% Salt-&-Pepper Noise

Original 512-by-512 bridge image



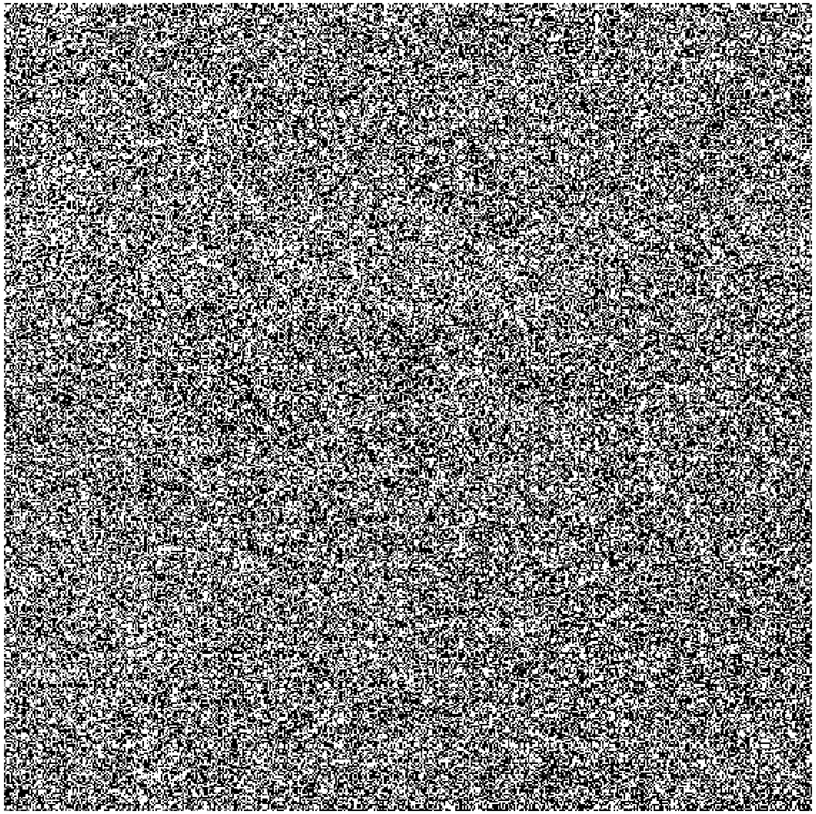
Corrupted with 70% salt-and-pepper noise

Numerical Results

Adaptive Median Filter
PSNR = 21.68dB
CPU time = 151.67s

Our Method
PSNR = 24.69dB (3.01dB)
CPU time = 182.05s

90% Salt-&-Pepper Noise



Corrupted with 90% salt-and-pepper noise

Numerical Results

Adaptive Median Filter

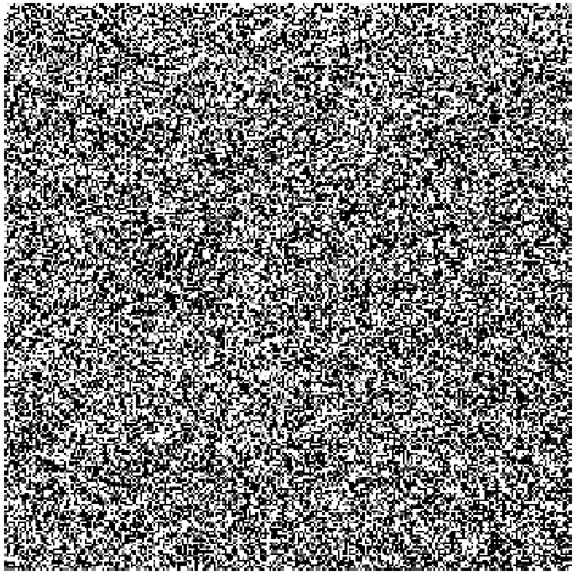
PSNR = 18.26dB

CPU time = 267.65s

Our Method

PSNR = 21.81dB (3.55dB)

CPU time = 382.43s



90% noise

AMF

AMF+Variation

AMF+Frame

Comparison with Variational Method

Image	Noise level	Variational	Framelet
Lena 256x256	50%	30.5	31.3
	70%	27.4	28.8
	90%	22.9	24.2
Lena 512x512	50%	33.1	33.8
	70%	29.7	31.2
	90%	25.4	26.5
Cameraman 256x256	70%	24.8	25.8
Goldhill 512x512		29.9	30.0
Boat 512x512		28.0	29.1
Barbara 512x512		24.6	25.7
Bridge 512x512		24.7	24.7

Random-Valued Impulse Noise

$\mathbf{f} = (f_{i,j})$: true image with $f_{i,j} \in [0, 255]$.
$\mathbf{y} = (y_{i,j})$: observed noisy image.

$$y_{i,j} = \begin{cases} n_{i,j} & \text{with probability } r, \\ f_{i,j} & \text{with probability } 1 - r, \end{cases}$$

where $n_{i,j}$ is randomly distributed in [0,255].

ROAD Statistic

The Local Image Statistic ROAD (2005)

Noisy Image

$y_{i-1,j-1}$	$y_{i-1,j}$	$y_{i-1,j+1}$
$y_{i,j-1}$	$y_{i,j}$	$y_{i,j+1}$
$y_{i+1,j-1}$	$y_{i+1,j}$	$y_{i+1,j+1}$

Absolute Difference:

$$d_{st}(y_{i,j}) = |y_{i+s,j+t} - y_{i,j}|$$

Sort $d_{st}(y_{i,j})$:

$$r_1(y_{i,j}) \leq \ldots \leq r_8(y_{i,j})$$

The ROAD Statistic:

$$\mathrm{ROAD}_m(y_{i,j}) = \sum_{k=1}^{m} r_k(y_{i,j})$$

ROAD Statistic

$$\begin{pmatrix} 213 & 171 & 88 \\ 216 & 186 & 107 \\ 218 & 202 & 139 \end{pmatrix} \longrightarrow \begin{pmatrix} \mathbf{27} & \mathbf{15} & 98 \\ \mathbf{30} & - & 79 \\ 32 & \mathbf{16} & 47 \end{pmatrix}$$

Original Neighborhood $\swarrow$ Absolute Differences

$$r_1 = 15,\ r_2 = 16$$
$$r_3 = 27,\ r_4 = 30 \longrightarrow \begin{array}{c} \mathrm{ROAD} = \sum_{i=1}^{4} r_i \\ 15 + 16 + 27 + 30 = 88 \end{array}$$

The four smallest absolute differences

Final Calculation of ROAD

ROAD Statistic

- **Advantages:**
  - noisy pixels usually have intensities vary greatly from those of its neighbors (i.e. their ROAD values will be large)
  - noise-free pixels should have at least half of the neighbors having similar intensity (i.e. their ROAD values will be small)

- **Disadvantages:**
  - for random-valued impulse noise, some noise values may be close to their neighbors' values
  - the ROAD values may be in the middle of range and not large enough to distinguish them.

ROLD Statistic

Absolute Difference:

$$d_{st}(y_{i,j}) = 1 + \max\{\log_a |y_{i+s,j+t} - y_{i,j}|, -b\}/b$$

Sort $D_{st}(y_{i,j})$:

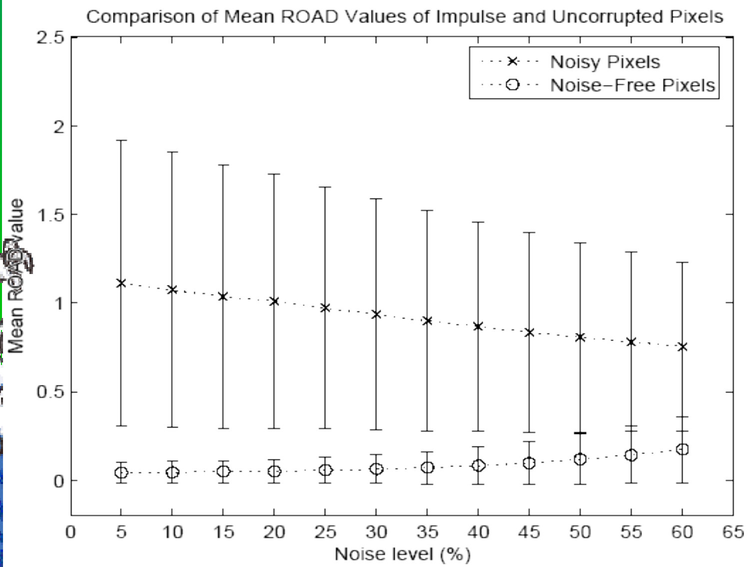
$$R_1(y_{i,j}) \leq \ldots \leq R_8(y_{i,j})$$

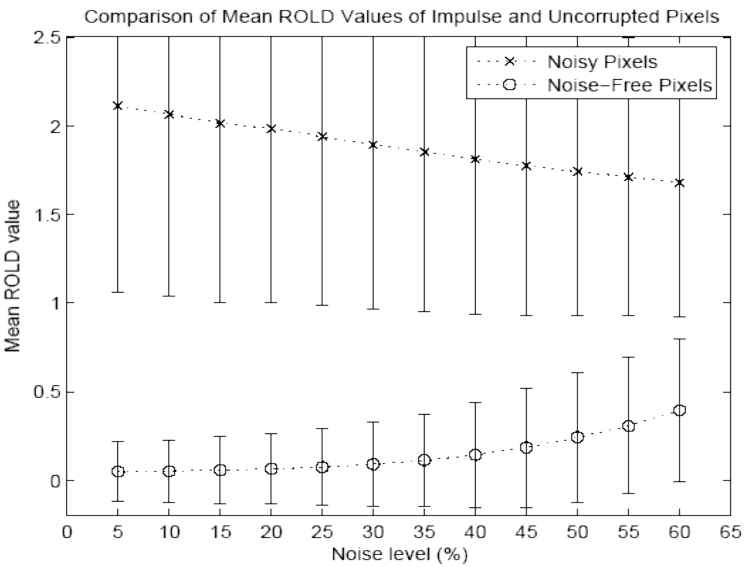
The ROLD Statistic:

$$\mathrm{ROLD}_m(y_{i,j}) = \sum_{k=1}^{m} R_k(y_{i,j})$$

Dong, C., and Xu, *IEEE TIP,* 2007

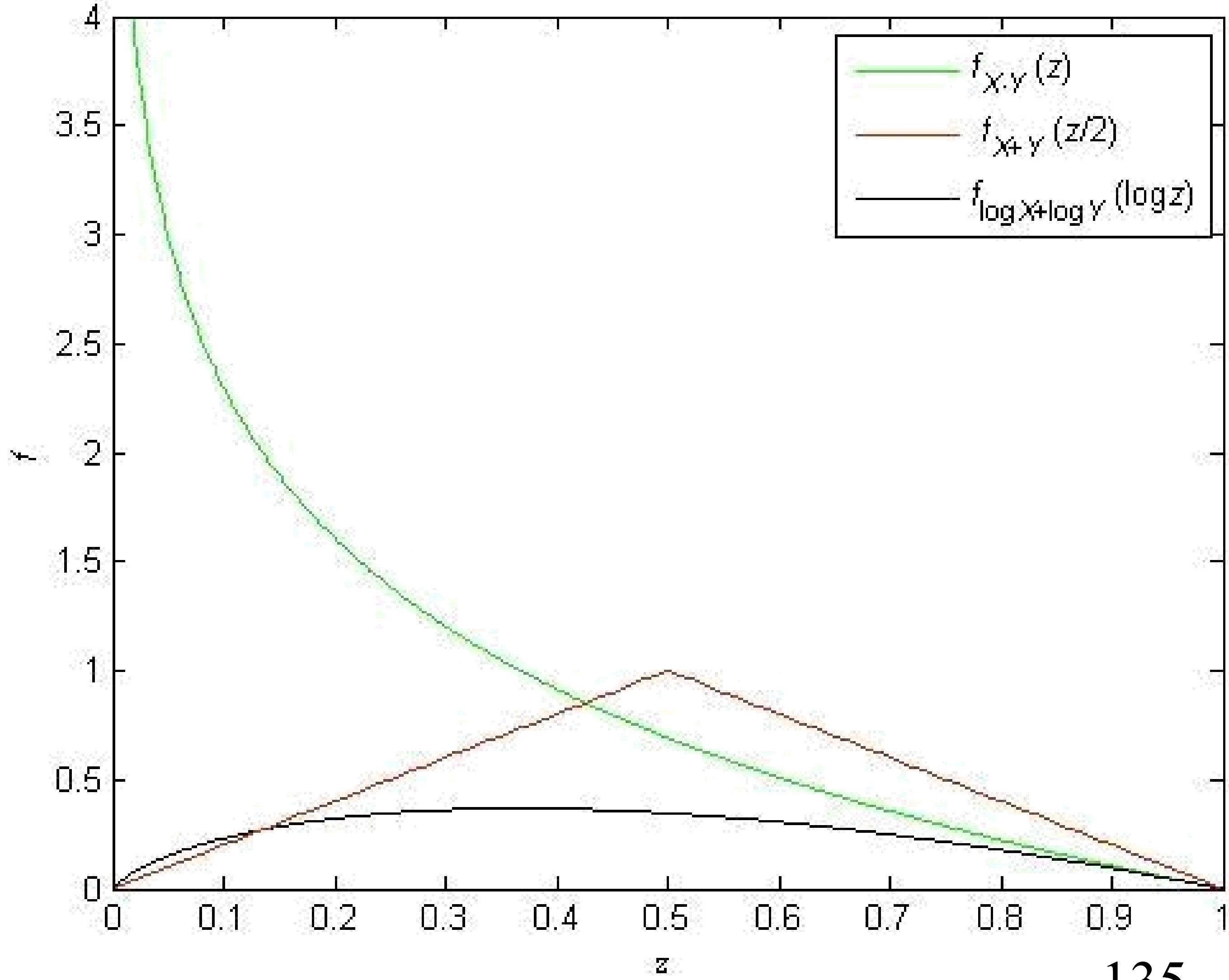
Comparison of ROAD & ROLD





- Differences between the means and the ranges of the error bars all increased
- Make it easier to separate noisy pixels from noise-free ones

Explanation by PDF's

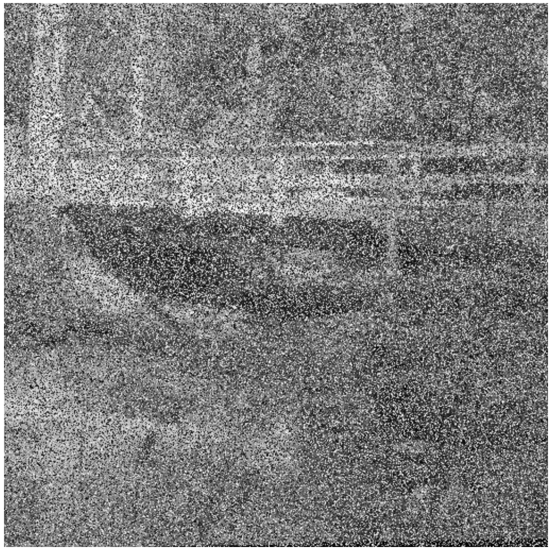


60% *Random-Valued*

ACWMF (21.19dB)

CHN Method (24.62dB)

New Detector (29.03dB)



60% *Random-Valued*

ACWMF (19.27dB)

CHN Method (20.89dB)

New Detector (22.59dB)

A New Noise Detector (ROLD)

Method	"Lena" image			"Bridge" image		
	20%	40%	60%	20%	40%	60%
Median Filter	32.37	27.64	21.58	25.04	22.17	19.36
Switching Scheme I	32.93	27.90	20.61	26.26	22.66	19.13
Switching Scheme II	33.43	27.75	20.61	25.90	22.85	19.04
SD-ROM Filter	35.29	28.59	21.64	27.04	23.33	19.43
PSM Filter	35.09	28.92	22.06	26.33	22.75	19.73
TSM Filter	34.21	28.30	21.67	26.52	22.89	19.60
MSM Filter	35.44	29.26	22.14	27.27	23.55	20.07
ACWM Filter	36.07	28.79	21.19	27.08	23.23	19.27
PWMAD Filter	36.50	31.41	24.30	26.90	23.83	20.83
ACWM-EPR	36.57	32.21	24.62	27.66	24.60	20.89
ROAD-EPR	36.79	32.32	28.37	27.42	24.52	22.04
ROLD-EPR	*37.45*	*32.76*	*29.03*	*27.86*	*24.79*	*22.59*

Outline

1. Tight Frames
2. Inpainting
3. Impulse Noise Removal
4. High Resolution Image Reconstruction
5. Video Enhancement
6. Extension
7. Convergence Analysis
8. Combining PDE and Framelets

Goal

Resolution = 256 × 256

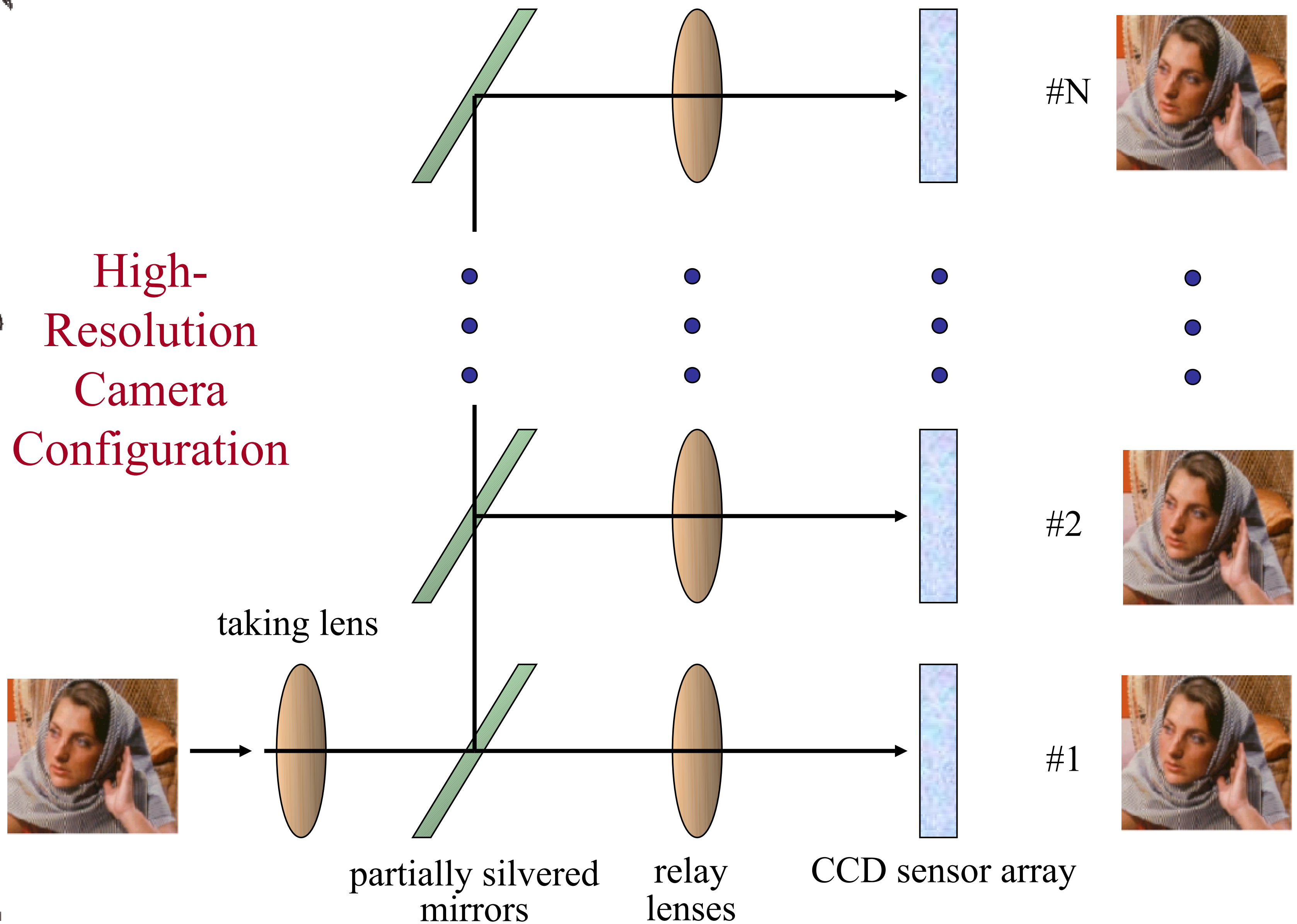
Resolution = 64 × 64

Four low resolution images (64×64) of the same scene. Each shifted by sub-pixel length.

Construct a high-resolution image (128×128) from them.

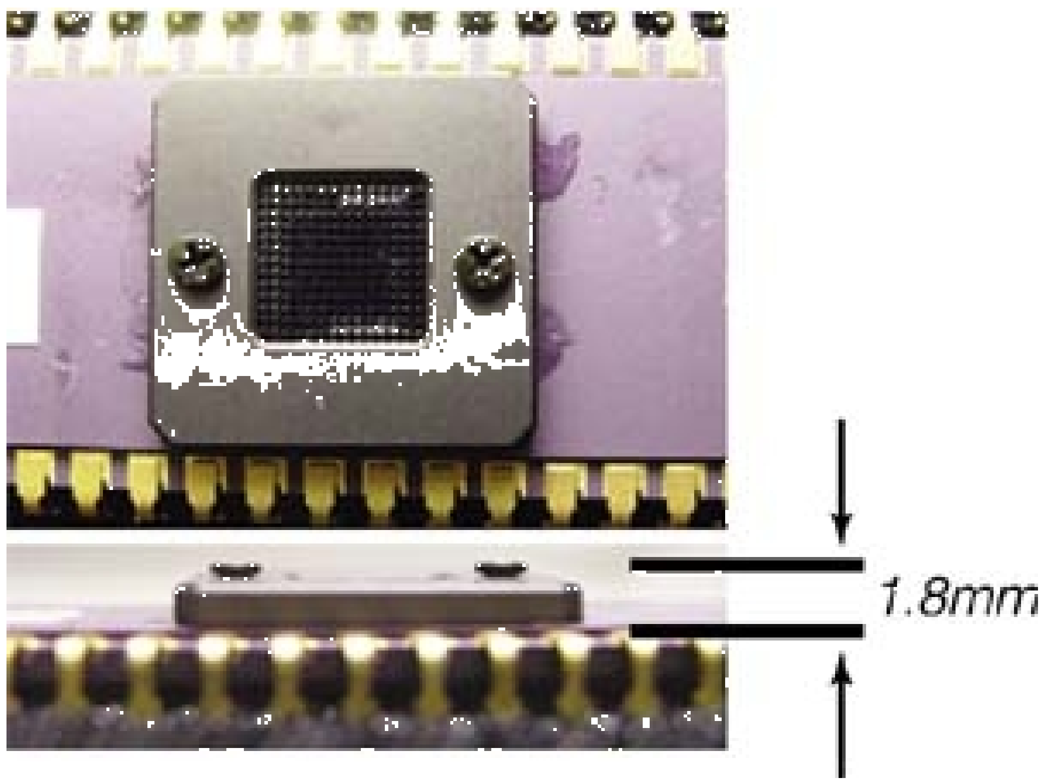
Boo and Bose (IJIST, 97):

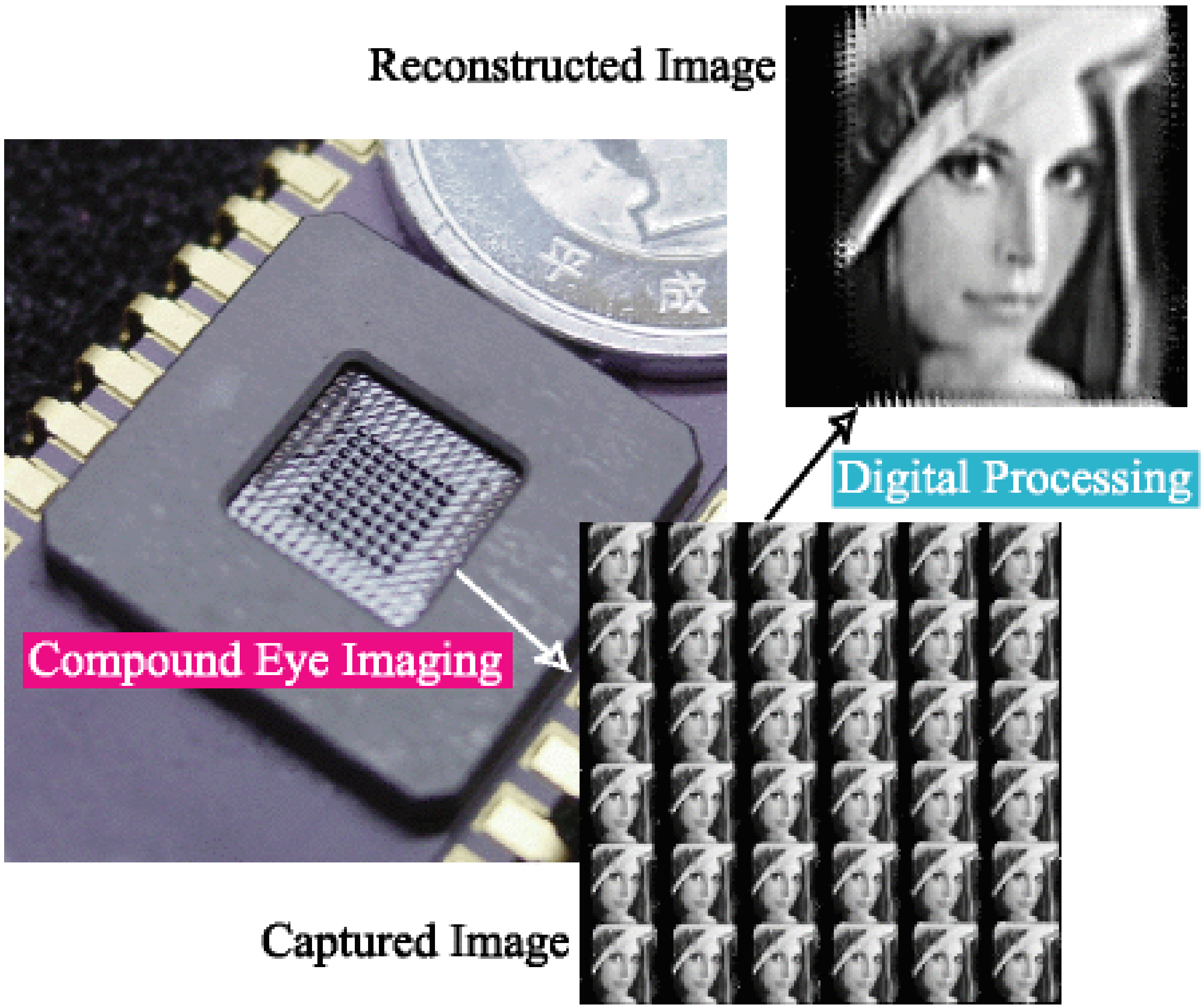
High-Resolution Camera Configuration



Ultra-thin Image Information Input Card

High-resolution image without long focal-length lens

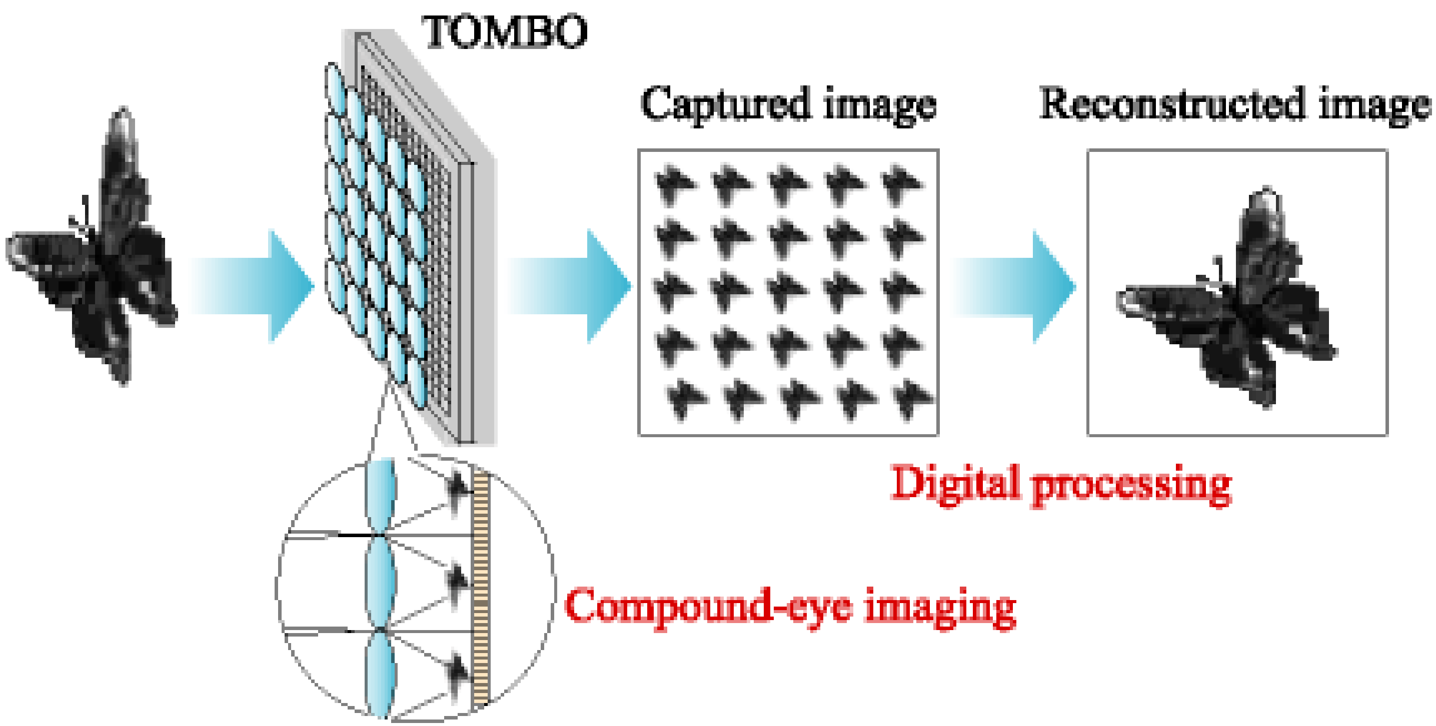




Reconstructed Image
Digital Processing
Compound Eye Imaging
Captured Image

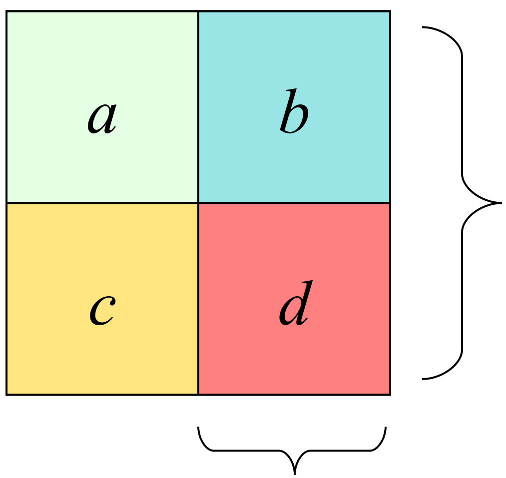
High-Resolution Image Reconstruction

Not 1, but many lens ---
compound eyes



Modeling of HR Image Reconstruction

4 low-resolution images merge into 1

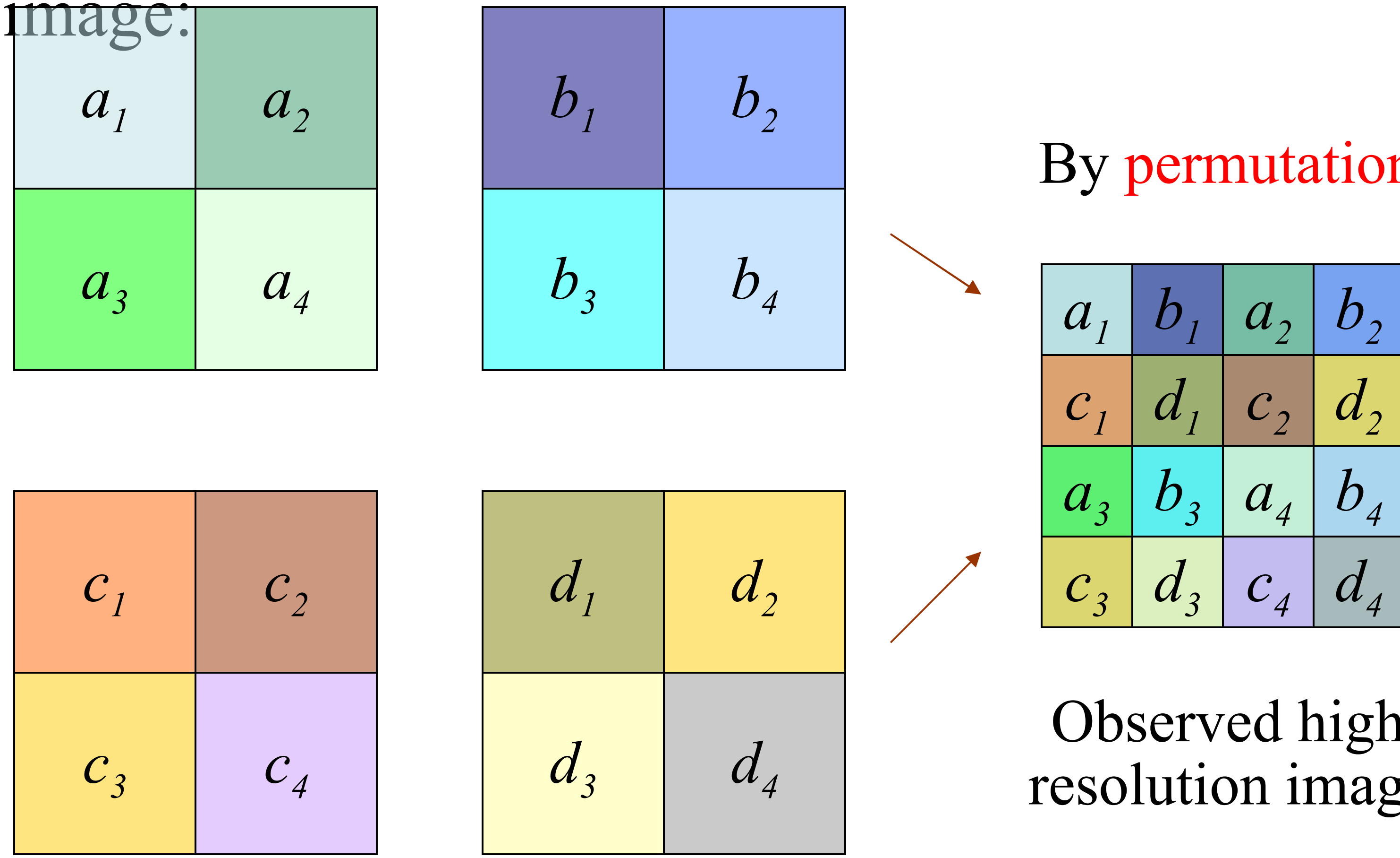


low-resolution pixel

given intensity = (a+b+c+d)/4

high-resolution pixel

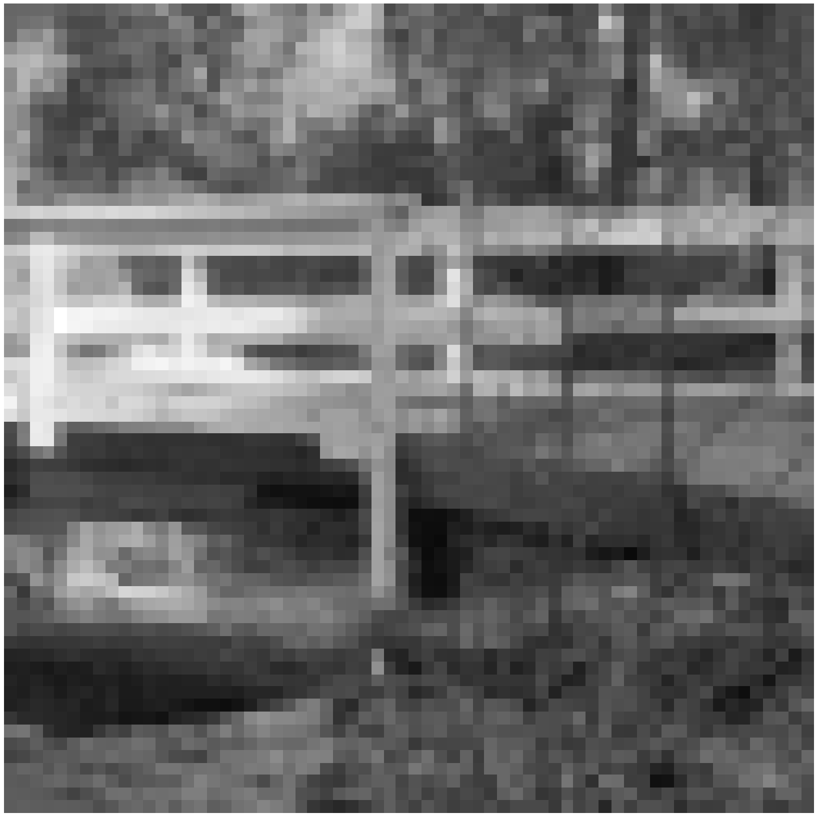
Four 2×2 images merged into one 4×4 image:



Observed high-resolution image

Four low resolution images

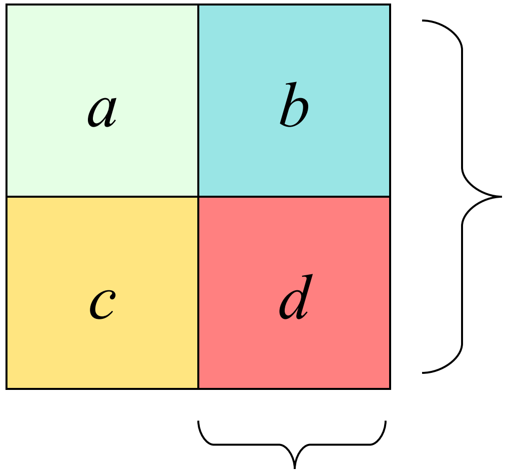
Observed High Resolution Image



One of the 4 LR Images

The Observed HR Image

Can we get something better?



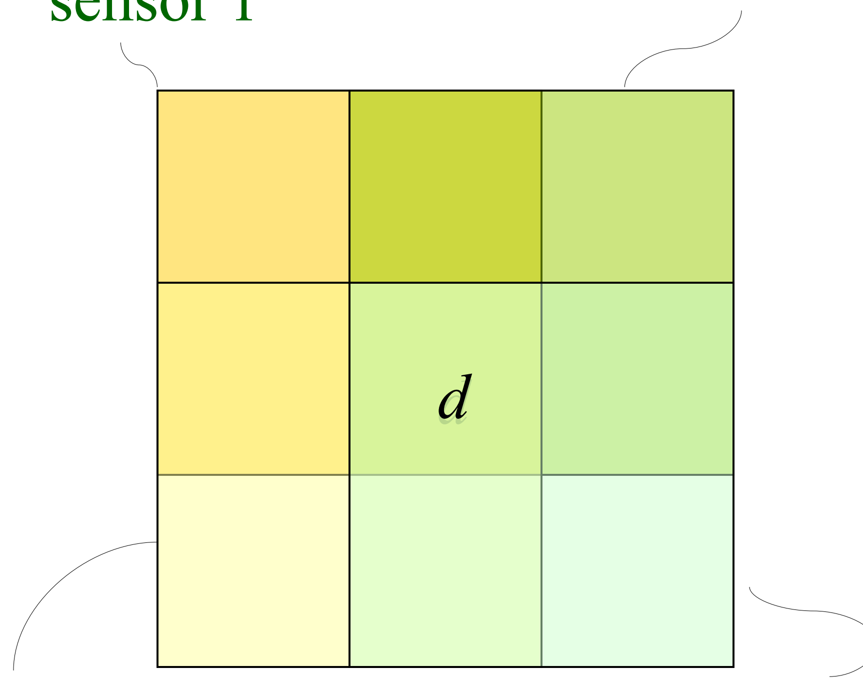
low-resolution pixel

given intensity = (a+b+c+d)/4

high-resolution pixel

sensor 1

sensor 2



sensor 3

sensor 4

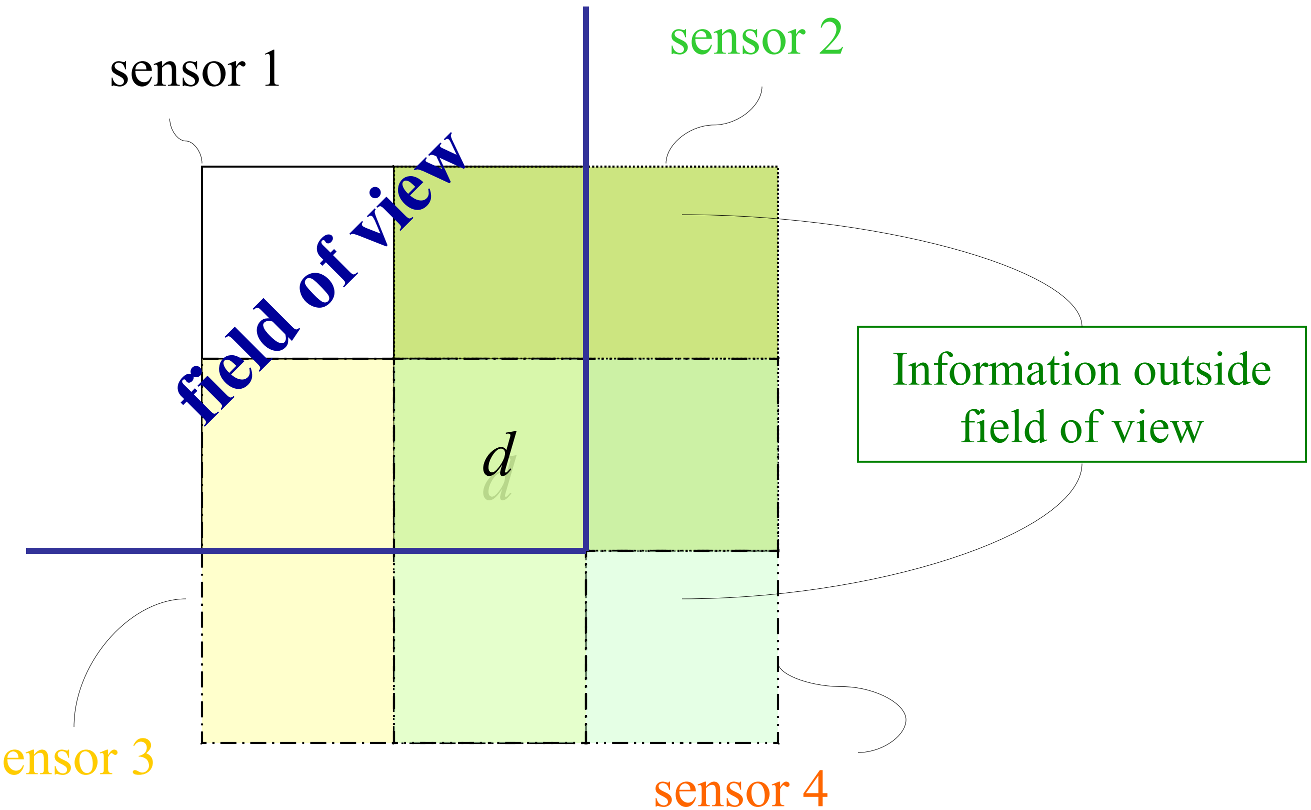
4 low-resolution images merge into 1

The Blurring Matrix:

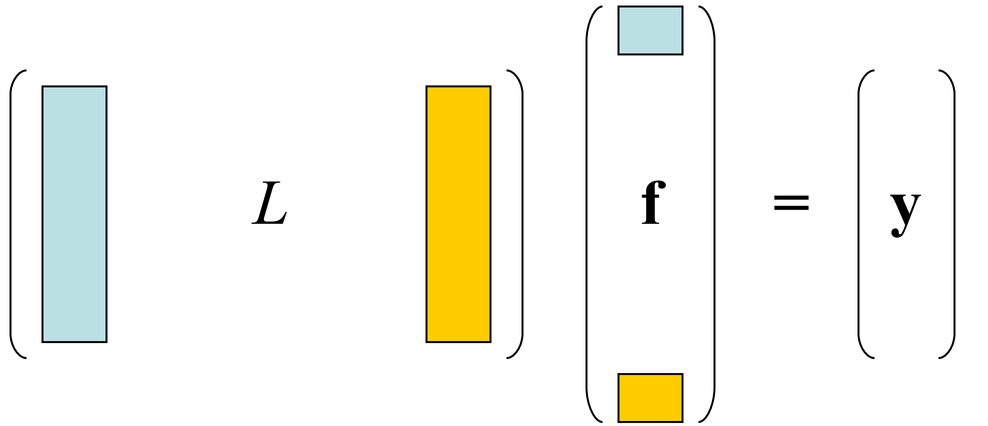
Let **f** be the true image, **y** the observed HR image, then

$$L\mathbf{f} = (L_x \otimes L_y)\mathbf{f} = \mathbf{y}$$

f involves information of true image outside the field of view.



Matrix is fat and long:



Assume something about the image outside the field of view (boundary conditions).

After adding boundary condition:

$$\begin{pmatrix} & & \\ & L & \\ & & \end{pmatrix} \begin{pmatrix} \\ \mathbf{f} \\ \\ \end{pmatrix} = \begin{pmatrix} \\ \mathbf{y} \\ \\ \end{pmatrix}.$$

$N^2 \times N^2$ $\quad N^2 \times 1$ $\quad N^2 \times 1$

Boundary Conditions

Periodic Boundary Condition (Gonzalez and Woods, 93):

Assume data are periodic near the boundary.

- $4\times\ \ 4$ sensor array (16 low-resolution to 1 high-resolution). Matrix is:

$$\frac{1}{8}\begin{pmatrix} 2 & 2 & 1 & & & 1 & 2 \\ 2 & 2 & 2 & 1 & & & 1 \\ 1 & 2 & 2 & 2 & 1 & & \\ & & & & & & \\ & & 1 & 2 & 2 & 2 & 1 \\ 1 & & & 1 & 2 & 2 & 2 \\ 2 & 1 & & & 1 & 2 & 2 \end{pmatrix} \otimes \frac{1}{8}\begin{pmatrix} 2 & 2 & 1 & & & 1 & 2 \\ 2 & 2 & 2 & 1 & & & 1 \\ 1 & 2 & 2 & 2 & 1 & & \\ & & & & & \mathrm{e} & \\ & & 1 & 2 & 2 & 2 & 1 \\ 1 & & & 1 & 2 & 2 & 2 \\ 2 & 1 & & & 1 & 2 & 2 \end{pmatrix}$$

- Block-circulant-circulant-block system.
- Diagonalized by 2D Fourier transforms in $O(N^2 \log N)$.

Ringing effect is prominent.

original image

reconstructed image

observed high-resolution image

Dirichlet (Zero) BC (Boo and Bose, IJIST 97):

Assume data zeros outside boundary

- Matrix is

$$\frac{1}{8}\begin{pmatrix} 2 & 2 & 1 & & & & \\ 2 & 2 & 2 & 1 & & & \\ 1 & 2 & 2 & 2 & 1 & & \\ & & & & & & \\ & & 1 & 2 & 2 & 2 & 1 \\ & & & 1 & 2 & 2 & 2 \\ & & & & 1 & 2 & 2 \end{pmatrix} \otimes \frac{1}{8}\begin{pmatrix} 2 & 2 & 1 & & & & \\ 2 & 2 & 2 & 1 & & & \\ 1 & 2 & 2 & 2 & 1 & & \\ & & & & & \text{e} & \\ & & 1 & 2 & 2 & 2 & 1 \\ & & & 1 & 2 & 2 & 2 \\ & & & & 1 & 2 & 2 \end{pmatrix}$$

- Block-Toeplitz-Toeplitz-Block system.
- Cannot be diagonalized by sine-transforms.
- Iterative solvers with circulant preconditioners (C. and Jin, SIAM, 07).

Ringing effect still prominent:

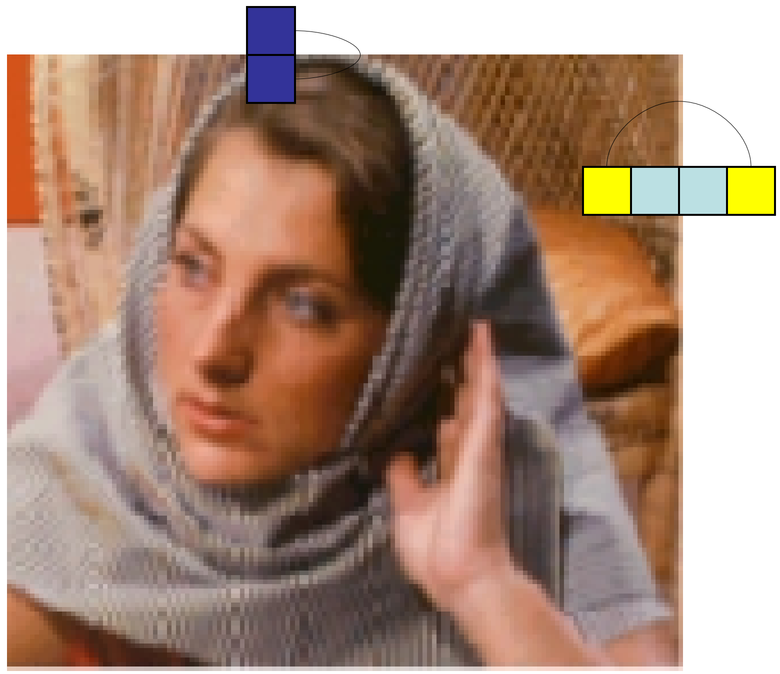
original image

reconstructed image

observed high-resolution image

Neumann Boundary Condition (Ng, C. & Tang (SISC 00)):

Assume data are reflective near boundary.



- Matrix is

$$\frac{1}{8}\begin{pmatrix} 4 & 3 & 1 & & & & \\ 3 & 2 & 2 & 1 & & & \\ 1 & 2 & 2 & 2 & 1 & & \\ & & & & & & \\ & & 1 & 2 & 2 & 2 & 1 \\ & & & 1 & 2 & 2 & 3 \\ & & & & 1 & 3 & 4 \end{pmatrix} \otimes \frac{1}{8}\begin{pmatrix} 4 & 3 & 1 & & & & \\ 3 & 2 & 2 & 1 & & & \\ 1 & 2 & 2 & 2 & 1 & & \\ & & & & & \mathrm{e} & \\ & & 1 & 2 & 2 & 2 & 1 \\ & & & 1 & 2 & 2 & 3 \\ & & & & 1 & 3 & 4 \end{pmatrix}$$

- Block Toeplitz-plus-Hankel with Toeplitz-plus-Hankel blocks. (*Hankel = constant along anti-diagonals*.)
- Diagonalized by 2D cosine-transforms in $O(N^2 \log N)$.
- Holds for sensor array of any size, and
- more generally for *all* symmetric blurring functions.

Ringing effect is smaller:

original image

reconstructed image

observed high-resolution image

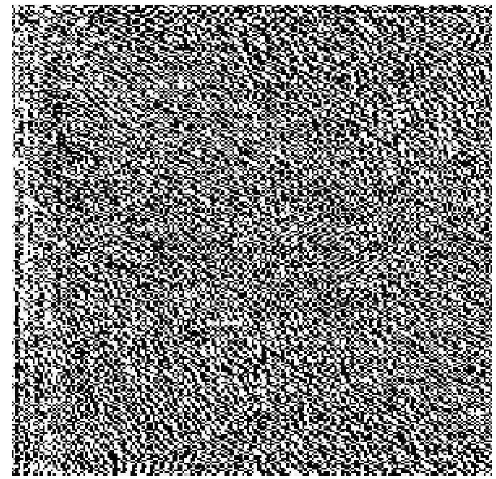
Anti-reflexive Boundary Condition (Serra, SISC (2003))

Assume data are negated and reflected near boundary.

- Matrix approximately diagonalized by 2D sine-transforms in $O(N^2 \log N)$..

The problem $L\,\mathbf{f} = \mathbf{y} + \mathbf{n}$ is ill-conditioned.

$\mathbf{y}$



$L^{-1}\mathbf{y}$

Regularization is required:

$$\min_{\mathbf{f}} \frac{1}{2}\left[\|L\mathbf{f} - \mathbf{y}\|_2^2 + \beta\|R\mathbf{f}\|_2^2\right],$$

where R can be I, ∇, Δ, or the TV norm operator.

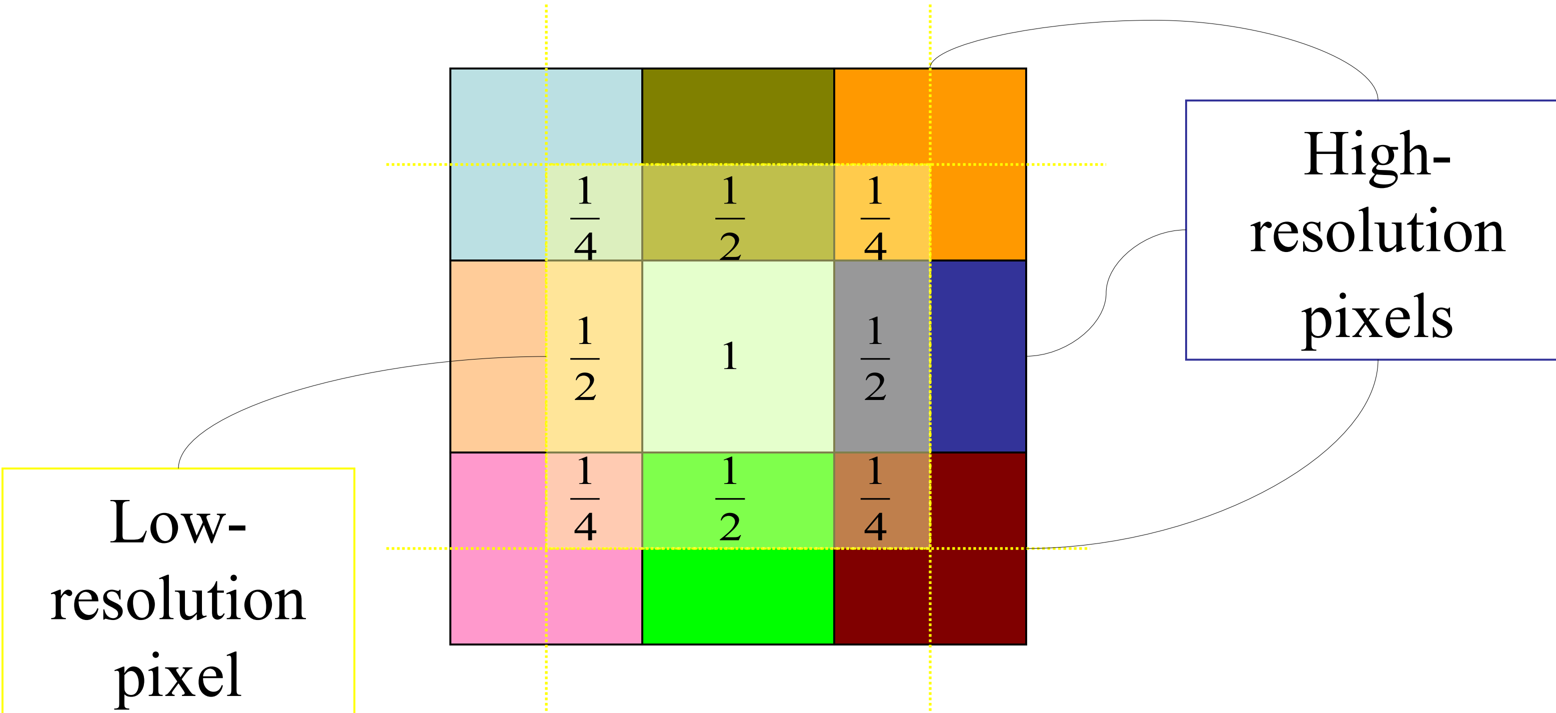
Euler-Lagrange equation $\Leftrightarrow$ Tikhonov regularization

$$(L^tL + \beta R^2)\mathbf{f} = L^t\mathbf{y},$$

$$(L^tL + \beta\Delta)^{-1}L^t\mathbf{y}$$

Tightframe Approach

Consider 2× 2 sensor array:



Averaging process = a lowpass filter with refinement mask:

$$\frac{1}{2}\left(\ldots,0,\frac{1}{2},1,\frac{1}{2},0,\ldots\right)\otimes\frac{1}{2}\left(\ldots,0,\frac{1}{2},1,\frac{1}{2},0,\ldots\right)$$

Key Observation

The low-resolution images $\mathbf{y}$ are obtained by passing the high-resolution image $\mathbf{f}$ via the $h_0 = \frac{1}{2}[1, 2, 1]$ filter. Thus we have

$$H_0 \mathbf{f} = \mathbf{y}.$$

The problem is to find $\mathbf{f}$ from $\mathbf{y} = H_0 \mathbf{f}$.

Recall that there exists H_1 and H_2 such that

$$H_0^* H_0 + H_1^* H_1 + H_2^* H_2 = I$$

Tight-frame Algorithm

Using

$$I = H_0^* H_0 + H_1^* H_1 + H_2^* H_2$$

we have

$$\begin{aligned} \mathbf{f} &= \sum_{i=0}^{2} H_i^* H_i \mathbf{f} \\ &= H_0^* H_0 \mathbf{f} + \sum_{i=1}^{2} H_i^* H_i \mathbf{f} \\ &= H_0^* \mathbf{y} + \sum_{i=1}^{2} H_i^* H_i \mathbf{f}. \end{aligned}$$

Tight-frame Algorithm

- Choose $\mathbf{f}^{(0)} \in L^2([-\pi,\pi]^2)$;
- Iterate until convergence

$$\mathbf{f}^{(r+1)} = H_0^* \mathbf{y} + \sum_{i=1}^{2} H_i^* \mathcal{T} H_i \mathbf{f}^{(r)},$$

where $\mathcal{T}$ is the thresholding operator.

Chan, Chan, Shen, Shen, SISC (2003), where we used biorthogonal wavelets corresponding to H_0 instead of linear tight frames.

Numerical Examples

2-by-2 sensor array:

SNR	Tikhonov with Δ		Wavelet Algorithm		
(dB)	PSNR	RE	PSNR	RE	Iter.
30	32.55	0.0437	34.48	0.0350	9
40	33.88	0.0375	35.23	0.0321	12

4-by-4 sensor array:

SNR	Tikhonov with Δ		Wavelet Algorithm		
(dB)	PSNR	RE	PSNR	RE	Iter.
30	29.49	0.0621	30.11	0.0579	30
40	30.17	0.0573	30.56	0.0549	45

For 4-by-4 sensors, $h_0 = \frac{1}{4}[1, 2, 2, 2, 1]$.

4×4 *sensor array*

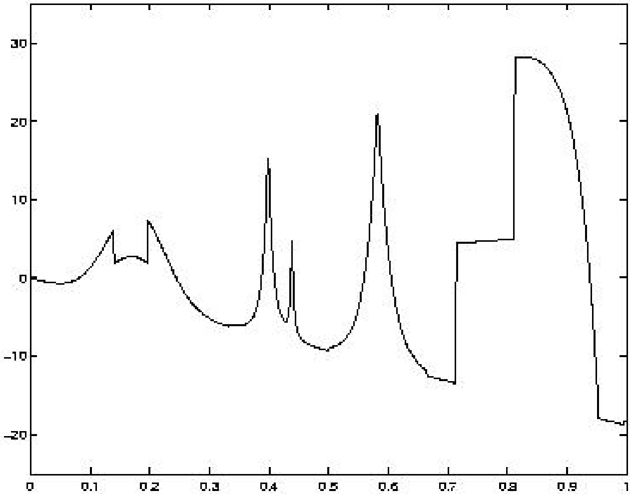
Observed Image

Low-resolution Frame

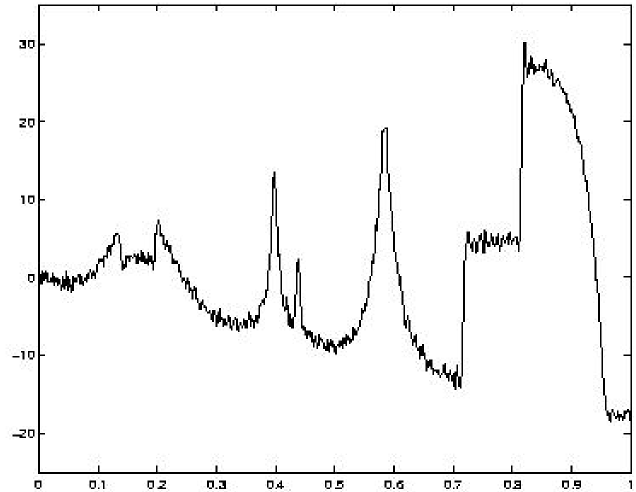

Tikhonov


Wavelet

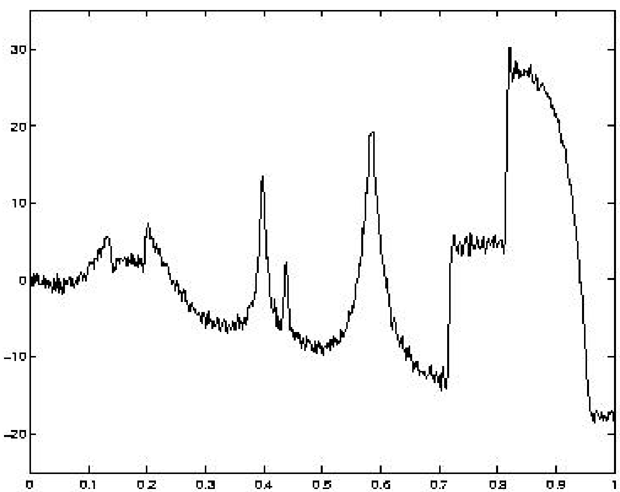
1-D Example: Signal from Donoho's Wavelet Toolbox. Blurred by *1-D* 4-to-1 filter.



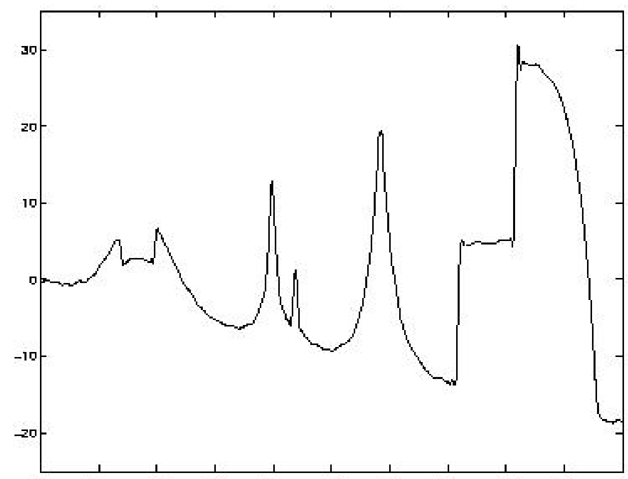
Original Signal



Observed HR Signal

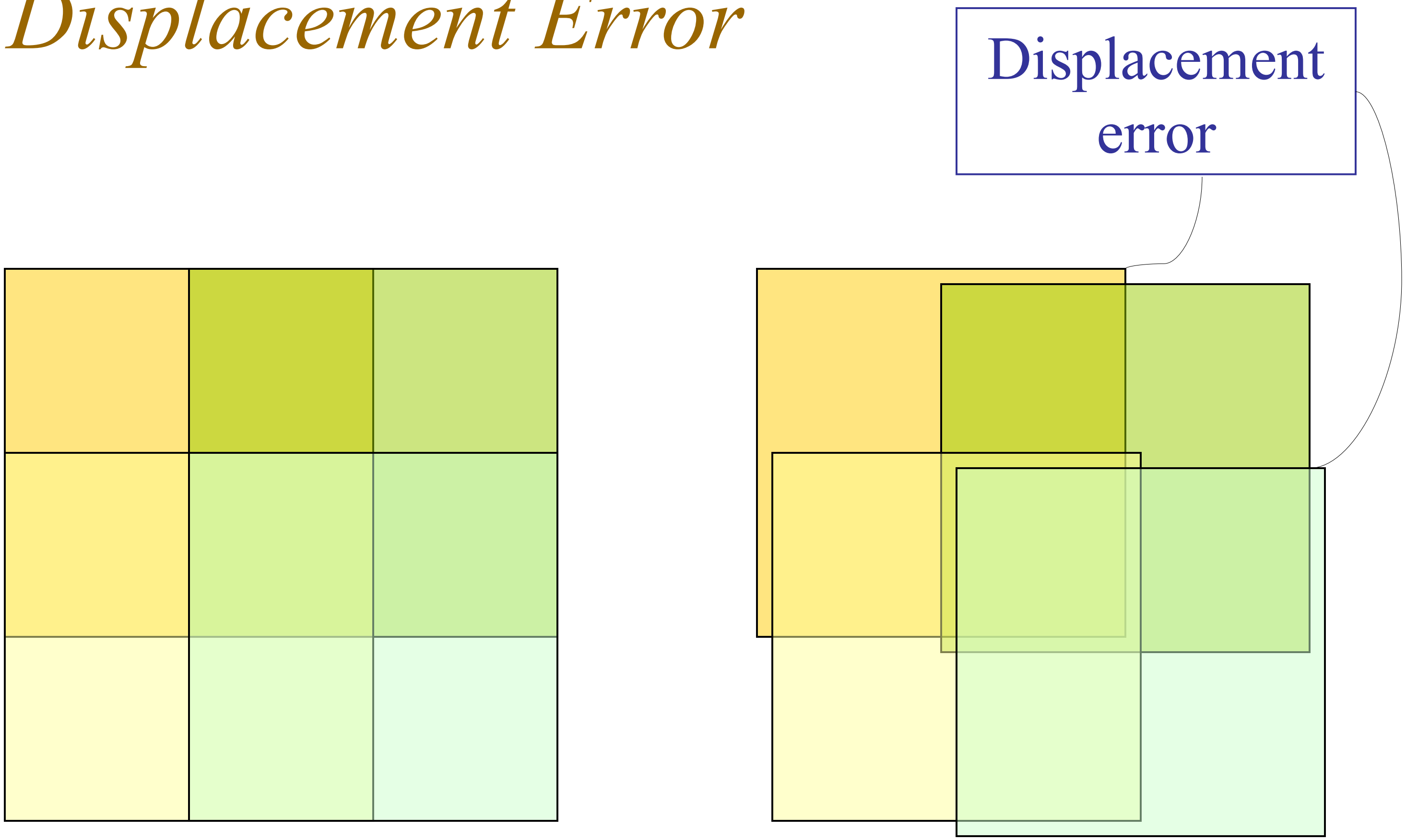


Tikhonov



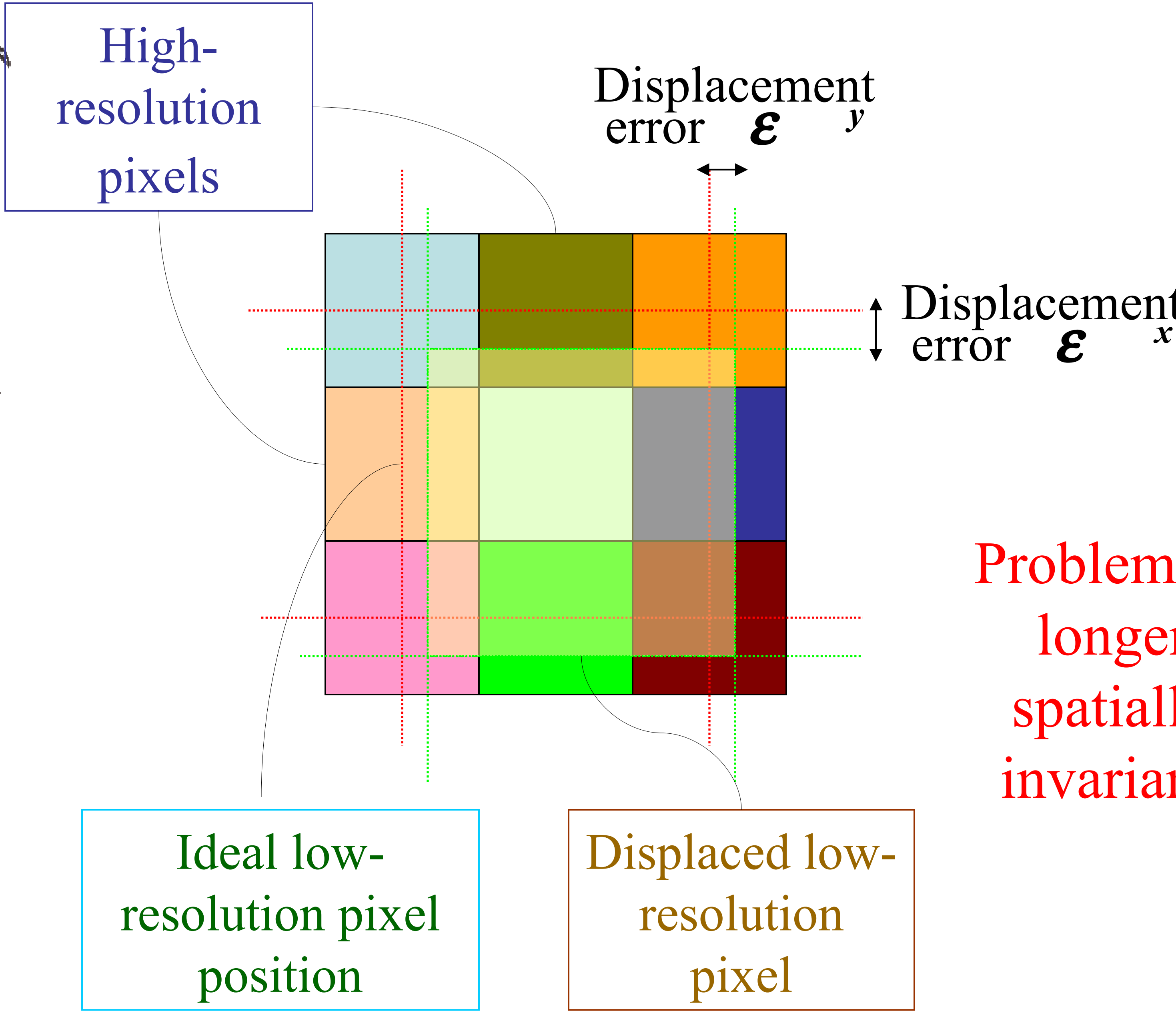
Wavelet

Displacement Error



Ideal pixel positions

Pixels with displacement errors



Problem no longer spatially invariant.

Tightframe Algorithm

For K-by-K sensor array, the filter is

$$a_{K,\epsilon} \equiv \frac{1}{K}\left(\frac{1}{2}+\epsilon, \underbrace{1, \cdots, 1}_{K-1}, \frac{1}{2}-\epsilon\right).$$

Therefore we have to find a minimally supported tightframe system with $a_{K,\epsilon}$ as its lowpass filter.

C., Chan, Shen, Shen, LAA (2003) using a bi-orthogonal wavelet system.

Drawbacks of Wavelet Algorithm

- The regularity of the scaling functions varies with the displacement errors ϵ, and in some cases, the function can even be discontinuous (Shen-Sun, 04).

- Since the filters are not symmetric, we only can impose the periodic boundary conditions, which is not good in practice.

- The design of the wavelet filters depends on the displacement errors ϵ.

Tight Frame Approach

Observe that the filter can be written as:

$$\begin{aligned} & \frac{1}{K}\left(\frac{1}{2}+\epsilon, 1, \cdots, 1, \frac{1}{2}-\epsilon\right) \\ = \; & \frac{1}{K}\left(\frac{1}{2}, 1, \cdots, 1, \frac{1}{2}\right)+\frac{\epsilon}{K}(1,0, \cdots, 0,-1). \end{aligned}$$

Tight Frame Approach

Construct a multi-resolution analysis with lowpass filter as

$$\left(\frac{1}{2}, 1, \cdots, 1, \frac{1}{2}\right)$$

and one of the highpass filters as

$$(1, 0, \cdots, 0, -1).$$

$\boxed{\textit{Wavelet cannot, but wavelet tight frame can}}$

An Example

For $K = 4$, the filters associated with the tight frame system are

$$\frac{1}{4}[\frac{1}{2}, 1, 1, 1, \frac{1}{2}], \ \frac{\sqrt{2}}{8}[1, 0, 0, 0, -1],$$

$$\frac{1}{4}[-\frac{1}{2}, 1, -1, 1, -\frac{1}{2}], \ \frac{1}{4}[\frac{1}{2}, 1, 0, -1, -\frac{1}{2}],$$

$$\frac{\sqrt{2}}{8}[1, 0, -2, 0, 1], \ \frac{1}{4}[-\frac{1}{2}, 1, 0, -1, \frac{1}{2}].$$

We have the perfect reconstruction formula:

$$\sum_i H_i^* H_i = I,$$

and the observed image $\mathbf{y}$ is given by

$$\mathbf{y} = (H_0 + \sqrt{2}\epsilon H_1)\mathbf{f}.$$

Thus

$$\mathbf{f} = H_0^* \left[\mathbf{y} - \sqrt{2}\epsilon H_1 \mathbf{f}\right] + \sum_{i \neq 0} H_i^* H_i \mathbf{f}$$

Tight Frame Algorithm

Iterate on r:

$$\mathbf{f}^{(r+1)} = H_0^* \left[\mathbf{y} - \sqrt{2}\epsilon H_1 \mathbf{f}^{(r)} \right] + \sum_{i \neq 0} H_i^* H_i \mathbf{f}^{(r)}$$

Applying the thresholding operator $\mathcal{T}$:

$$\mathbf{f}^{(r+1)} = H_0^* \mathbf{y} - \sqrt{2}\epsilon H_0^* \mathcal{T}(H_1 \mathbf{f}^{(r)}) + \sum_{i \neq 0} H_i^* \mathcal{T}(H_i \mathbf{f}^{(r)})$$

C., Shen, Shen, ACHA (2004).

16 merges into 1 with calibration error

Boat Image:

SNR	2-by-2 sensor array		4-by-4 sensor array	
(dB)	Wavelet	Framelet	Wavelet	Framelet
20	30.45	33.87	27.16	29.35
30	30.80	35.41	27.20	30.38
40	30.85	36.26	27.21	31.06

Bridge Image:

SNR	2-by-2 sensor array		4-by-4 sensor array	
(dB)	Wavelet	Framelet	Wavelet	Framelet
20	27.66	28.89	23.99	25.66
30	27.92	29.22	24.01	26.05
40	28.00	29.37	24.01	26.19

Wavelet
(PSNR=27.2dB)

Tight Frame
(PSNR=30.38dB)

Wavelet
(PSNR=24.1dB)

Tight Frame
(PSNR=26.05dB)

Outline

1. Tight Frames
2. Inpainting
3. Impulse Noise Removal
4. High Resolution Image Reconstruction
5. Video Enhancement
6. Extension
7. Convergence Analysis
8. Combining PDE and Framelets

Goal

One of the frame
in a video

Before enhancement

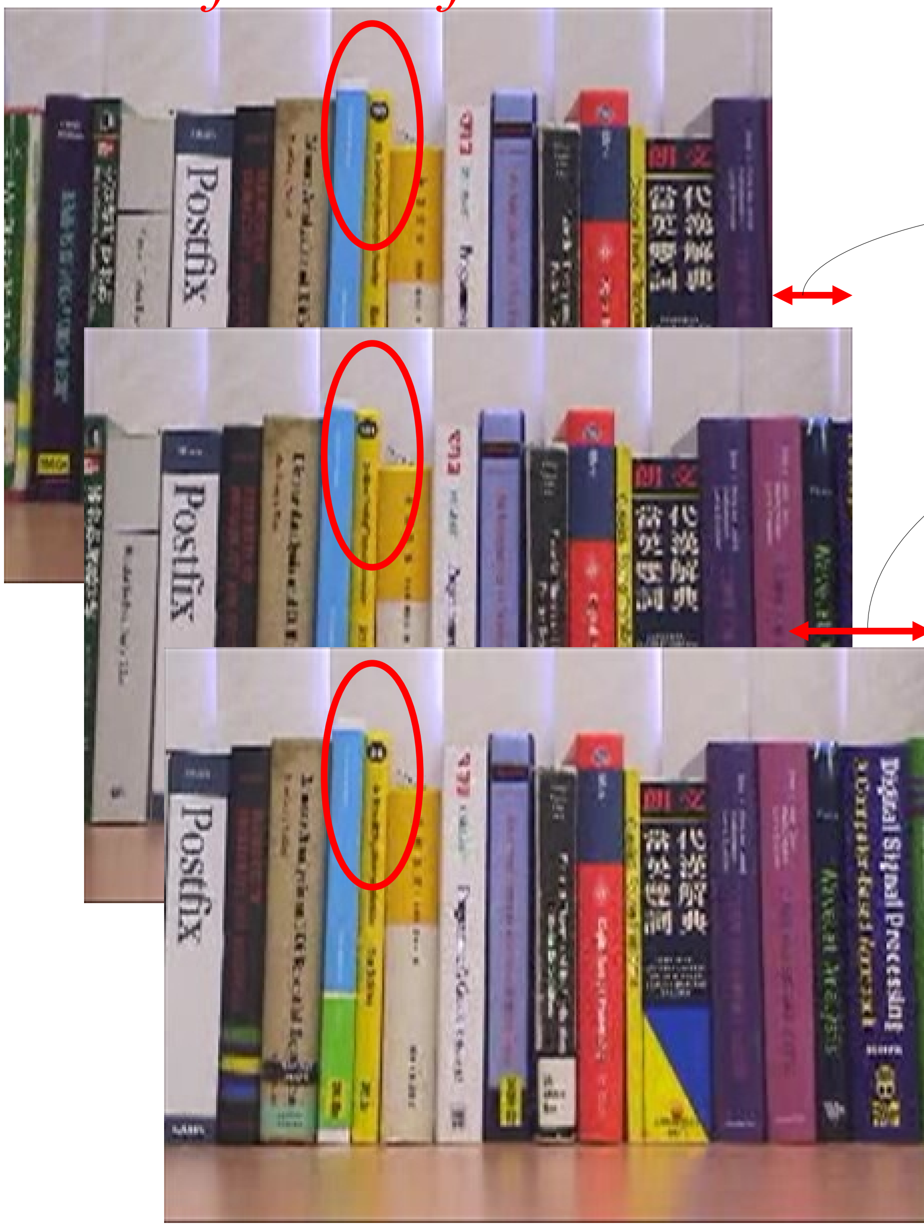
After enhancement

Video Enhancement

A 352-by-288 video from a video recorder

30 frames/second

Reference frame



Displacement error ε

t

Improving resolution of reference frame

Use the 91st to 120th frames to improve the 100th frame f_{100} in the movie

Affine Model:

$$f_{100+j} = \begin{bmatrix} c_{0,j} & c_{2,j} \\ c_{1,j} & c_{3,j} \end{bmatrix} f_{100} + \begin{bmatrix} c_{5,j} \\ c_{6,j} \end{bmatrix}, \quad j = \pm 1, \pm 2, \ldots$$

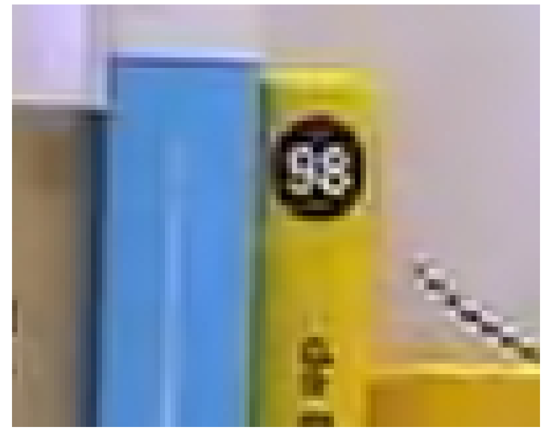
where $\{c_{i,j}\}_{i=1}^{6}$ are obtained by least-squares, and $c_{5,j}$ and $c_{6,j}$ the displacement errors for frame f_{100+j}.

704-by-578 image of f_{100} by bilinear interpolation

704-by-578 image of f_{100} by tight frame method using 20 frames from the movie

Video Enhancement

Bilinear method



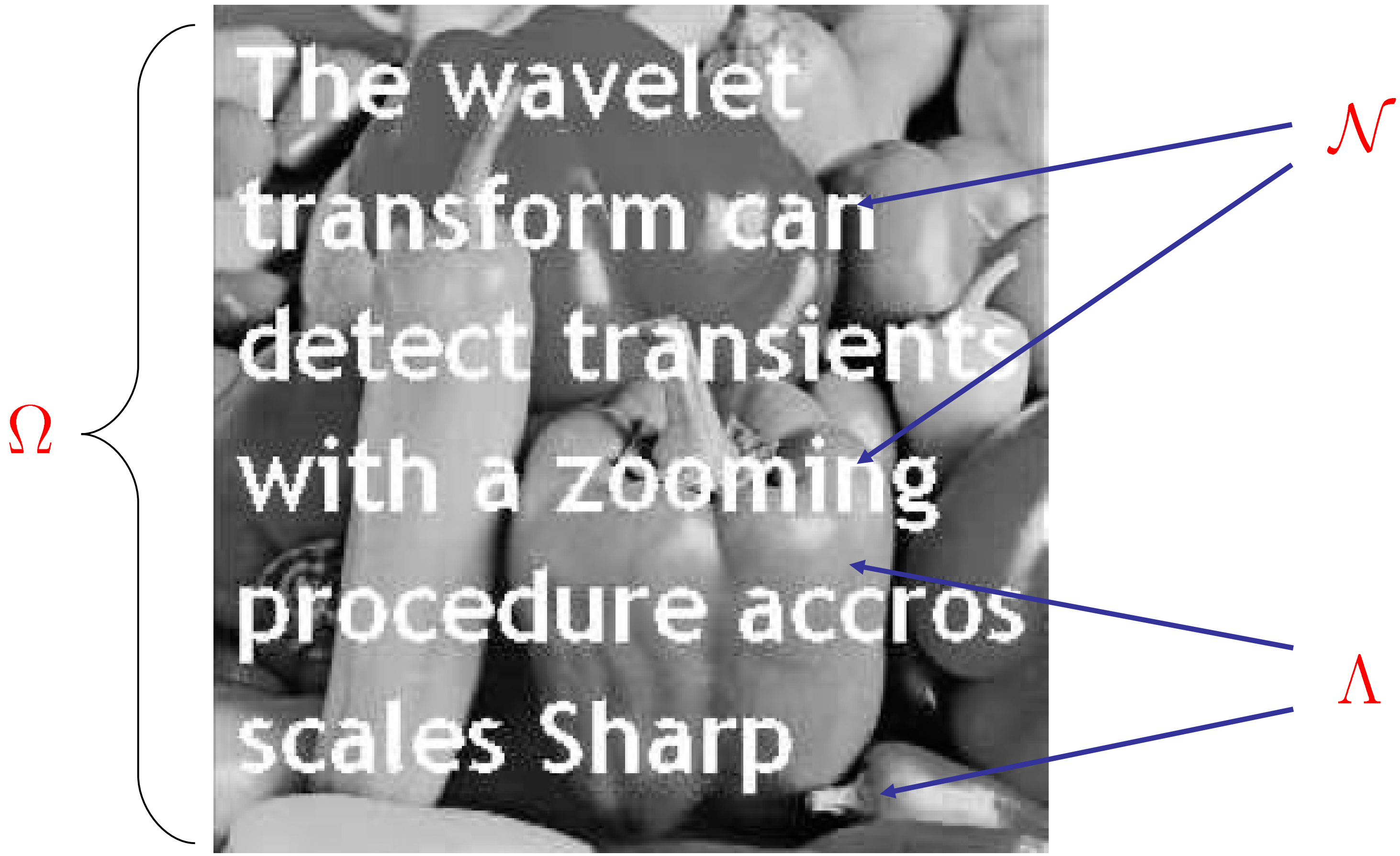
Tight frame method

C., Shen, and Xia, ACHA, 2007

Outline

1. Tight Frames
2. Inpainting
3. Impulse Noise Removal
4. High Resolution Image Reconstruction
5. Video Enhancement
6. Extension
7. Convergence Analysis
8. Combining PDE and Framelets

General Framework for Missing Data Recovery



General Framework for Missing Data Recovery in Image Domain

In both inpainting and impulse noise removal, we have missing data in the image domain. Our goal is to find the image f in Ω from the data y given only on Λ, i.e., solve f from

$$\mathcal{P}_\Lambda f = \mathcal{P}_\Lambda y.$$

The iteration we used is

$$f^{(r+1)} = (\mathcal{I} - \mathcal{P}_\Lambda)\mathcal{A}^* \mathcal{T}_\lambda \mathcal{A} f^{(r)} + \mathcal{P}_\Lambda y.$$

Tight Frame Algorithm

For $r = 0, 1, \ldots,$ until convergence:

1. Compute $c^{(r)} = \mathcal{A} f^{(r)}$.

2. Threshold $c^{(r)}$ by threshold λ to get $\tilde{c}^{(r)}$.

3. Reconstruct $f^{(r+1)} = \mathcal{A}^* \tilde{c}^{(r)}$.

4. Data fitting: set

$$[f^{(r+1)}]_p = \begin{cases} [f^{(r+1)}]_p, & p \in \mathcal{N}, \\ [y]_p\,, & p \in \Lambda. \end{cases}$$

Convergence (Cai, C., Shen, ACHA 2008)

In image domain, f minimizes:

$$\min_{\mathcal{P}_\Lambda f=\mathcal{P}_\Lambda y}\{\min_c\{\tfrac{1}{2}\|\mathcal{A}f-c\|_2^2+\|\mathrm{diag}(\lambda)c\|_1\}\}.$$

In frequency domain, c minimizes:

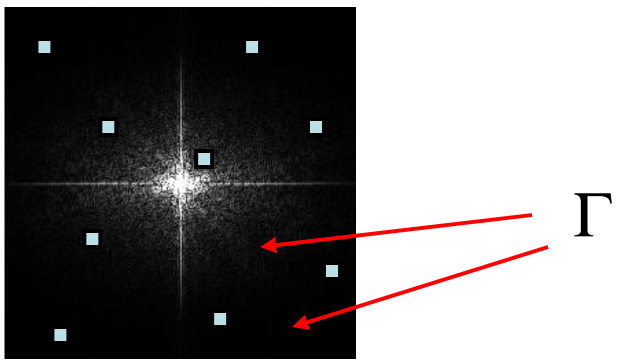
$$\min_c\{\tfrac{1}{2}\|\mathcal{P}_\Lambda(\mathcal{A}^*c)-\mathcal{P}_\Lambda y\|_2^2+\tfrac{1}{2}\|(I-\mathcal{A}\mathcal{A}^*)c\|_2^2+\|\mathrm{diag}(\lambda)c\|_1\}.$$

Convergence proved by convex analysis.

Extension to Frequency Domain Inpainting

- Problem Setting: find the image f from the framelet coefficients d on Γ, i.e., solve f from

$$\mathcal{P}_\Gamma \mathcal{A} f = \mathcal{P}_\Gamma d.$$



- Algorithm: the same idea as the image domain inpainting algorithm

$$f^{(r+1)} = \mathcal{A}^* \mathcal{T}_\lambda\big((\mathcal{I} - \mathcal{P}_\Gamma)\mathcal{A} f^{(r)} + \mathcal{P}_\Gamma d\big).$$

Tight Frame Algorithm

For $r = 0, 1, \ldots,$ until convergence:

1. Compute $c^{(r)} = \mathcal{A} f^{(r)}$.

2. Data fitting: set

$$[\hat{c}^{(r)}]_p = \begin{cases} [c^{(r)}]_p, & p \in \Omega \setminus \Gamma, \\ [d]_p\,, & p \in \Gamma. \end{cases}$$

3. Threshold $\hat{c}^{(r)}$ by threshold λ to get $\tilde{c}^{(r)}$.

3. Reconstruct $f^{(r+1)} = \mathcal{A}^* \tilde{c}^{(r)}$.

Convergence Results

In image domain, f minimizes:

$$\min_f \{\min_{c\in\mathbf{D}} \{\frac{1}{2}\|\mathcal{A}f - c\|_2^2 + \|\mathrm{diag}(\lambda)c\|_1\}\}.$$

with

$$\mathbf{D} = \{c : \mathcal{P}_\Gamma c = \mathcal{P}_\Gamma d\}.$$

In frequency domain, c minimizes:

$$\min_c \{\frac{1}{2}\|\mathcal{P}_\Gamma(c-d)\|_2^2 + \frac{1}{2}\|(I-\mathcal{A}\mathcal{A}^*)c\|_2^2 + \|\mathrm{diag}(\lambda)c\|_1\}.$$

Cai, C., Shen, Shen, Adv. Comp. Math. (2009)

Application 1: High-Resolution Image Reconstruction

Four low resolution images (64×64) of the same scene. Each shifted by sub-pixel length.

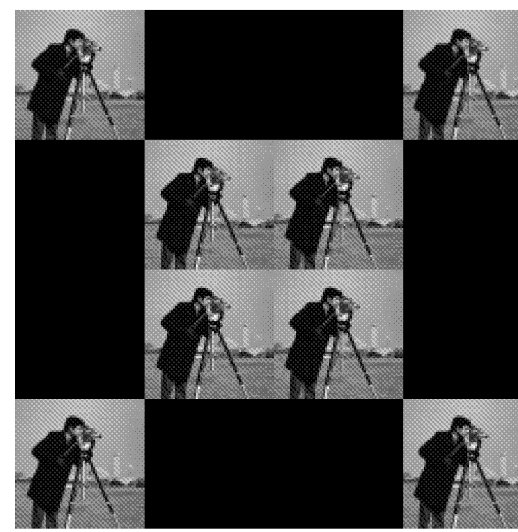
Construct a high-resolution image (128×128) from them

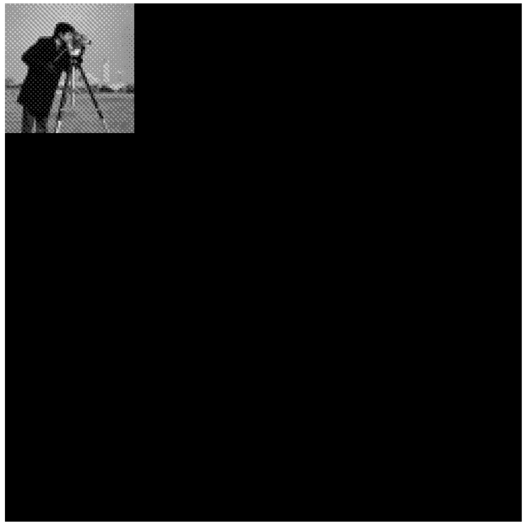
$$
\left[\begin{array}{cccccc}
 & & \vdots & & & \\
\cdots & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & \cdots & \cdots \\
\cdots & \cdots & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & \cdots \\
 & & \vdots & & & \color{red}{H_0} \\
\hline
 & & \vdots & & & \\
\cdots & -\frac{\sqrt{2}}{4} & 0 & \frac{\sqrt{2}}{4} & \cdots & \cdots \\
\cdots & \cdots & -\frac{\sqrt{2}}{4} & 0 & \frac{\sqrt{2}}{4} & \cdots \\
 & & \vdots & & & \color{red}{H_1} \\
\hline
 & & \vdots & & & \\
\cdots & -\frac{1}{4} & \frac{1}{2} & -\frac{1}{4} & \cdots & \cdots \\
\cdots & \cdots & -\frac{1}{4} & \frac{1}{2} & -\frac{1}{4} & \cdots \\
 & & \vdots & & & \color{red}{H_2}
\end{array}\right]
f =
\left[\begin{array}{c}
\bullet \\ \bullet \\ \bullet \\ \bullet \\ \hline \circ \\ \circ \\ \circ \\ \circ \\ \hline \circ \\ \circ \\ \circ \\ \circ
\end{array}\right]
\qquad \Gamma
$$

Application 2: Super-Resolution Image Reconstruction

$$
\left[\begin{array}{cccccc}
 & & \vdots & & & \\
\cdots & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & \cdots & \cdots \\
\cdots & \cdots & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & \cdots \\
 & & \vdots & & & H_0 \\
\hline
 & & \vdots & & & \\
\cdots & -\frac{\sqrt{2}}{4} & 0 & \frac{\sqrt{2}}{4} & \cdots & \cdots \\
\cdots & \cdots & -\frac{\sqrt{2}}{4} & 0 & \frac{\sqrt{2}}{4} & \cdots \\
 & & \vdots & & & H_1 \\
\hline
 & & \vdots & & & \\
\cdots & -\frac{1}{4} & \frac{1}{2} & -\frac{1}{4} & \cdots & \cdots \\
\cdots & \cdots & -\frac{1}{4} & \frac{1}{2} & -\frac{1}{4} & \cdots \\
 & & \vdots & & & H_2
\end{array}\right]
f = \begin{array}{c}
\bullet \\ \circ \\ \bullet \\ \circ \\
\hline
\circ \\ \circ \\ \circ \\ \circ \\
\hline
\circ \\ \circ \\ \circ \\ \circ
\end{array}
$$

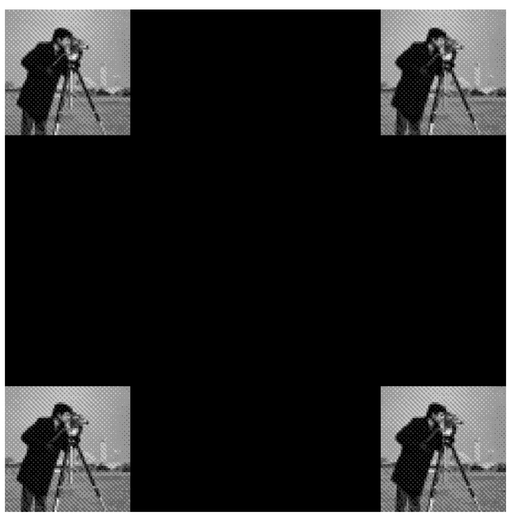
Γ





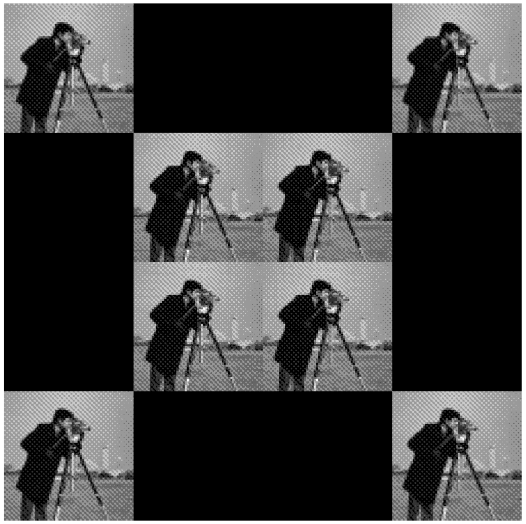
1 LR image

recovered



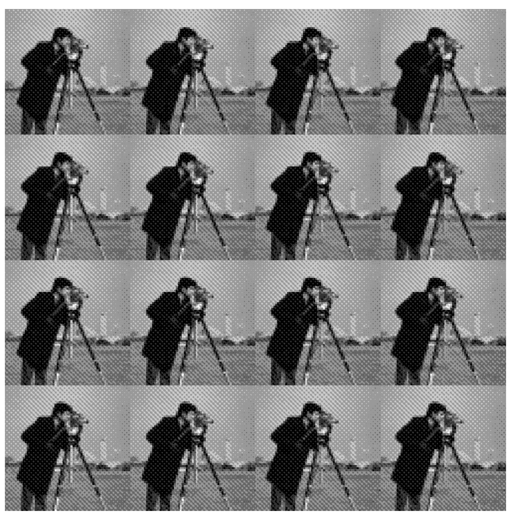
4 LR images

recovered



8 LR image

recovered



16 LR images

recovered

Application 3: Video Still Enhancement

One of the frame in a video

Before enhancement

After enhancement

The 100th frame
f_{100} *in the movie*

Affine Model:

$$f_{100+j} = \begin{bmatrix} c_{0,j} & c_{2,j} \\ c_{1,j} & c_{3,j} \end{bmatrix} f_{100} + \begin{bmatrix} c_{5,j} \\ c_{6,j} \end{bmatrix}, \;\; j = \pm 1, \pm 2, \ldots$$

where $\{c_{i,j}\}_{i=1}^{6}$ are obtained by least-squares, and $c_{5,j}$ and $c_{6,j}$ the displacement errors for frame f_{100+j}.

$$
\left[\begin{array}{cccccc}
 & & \vdots & & & \\
\cdots & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & \cdots & \cdots \\
\cdots & \cdots & \frac{1}{4}-\epsilon & \frac{1}{2} & \frac{1}{4}+\epsilon & \cdots \\
 & & \vdots & & & \\
\hline
 & & \vdots & & & \\
\cdots & -\frac{\sqrt{2}}{4} & 0 & \frac{\sqrt{2}}{4} & \cdots & \cdots \\
\cdots & \cdots & -\frac{\sqrt{2}}{4} & 0 & \frac{\sqrt{2}}{4} & \cdots \\
 & & \vdots & & & \\
\hline
 & & \vdots & & & \\
\cdots & -\frac{1}{4} & \frac{1}{2} & -\frac{1}{4} & \cdots & \cdots \\
\cdots & \cdots & -\frac{1}{4} & \frac{1}{2} & -\frac{1}{4} & \cdots \\
 & & \vdots & & &
\end{array}\right]
\qquad
f = \begin{array}{c}
\bullet \\ \circ \\ \bullet \\ \bullet \\ \hline \circ \\ \circ \\ \circ \\ \circ \\ \hline \circ \\ \circ \\ \circ \\ \circ \\ \hline
\end{array}
\qquad \Gamma
$$

Application 4: Infrared Imaging

Consider

$$s = f(x, y) + \eta(x, y, t)$$

where

- f : the celestial source
- η : background noise

The goal is to extract the weak astronomical signal f from the large background noise η.

Chop-and-Nod Procedure

First obtain signal s_P at pixel $P = (x, y)$ at t:

$$s_P = f(x, y) + \eta(x, y, t).$$

Then move telescope at Δ-distance away to obtain signals s_{P_+} and s_{P_-}:

$$\begin{aligned} s_{P_+} &= f(x, y + \Delta) + \eta(x, y + \Delta, t'), \\ s_{P_-} &= f(x, y - \Delta) + \eta(x, y - \Delta, t''). \end{aligned}$$

(Δ is chopping amplitude.)

Chop-and-Nod Procedure

Two chopped images are nodded:

$$\Delta s_+ = s_P - s_{P_+} = f(x,y) - f(x,y+\Delta) + \Delta\eta_+,$$

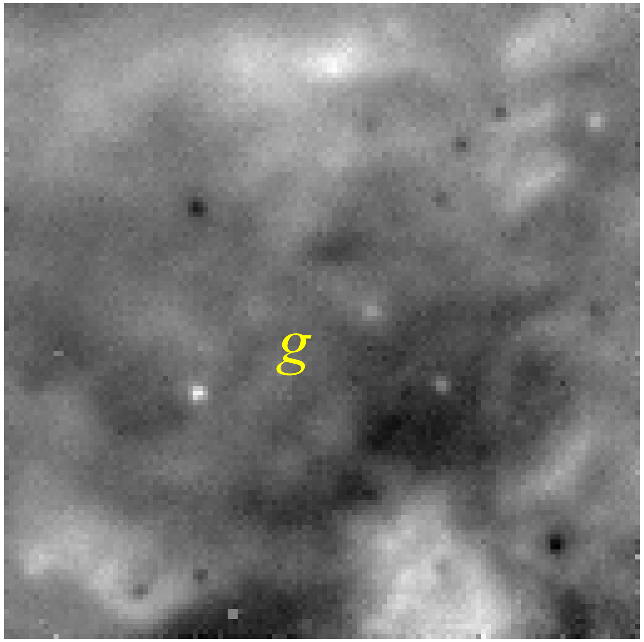
$$\Delta s_- = s_{P_-} - s_P = f(x,y-\Delta) - f(x,y) + \Delta\eta_-.$$

Then the chopped-nodded image is

$$\begin{aligned} g(x,y) &= \Delta s_+ - \Delta s_- \\ &= -f(x,y-\Delta) + 2f(x,y) - f(x,y+\Delta) \\ &\qquad +\Delta\eta_+ - \Delta\eta_- \\ &\approx -f(x,y-\Delta) + 2f(x,y) - f(x,y+\Delta). \end{aligned}$$

Application 4: Infrared Imaging

$$f \xrightarrow[\text{Nodding}]{\text{Chopping \&}}$$



$$g(i,m) = -f(j,m-\Delta) + 2f(j,m) - f(j,m+\Delta)$$

$$
\left[\begin{array}{cccccc}
 & & \vdots & & & \\
\cdots & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & \cdots & \cdots \\
\cdots & \cdots & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & \cdots \\
 & & \vdots & & & \color{red}{H_0} \\
\hline
 & & \vdots & & & \\
\cdots & -\frac{\sqrt{2}}{4} & 0 & \frac{\sqrt{2}}{4} & \cdots & \cdots \\
\cdots & \cdots & -\frac{\sqrt{2}}{4} & 0 & \frac{\sqrt{2}}{4} & \cdots \\
 & & \vdots & & & \color{red}{H_1} \\
\hline
 & & \vdots & & & \\
\cdots & -\frac{1}{4} & \frac{1}{2} & -\frac{1}{4} & \cdots & \cdots \\
\cdots & \cdots & -\frac{1}{4} & \frac{1}{2} & -\frac{1}{4} & \cdots \\
 & & \vdots & & & \color{red}{H_2}
\end{array}\right]
\quad f = \begin{array}{c}
\circ \\ \circ \\ \circ \\ \circ \\ \hline \circ \\ \circ \\ \circ \\ \circ \\ \hline \bullet \\ \bullet \\ \bullet \\ \bullet
\end{array}
\quad \longrightarrow \Gamma
$$

Minimization Properties

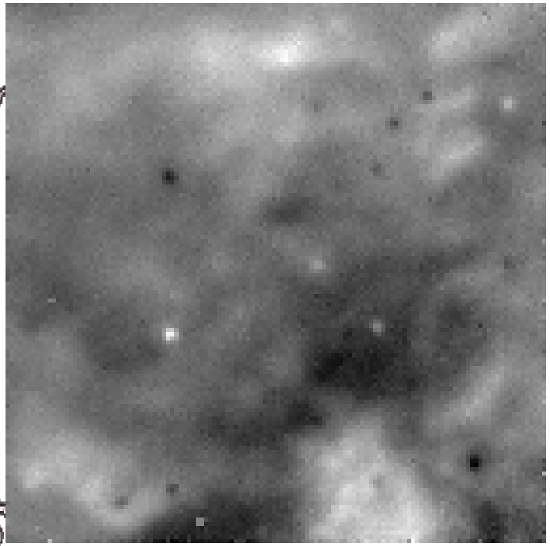
The limit minimizes

$$\min_{f \in \mathbf{P}_+} \left\{ \min_{c \in \mathbf{C}} \{ \tfrac{1}{2} \|\mathcal{A} f - c\|_2^2 + \|\mathrm{diag}(\lambda) c_{H_0}\|_1 \right.$$

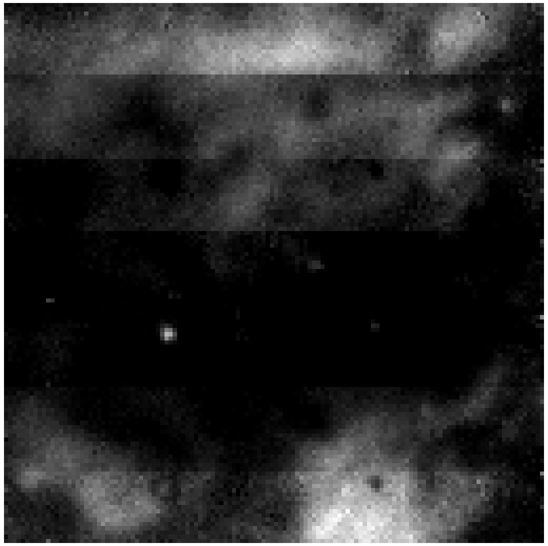
$$\left. + \|\mathrm{diag}(\lambda) c_{H_1}\|_1 \} \right\},$$

where

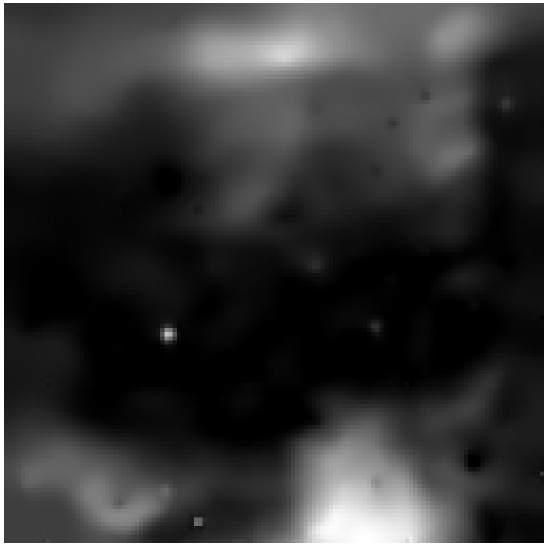
- $\mathbf{P}_+ = \{f : f \geq 0 \text{ componentwise}\}$,
- $\mathbf{C} = \{c_{H_2} \text{ satisfies constraints}\}$.



Observed Image from United Kingdom Infrared Telescope

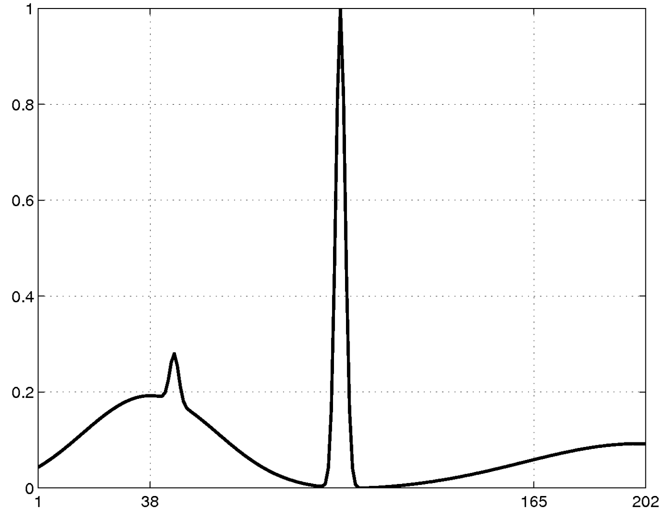


Reconstruction by Projected Landweber's Iteration

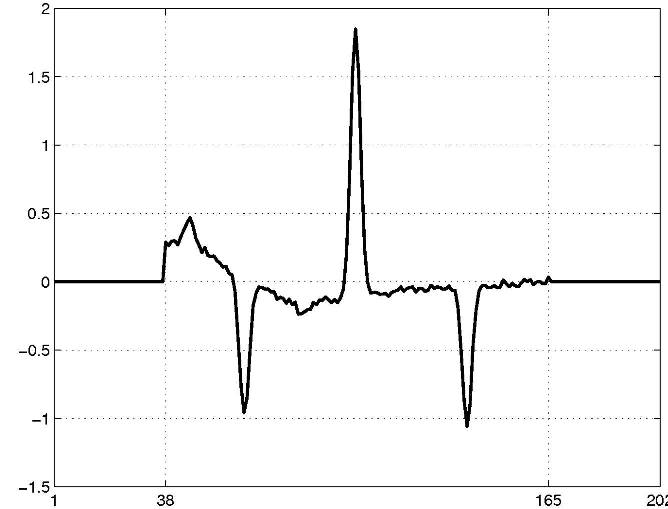


Reconstruction by Framelet-Based Method

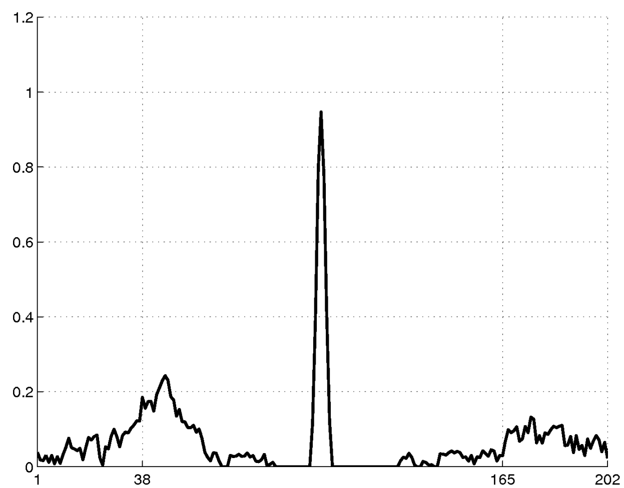
Cai, C., Shen, Shen, SISC, 2008.



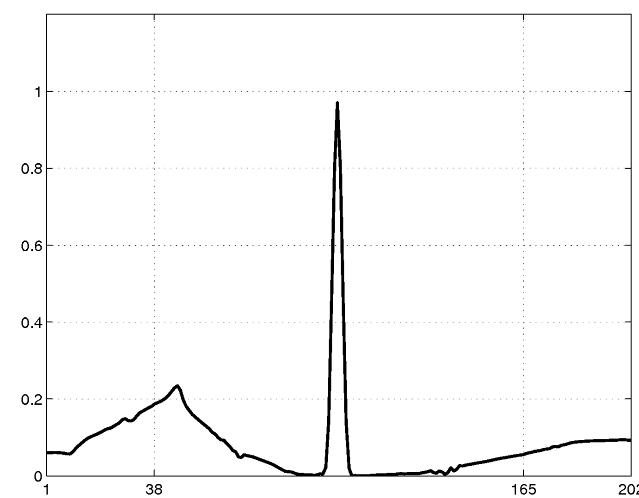
Original



Chopped & Nodded



Landweber



Framelet

More General Framework

Inpainting for missing data both in image and frequency domain.

$$f^{(r+1)} = (\mathcal{I}-\mathcal{P}_\Lambda)\Big(\mathcal{A}^*\mathcal{T}_\lambda\big((\mathcal{I}-\mathcal{P}_\Gamma)\mathcal{A}f^{(r)}+\mathcal{P}_\Gamma d\big)\Big)+\mathcal{P}_\Lambda y.$$

- The algorithm has many potential applications.
- Convergence and minimization properties can also be proved by convex analysis.

Cai, C., Shen, Shen, Numerisch Mathematatik (2009)

Outline

1. Tight Frames
2. Inpainting
3. Impulse Noise Removal
4. High Resolution Image Reconstruction
5. Video Enhancement
6. Extension
7. Convergence Analysis
8. Combining PDE and Framelets

Inpainting Algorithm

Let

$$\mathcal{P}_\Lambda[i,j] = \begin{cases} 1 & \text{if } i = j \in \Lambda, \\ 0 & \text{otherwise.} \end{cases}$$

Then

$$f = \mathcal{P}_\Lambda f + (\mathcal{I} - \mathcal{P}_\Lambda) f.$$

Perfect reconstruction formula gives

$$f = \mathcal{P}_\Lambda y + (\mathcal{I} - \mathcal{P}_\Lambda)\mathcal{A}^*\mathcal{A} f.$$

Natural Algorithm:

$$f^{(r+1)} = \mathcal{P}_\Lambda y + (\mathcal{I} - \mathcal{P}_\Lambda)\mathcal{A}^*\mathcal{A} f^{(r)}$$

Inpainting Algorithm

Threshold $\mathcal{A}f^{(r)}$:

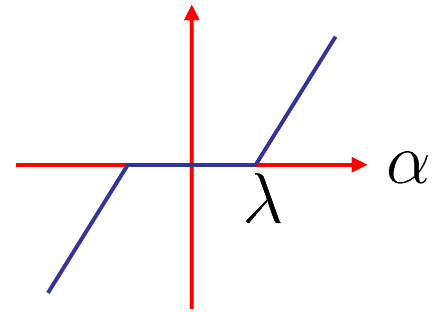
Framelet Inpainting Algorithm:

1. Set an initial guess $f^{(0)}$
2. Iterate on r until convergence:

$$f^{(r+1)} = \mathcal{P}_\Lambda y + (\mathcal{I} - \mathcal{P}_\Lambda)\mathcal{A}^*\mathcal{T}_\lambda(\mathcal{A}f^{(r)})$$

The thresholding allows information from Λ to permeate into $\mathcal{N} = \Omega \setminus \Lambda$.

Thresholding Operator



Soft thresholding operator $\mathcal{T}_\lambda$:

$$\mathcal{T}_\lambda((\alpha_1, \ldots, \alpha_l, \ldots)^T) = (t_{\lambda_1}(\alpha_1), \ldots, t_{\lambda_l}(\alpha_l), \ldots)^T,$$

where

$$\begin{aligned} t_{\lambda_i}(\alpha_i) &= \operatorname{sign}(\alpha_i)\max(|\alpha_i| - \lambda_i, 0) \\ &= \begin{cases} \alpha_i - \lambda_i & \text{if } \alpha_i > \lambda_i, \\ \alpha_i + \lambda_i & \text{if } \alpha_i \leq -\lambda_i, \\ 0 & \text{if } |\alpha_i| \leq \lambda_i. \end{cases} \end{aligned}$$

A possible choice of λ is Donoho's $\sigma\sqrt{2\ln N}$.

Thresholding Operator

Key observation:

$$t_{\lambda_i}(\alpha) = \arg\min_{\gamma} \left\{ \tfrac{1}{2}(\gamma - \alpha)^2 + |\lambda_i \gamma| \right\}, \ \alpha, \lambda_i \in \mathbb{R}.$$

Hence

$$\begin{aligned} \tilde{c}^{(r)} &\equiv \mathcal{T}_\lambda(\mathcal{A}f^{(r)}) \\ &= [t_{\lambda_1}([\mathcal{A}f^{(r)}]_1), \ldots, t_{\lambda_j}([\mathcal{A}f^{(r)}]_j), \ldots] \\ &= \arg\min_{c} \left\{ \tfrac{1}{2}\|\mathcal{A}f^{(r)} - c\|_2^2 + \|\mathrm{diag}(\lambda)c\|_1 \right\}. \end{aligned}$$

Projection Operator

Define $\mathbf{C} = \{g : \mathcal{P}_\Lambda g = \mathcal{P}_\Lambda y\}$. Then

$$\begin{aligned}\mathcal{P}_{\mathbf{C}}(f) &= \arg\min_{g\in\mathbf{C}} \tfrac{1}{2}\|f-g\|_2^2 \\ &= \arg\min_{g} \left\{\tfrac{1}{2}\|f-g\|_2^2 + \iota_{\mathbf{C}}(g)\right\}\end{aligned}$$

where

$$\iota_{\mathbf{C}}(g) \equiv \begin{cases} 0, & g \in \mathbf{C}, \\ +\infty, & \text{otherwise}. \end{cases}$$

Key Observation:

$$\mathcal{P}_{\mathbf{C}}(f) = \mathcal{P}_\Lambda y + (\mathcal{I} - \mathcal{P}_\Lambda) f.$$

Projection Operator

Proof: We have

$$\begin{aligned} & \mathcal{P}_\Lambda(\mathcal{P}_\Lambda y + (\mathcal{I} - \mathcal{P}_\Lambda) f) = \mathcal{P}_\Lambda y \\ \Longrightarrow \quad & \mathcal{P}_\Lambda y + (\mathcal{I} - \mathcal{P}_\Lambda) f \in \mathbf{C}. \end{aligned}$$

Moreover, $\forall g \in \mathbf{C}$, we have

$$\begin{aligned} \|g - f\|_2^2 &= \sum_{i \in \Lambda} (g_i - f_i)^2 + \sum_{i \in \Omega \setminus \Lambda} (g_i - f_i)^2 \\ &= \sum_{i \in \Lambda} (y_i - f_i)^2 + \sum_{i \in \Omega \setminus \Lambda} (g_i - f_i)^2 \\ &\geq \sum_{i \in \Lambda} (y_i - f_i)^2 = \|\mathcal{P}_\Lambda y + (\mathcal{I} - \mathcal{P}_\Lambda) f - f\|_2^2. \end{aligned}$$

Alternate Direction Minimization

Recall that

$$f^{(r+1)} = \mathcal{P}_\Lambda y + (\mathcal{I} - \mathcal{P}_\Lambda)\mathcal{A}^*\mathcal{T}_\lambda(\mathcal{A}f^{(r)})$$

and $\tilde{c}^{(r)} = \mathcal{T}_\lambda(\mathcal{A}f^{(r)})$.

Hence our framelet algorithm becomes:

$$\begin{aligned} f^{(r+1)} &= \mathcal{P}_\Lambda y + (\mathcal{I} - \mathcal{P}_\Lambda)\mathcal{A}^*\tilde{c}^{(r)} \\ &= \mathcal{P}_{\mathbf{C}}(\mathcal{A}^*\tilde{c}^{(r)}) \\ &= \arg\min_g \left\{ \frac{1}{2}\|\mathcal{A}^*\tilde{c}^{(r)} - g\|_2^2 + \iota_{\mathbf{C}}(g) \right\}. \end{aligned}$$

Minimization Functional

We show that $\lim_{r\to\infty}\{f^{(r)}\} = f^\star$ exists and is a minimizer of

$$\min_{f\in\mathbf{C}}\{\min_{c}\{\tfrac{1}{2}\|\mathcal{A}f - c\|_2^2 + \|\mathrm{diag}(\lambda)c\|_1\}\};$$

and that $c^\diamond = \mathcal{T}_\lambda(\mathcal{A}f^\star)$ is a minimizer of

$$\min_{c}\Big\{\tfrac{1}{2}\|\mathcal{P}_\Lambda(\mathcal{A}^* c) - \mathcal{P}_\Lambda y\|_2^2 + \\ + \tfrac{1}{2}\|(\mathcal{I} - \mathcal{A}\mathcal{A}^*)c\|_2^2 + \|\mathrm{diag}(\lambda)c\|_1\Big\}.$$

Proximal Forward-Backward Splitting

Combettes and Wajs (SIMMS, 2005)

Theorem. Let F_1 be convex, lower semi-continuous, and F_2 be convex with a $1/b$-Lipschitz continuous gradient with $b > 1/2$. Then the minimization problem

$$\min_f \{F_1(f) + F_2(f)\}$$

can be solved by the iteration:

$$f^{(r+1)} = \mathrm{prox}_{F_1}(f^{(r)} - \nabla F_2(f^{(r)})),$$

if minimum exists.

Proximal Forward-Backward Splitting

Combettes and Wajs (SIMMS, 2005)

- Theorem does not guarantee existence of minimum. It has to be proved separately.
- No convergence rate is given.
- prox is Moreaus proximity operator:

$$\operatorname{prox}_{\varphi}(f) \equiv \arg\min_{g}\{\tfrac{1}{2}\|f-g\|_2^2+\varphi(g)\}.$$

Proximal Operators

- For examples:

$$\begin{aligned}
\tilde{c}^{(r)} &= \mathcal{T}_\lambda(\mathcal{A}f^{(r)}) \\
&= \arg\min_c \left\{ \tfrac{1}{2}\|\mathcal{A}f^{(r)} - c\|_2^2 + \|\mathrm{diag}(\lambda)c\|_1 \right\}. \\
&= \mathrm{prox}_{\|\mathrm{diag}(\lambda)\cdot\|_1}(\mathcal{A}f^{(r)})
\end{aligned}$$

and

$$\begin{aligned}
f^{(r+1)} &= \mathcal{P}_{\mathbf{C}}(\mathcal{A}^*\tilde{c}^{(r)}) \\
&= \arg\min_g \left\{ \tfrac{1}{2}\|\mathcal{A}^*\tilde{c}^{(r)} - g\|_2^2 + \iota_{\mathbf{C}}(g) \right\} \\
&= \mathrm{prox}_{\iota_{\mathbf{C}}}(\mathcal{A}^*\tilde{c}^{(r)})
\end{aligned}$$

An Equivalent Formulation

We now show that our tight frame method

$$f^{(r+1)} = \mathcal{P}_\Lambda y + (\mathcal{I} - \mathcal{P}_\Lambda)\mathcal{A}^*\mathcal{T}_\lambda\mathcal{A}f^{(r)}$$

is a proximal forward-backward splitting:

$$f^{(r+1)} = \mathrm{prox}_{F_1}(f^{(r)} - \nabla F_2(f^{(r)})),$$

for some minimization problem

$$\min_f\{F_1(f) + F_2(f)\}.$$

Minimization Functional for f

For our $\{f^{(r)}\}$, the functional is:

$$\min_f \{\underbrace{\iota_{\mathbf{C}}(f)}_{F_1} + \underbrace{\operatorname{env}_{\|\operatorname{diag}(\lambda)\cdot\|_1}(\mathcal{A}f)}_{F_2}\}$$

where

- $\iota_{\mathbf{C}}$ is the indicator function of the constraints:

$$\iota_{\mathbf{C}}(f) = \begin{cases} 0 & \text{if } \mathcal{P}_\Lambda f = \mathcal{P}_\Lambda y, \\ +\infty & \text{if otherwise,} \end{cases}$$

  i.e. data-fitting, and

Minimization Functional for f

- env is the envelope function:

$$\mathrm{env}_{\varphi}(c) \equiv \min_{d}\{\tfrac{1}{2}\|c-d\|_2^2 + \varphi(d)\}.$$

It gives the minimum value of $\mathrm{prox}_{\varphi}(\cdot)$.

- Note that F_2:

$$\mathrm{env}_{\|\mathrm{diag}(\lambda)\cdot\|_1}(c) \equiv \min_{d}\{\tfrac{1}{2}\|c-d\|_2^2 + \|\mathrm{diag}(\lambda)d\|_1\}$$

comes from the soft-thresholding operator:

$$\mathcal{T}_{\lambda}(c) = \arg\min_{d}\{\tfrac{1}{2}\|c-d\|_2^2 + \|\mathrm{diag}(\lambda)d\|_1\}.$$

Proximal Forward-backward Splitting

Proof: We have

$$\begin{aligned} f^{(r+1)} &= \mathrm{prox}_{\iota_{\mathbf{C}}}(\mathcal{A}^* \tilde{c}^{(r)}) \\ &= \mathrm{prox}_{\iota_{\mathbf{C}}}[\mathcal{A}^* \mathrm{prox}_{\|\mathrm{diag}(\lambda)\cdot\|_1}(\mathcal{A} f^{(r)})] \end{aligned}$$

Combettes and Wajs:

For any φ: $\nabla_c[\mathrm{env}_\varphi(c)] \equiv c - \mathrm{prox}_\varphi(c)$. Hence:

$$\nabla_f[\mathrm{env}_{\|\mathrm{diag}(\lambda)\cdot\|_1}(\mathcal{A} f)] = \mathcal{A}^*(\mathcal{A} f - \mathrm{prox}_{\|\mathrm{diag}(\lambda)\cdot\|_1}(\mathcal{A} f)).$$

Let $\xi = \|\text{diag}(\lambda) \cdot \|_1$. We have

$$\begin{aligned} f^{(r+1)} &= \text{prox}_{\iota_{\mathbf{C}}}(\mathcal{A}^* \text{prox}_\xi(\mathcal{A} f^{(r)})) \\ &= \text{prox}_{\iota_{\mathbf{C}}}(f^{(r)} - \mathcal{A}^*\mathcal{A} f^{(r)} + \mathcal{A}^* \text{prox}_\xi(\mathcal{A} f^{(r)})) \\ &= \text{prox}_{\iota_{\mathbf{C}}}(f^{(r)} - \mathcal{A}^*[\mathcal{A} f^{(r)} - \text{prox}_\xi(\mathcal{A} f^{(r)})]) \\ &= \text{prox}_{\iota_{\mathbf{C}}}(f^{(r)} - \nabla[\text{env}_\xi(\mathcal{A} f^{(r)})]) \\ &= \text{prox}_{F_1}(f^{(r)} - \nabla F_2(f^{(r)})) \end{aligned}$$

where

- $F_1(f) = \iota_{\mathbf{C}}(f)$,
- $F_2(f) = \text{env}_\xi(\mathcal{A} f) = \text{env}_{\|\text{diag}(\lambda)\cdot\|_1}(\mathcal{A} f)$.

Limiting Functional

The proximal forward-backward splitting converges to:

$$\begin{aligned}
& \min_f \{F_1(f) + F_2(f)\} \\
= \quad & \min_f \{\iota_{\mathbf{C}}(f) + \mathrm{env}_\xi(\mathcal{A}f)\} \\
= \quad & \min_{f \in \mathbf{C}} \{\mathrm{env}_\xi(\mathcal{A}f)\} \\
= \quad & \min_{f \in \mathbf{C}} \{\min_c \{\tfrac{1}{2}\|\mathcal{A}f - c\|_2^2 + \|\mathrm{diag}(\lambda)c\|_1\}\}.
\end{aligned}$$

Minimization Problem for f

In image domain, $\{f^{(r)}\}$ in

$$f^{(r+1)} = \mathcal{P}_\Lambda y + (\mathcal{I} - \mathcal{P}_\Lambda)\mathcal{A}^*\mathcal{T}_\lambda \mathcal{A} f^{(r)}$$

is the forward-backward splitting for minimizing:

$$\min_{f \in \mathbf{C}} \{\min_c \{\tfrac{1}{2}\|\mathcal{A}f - c\|_2^2 + \|\mathrm{diag}(\lambda)c\|_1\}\}.$$

Note that

- $f \in \mathbf{C} \equiv \{\mathcal{P}_\Lambda f = \mathcal{P}_\Lambda y\}$: data-fitting
- $\|\mathrm{diag}(\lambda)c\|_1$: sparsity of c ($\Leftarrow$ piecewise smooth image)

Lipschitz Constant

To prove convergence, we have to verify that F_2 has a $1/b$-Lipschitz continuous gradient with $b > 1/2$.

Combettes and Wajs: for any φ:

- $$\nabla[\mathrm{env}_\varphi(c)] = c - \mathrm{prox}_\varphi(c),$$

- $$\|(c - \mathrm{prox}_\varphi(c)) - (d - \mathrm{prox}_\varphi(d))\|_2 \le \|c - d\|_2.$$

Lipschitz Constant

Recall we have, with $\xi \equiv \|\text{diag}(\lambda) \cdot \|_1$,

$$\nabla F_2(f) = \nabla[\text{env}_\xi(\mathcal{A}f)] = \mathcal{A}^*(\mathcal{A}f - \text{prox}_\xi(\mathcal{A}f)),$$

and

$$\begin{aligned} & \|\nabla F_2(f) - \nabla F_2(g)\|_2 \\ = \; & \|\mathcal{A}^*(\mathcal{A}f - \text{prox}_\xi(\mathcal{A}f)) - \mathcal{A}^*(\mathcal{A}g - \text{prox}_\xi(\mathcal{A}g))\|_2 \\ \leq \; & \|\mathcal{A}^*\|_2 \|(\mathcal{A}f - \text{prox}_\xi(\mathcal{A}f)) - (\mathcal{A}g - \text{prox}_\xi(\mathcal{A}g))\|_2 \\ \leq \; & \|\mathcal{A}^*\|_2 \|\mathcal{A}(f-g)\|_2 \leq \|f-g\|_2, \end{aligned}$$

i.e. $F_2(f)$ has 1-Lipschitz continuous gradient

Minimization Problem for c

In frequency domain, $\tilde{c}^{(r)} \equiv \mathcal{T}_\lambda(\mathcal{A}f^{(r)})$ can be rewritten as

$$\tilde{c}^{(r)} = \arg\min_c \{\tfrac{1}{2}\|\mathcal{A}f^{(r)} - c\|_2^2 + \|\mathrm{diag}(\lambda)c\|_1\}.$$

This is equivalent to

$$\begin{aligned}\tilde{c}^{(r)} &= \mathrm{prox}_{\|\mathrm{diag}\lambda\cdot\|_1}\{\mathcal{A}f^{(r)}\} \\ &= \mathrm{prox}_{\|\mathrm{diag}(\lambda)\cdot\|_1}\{\mathcal{A}[\mathrm{prox}_{\iota_{\mathbf{C}}}(\mathcal{A}^*\tilde{c}^{(r-1)})]\}\end{aligned}$$

This is another forward-backward splitting:

$$\tilde{c}^{(r)} = \mathrm{prox}_{F_3}\{\tilde{c}^{(r-1)} - \nabla F_4(\tilde{c}^{(r-1)})\}.$$

Proof: We have

$$
\begin{aligned}
& \mathcal{A} f^{(r+1)} \\
= \;& \mathcal{A}[\mathcal{P}_\Lambda y + (\mathcal{I} - \mathcal{P}_\Lambda)\mathcal{A}^* \tilde{c}^{(r)}] \\
= \;& \tilde{c}^{(r)} - \tilde{c}^{(r)} + \mathcal{A}\mathcal{P}_\Lambda y + \mathcal{A}\mathcal{A}^* \tilde{c}^{(r)} - \mathcal{A}\mathcal{P}_\Lambda \mathcal{A}^* \tilde{c}^{(r)} \\
= \;& \tilde{c}^{(r)} - [(\mathcal{I} - \mathcal{A}\mathcal{A}^*)\tilde{c}^{(r)} + \mathcal{A}\mathcal{P}_\Lambda(\mathcal{P}_\Lambda \mathcal{A}^* \tilde{c}^{(r)} - \mathcal{P}_\Lambda y)] \\
= \;& \tilde{c}^{(r)} - [(\mathcal{I} - \mathcal{A}\mathcal{A}^*)^2 \tilde{c}^{(r)} + \mathcal{A}\mathcal{P}_\Lambda(\mathcal{P}_\Lambda \mathcal{A}^* \tilde{c}^{(r)} - \mathcal{P}_\Lambda y)] \\
= \;& \tilde{c}^{(r)} - \nabla \left(\frac{1}{2} \|(\mathcal{I} - \mathcal{A}\mathcal{A}^*)\tilde{c}^{(r)}\|_2^2 + \frac{1}{2} \|\mathcal{P}_\Lambda \mathcal{A}^* \tilde{c}^{(r)} - \mathcal{P}_\Lambda y\|_2^2 \right)
\end{aligned}
$$

Thus

$$
\begin{aligned}
\tilde{c}^{(r+1)} &= \operatorname{prox}_{\|\operatorname{diag}\lambda \cdot\|_1} \{\mathcal{A} f^{(r+1)}\} \\
&= \operatorname{prox}_{\|\operatorname{diag}\lambda \cdot\|_1} \{\tilde{c}^{(r)} - \nabla F_4(\tilde{c}^{(r)})\}.
\end{aligned}
$$

Minimization Functional of c

In frequency domain, the minimization functional is:

$$\underbrace{\|\mathrm{diag}(\lambda)c\|_1}_{F_3} + \underbrace{\frac{1}{2}\|\mathcal{P}_\Lambda(\mathcal{A}^*c) - \mathcal{P}_\Lambda y\|_2^2 + \frac{1}{2}\|(I - \mathcal{A}\mathcal{A}^*)c\|_2^2}_{F_4}.$$

Note that

- $\|\mathrm{diag}(\lambda)c\|_1$: sparsity of c
- $\|\mathcal{P}_\Lambda(\mathcal{A}^*c) - \mathcal{P}_\Lambda y\|_2$: data-fitting
- $\|(I - \mathcal{A}\mathcal{A}^*)c\|_2$: c close to $\mathrm{Range}(\mathcal{A})$
- $c \in \mathrm{Range}(\mathcal{A})$: $\|c\|_1 \approx \|f\|_{B^\sigma_{1,1}}$.

Lipschitz Constant

By direct calculation:

$$
\begin{aligned}
& \|\nabla F_4(c) - \nabla F_4(d)\|_2 \\
= \; & \|\nabla[\tfrac{1}{2}\|\mathcal{P}_\Lambda \mathcal{A}^* c - \mathcal{P}_\Lambda y\|_2^2 + \tfrac{1}{2}\|(\mathcal{I} - \mathcal{A}\mathcal{A}^*)c\|_2^2] \\
& \quad -\nabla[\tfrac{1}{2}\|\mathcal{P}_\Lambda \mathcal{A}^* d - \mathcal{P}_\Lambda y\|_2^2 + \tfrac{1}{2}\|(\mathcal{I} - \mathcal{A}\mathcal{A}^*)d\|_2^2]\|_2 \\
= \; & \|[(\mathcal{I} - \mathcal{A}\mathcal{A}^*)c + \mathcal{A}\mathcal{P}_\Lambda(\mathcal{P}_\Lambda \mathcal{A}^* c - \mathcal{P}_\Lambda y)] \\
& \quad -[(\mathcal{I} - \mathcal{A}\mathcal{A}^*)d + \mathcal{A}\mathcal{P}_\Lambda(\mathcal{P}_\Lambda \mathcal{A}^* d - \mathcal{P}_\Lambda y)]\|_2 \\
= \; & \|(\mathcal{I} - \mathcal{A}(\mathcal{I} - \mathcal{P}_\Lambda)\mathcal{A}^*)(c - d)\|_2 \\
\leq \; & \|\mathcal{I} - \mathcal{A}(\mathcal{I} - \mathcal{P}_\Lambda)\mathcal{A}^*\|_2 \|c - d\|_2 \\
\leq \; & \|\mathcal{I} - \mathcal{A}(\mathcal{I} - \mathcal{P}_\Lambda)^2\mathcal{A}^*\|_2 \|c - d\|_2 \leq \|c - d\|_2.
\end{aligned}
$$

Existence of Minimizers

It remains to show that a minimum exists.

- Easy Case:
  Threshold every coefficients:

$$\lambda_i > 0, \quad 1 \leq i \leq N.$$

- Difficult Case:
  Similar to data-compression, do not threshold low-pass coefficients.

Numerical results show no significant differences

Existence of Minimizers (Easy Case)

Lemma. Let $\mathcal{A}$ be a tight frame system. Then the minimization problem has at least one minimizer.

Proof. Combettes and Wajs: Minimizer exists if $F_1(f) + F_2(f)$ is coercive, i.e. whenever

$$\|f\|_2 \to +\infty \Longrightarrow \{F_1(f) + F_2(f)\} \to +\infty.$$

Recall

$$F_2(f) = \min_c \{\tfrac{1}{2}\|\mathcal{A}f - c\|_2^2 + \|\mathrm{diag}(\lambda)c\|_1\},$$

where the minimizer is precisely $\mathcal{T}_\lambda(\mathcal{A}f)$.

Existence of Minimizers (Easy Case)

Hence

$$
\begin{aligned}
F_2(f) &\geq \|\mathrm{diag}(\lambda)\mathcal{T}_\lambda(\mathcal{A}f)\|_1 = \sum_{i=1}^{N} \lambda_i |t_{\lambda_i}(\mathcal{A}f)_i| \\
&\geq \lambda_{\min} \sum_{i=1}^{N} |t_{\lambda_i}(\mathcal{A}f)_i| \geq \lambda_{\min} \sum_{i=1}^{N} (|(\mathcal{A}f)_i| - \lambda_i) \\
&\geq \lambda_{\min}\|\mathcal{A}f\|_1 - \lambda_{\min}\lambda_{\max}N \\
&\geq \lambda_{\min}\|\mathcal{A}f\|_2 - \lambda_{\min}\lambda_{\max}N \\
&= \lambda_{\min}\|f\|_2 - \lambda_{\min}\lambda_{\max}N. \quad \square
\end{aligned}
$$

Convergence Result

Theorem. For any tightframe system, if we threshold every coefficients, then our tight frame algorithm $f^{(r)}$ converges to

$$\min_{f\in C}\{\min_{c}\{\tfrac{1}{2}\|\mathcal{A}f - c\|_2^2 + \|\mathrm{diag}(\lambda)c\|_1\}\},$$

while $\tilde{c}^{(r)} = \mathcal{T}_\lambda(\mathcal{A}f^{(r)})$ converges to

$$\min_{c}\{\|\mathrm{diag}(\lambda)c\|_1+\tfrac{1}{2}\|\mathcal{P}_\Lambda(\mathcal{A}^*c)-\mathcal{P}_\Lambda y\|_2^2+\tfrac{1}{2}\|(I-\mathcal{A}\mathcal{A}^*)c\|_2^2\}.$$

For wavelets,

$$\min_{c}\{\|\mathrm{diag}(\lambda)c\|_1 + \tfrac{1}{2}\|\mathcal{P}_\Lambda(\mathcal{A}^*c) - \mathcal{P}_\Lambda y\|_2^2\}.$$

Existence of Minimizers (Difficult Case)

Difficult Case: Low freqeuncy not threshold.

Let H be the filter of $\mathcal{A}$ where we do not threshold.

In inpainting, H is the low-pass filter H_0.

Assumptions on Existence of Minimizers:

(i) 1 is not an eigenvalue of H^*H; or

(ii) 1 is a simple eigenvalue of H^*H, and its eigenvector u satisfies $\mathcal{P}_\Lambda u \neq 0$.

Convergence Result

Lemma. Let H, the filter we do not threshold, satisfy the assumptions. Then

$$\min_{f\in C}\{\min_{c}\{\tfrac{1}{2}\|\mathcal{A}f - c\|_2^2 + \|\mathrm{diag}(\lambda)c\|_1\}\},$$

and

$$\min_{c}\{\|\mathrm{diag}(\lambda)c\|_1+\tfrac{1}{2}\|\mathcal{P}_\Lambda(\mathcal{A}^*c)-\mathcal{P}_\Lambda y\|_2^2+\tfrac{1}{2}\|(I-\mathcal{A}\mathcal{A}^*)c\|_2^2\}$$

both have at least one minimum.

Just show the functionals are coercive under the assumption

Convergence Result

Corollary. Under the assumptions, $f^{(r)}$ converges to

$$\min_{f \in C}\{\min_{c}\{\tfrac{1}{2}\|\mathcal{A}f - c\|_2^2 + \|\mathrm{diag}(\lambda)c\|_1\}\},$$

while $\tilde{c}^{(r)} = \mathcal{T}_\lambda(\mathcal{A}f^{(r)})$ converges to

$$\min_{c}\{\|\mathrm{diag}(\lambda)c\|_1 + \tfrac{1}{2}\|\mathcal{P}_\Lambda(\mathcal{A}^* c) - \mathcal{P}_\Lambda y\|_2^2 + \tfrac{1}{2}\|(I - \mathcal{A}\mathcal{A}^*)c\|_2^2\}.$$

Combettes and Wajs (2005): proximal forward backward splitting converges if minimum exists

Convergence Result for Splines

Lemma. For spline tightframe system with reflective boundary conditions, 1 is a simple eigenvalue of $H_0^* H_0$ with $\mathbf{1} = [1, \ldots, 1]^t$ as its eigenvector.

Clearly $\mathcal{P}_\Lambda \mathbf{1} \neq \mathbf{0}$. Thus Assumption (ii) is satisfied.

H_0 with reflective boundary conditions is a symmetric block Toeplitz-plus-Hankel with Toeplitz-plus-Hankel block matrices. It can always be diagonalized by discrete cosine transform, and eigenvalues and eigenvectors can be computed exactly (Ng, C., Tang, SISC (2000)).

Convergence Result for Splines

Theorem. For any spline tightframe systems, even if we do not threshold the low-pass coefficients, our tight frame algorithm $f^{(r)}$ converges to

$$\min_{f \in C}\{\min_{c}\{\tfrac{1}{2}\|\mathcal{A}f - c\|_2^2 + \|\mathrm{diag}(\lambda)c\|_1\}\},$$

while $\tilde{c}^{(r)} = \mathcal{T}_\lambda(\mathcal{A}f^{(r)})$ converges to

$$\min_{c}\{\|\mathrm{diag}(\lambda)c\|_1 + \tfrac{1}{2}\|\mathcal{P}_\Lambda(\mathcal{A}^*c) - \mathcal{P}_\Lambda y\|_2^2 + \tfrac{1}{2}\|(I - \mathcal{A}\mathcal{A}^*)c\|_2^2\}.$$

Cai, C., Shen, ACHA (2008)

$$\min_c\{\frac{1}{2}\|\mathcal{P}_\Lambda(\mathcal{A}^t c - y)\|_2^2 + \beta\|(\mathcal{I} - \mathcal{A}\mathcal{A}^t)c\|_2^2 + \|\mathrm{diag}(\lambda)c\|_1\}.$$

Sparsity

Regularity

- $\beta = 0$: Sparsity — *synthesis approach*

$$\min_c\{\frac{1}{2}\|\mathcal{P}_\Lambda(\mathcal{A}^t c - y)\|_2^2 + \|\mathrm{diag}(\lambda)c\|_1\}$$

  - *Fadili et.al., 2007; Daubechies et.al., 2007*

- $\beta = 1/2$: Sparsity + Regularity — *balanced approach*

$$\min_c\{\frac{1}{2}\|\mathcal{P}_\Lambda(\mathcal{A}^t c - y)\|_2^2 + \frac{1}{2}\|(\mathcal{I} - \mathcal{A}\mathcal{A}^t)c\|_2^2 + \|\mathrm{diag}(\lambda)c\|_1\}$$

  - Our tight-frame method SISC (2003)

- $\beta = \infty \Longrightarrow c \in \mathrm{Range}(\mathcal{A})$: Regularity — *analysis approach*

$$\min_f\{\frac{1}{2}\|\mathcal{P}_\Lambda(f - y)\|_2^2 + \|\mathrm{diag}(\lambda)\mathcal{A}f\|_1\}.$$

  - ~ total-variation inpainting for one level shift-invariant Haar wavelet transform

Outline

1. Tight Frames
2. Inpainting
3. Impulse Noise Removal
4. High Resolution Image Reconstruction
5. Video Enhancement
6. Extension
7. Convergence Analysis
8. Combining PDE and Framelets

Total Variation Revisit

We minimize

$$\min_{\mathbf{f}} F_{\mathbf{y}}(\mathbf{f}) = \min_{\mathbf{f}} \left\{ \frac{1}{2}\|\mathbf{f}-\mathbf{y}\|_{L^2}^2 + \beta \int |\nabla \mathbf{f}| \right\}.$$

The Euler-Lagrange equation is:

$$\mathbf{f} - \mathbf{y} + \beta \nabla \cdot \left(\frac{1}{|\nabla f|} \nabla \mathbf{f} \right) = \mathbf{0}.$$

254

Total Variation Revisit

We can rewrite it as:

$$\mathbf{f} - \mathbf{y} + \beta \nabla \cdot \left(\begin{bmatrix} \frac{1}{|\nabla f|} & 0 \\ 0 & \frac{1}{|\nabla f|} \end{bmatrix} \nabla \mathbf{f} \right) = \mathbf{0},$$

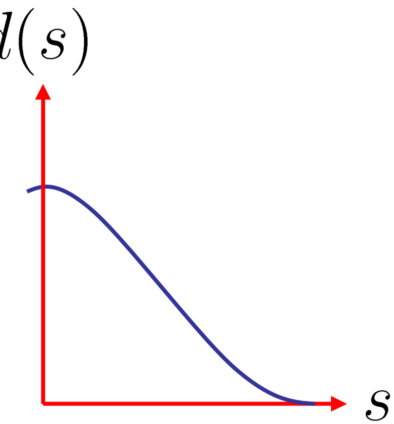
i.e. $\partial_x \mathbf{f}$ and $\partial_y \mathbf{f}$ are diffused isotropically according to $1/|\nabla f|$.

If $|\nabla f|$ large (edge), do not diffuse.

More general form:

$$\mathbf{f} - \mathbf{y} + \beta \nabla \cdot (\mathcal{D} \nabla \mathbf{f}) = \mathbf{0}.$$

Isotropic Diffusion



More generally:

$$\mathcal{D} = \begin{bmatrix} d(|\nabla \mathbf{f}|) & 0 \\ 0 & d(|\nabla \mathbf{f}|) \end{bmatrix},$$

where $d(|s|)$ is decreasing to 0 as $|s| \to \infty$.

For example (Weickert 1998):

$$d(|s|) = \begin{cases} 1 - e^{-\frac{3.31488}{(s/\alpha)^8}} & s > 0, \\ 1 & s = 0. \end{cases}$$

256

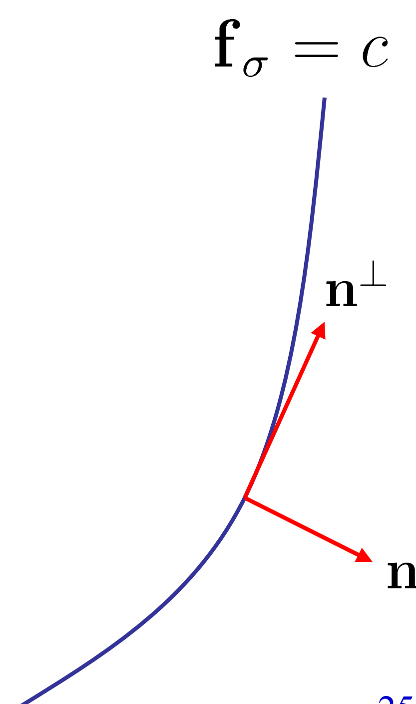
Anisotropic Diffusion

For anisotropic edge-enhancing diffusion:

$$\mathcal{D} = \mathcal{V} \begin{bmatrix} d(|\nabla \mathbf{f}_\sigma|) & 0 \\ 0 & 1 \end{bmatrix} \mathcal{V}^t,$$

where

- $\mathbf{f}_\sigma$ is a Gaussian-smoothed $\mathbf{f}$
- $\mathbf{n} = \dfrac{\nabla \mathbf{f}_\sigma}{|\nabla \mathbf{f}_\sigma|}$ (normal direction)
- $\mathcal{V} = [\mathbf{n} \;\; \mathbf{n}^\perp]$



257

Anisotropic Diffusion

Recall the Euler-Lagrange equation is:

$$\mathbf{f} - \mathbf{y} + \beta \nabla \cdot \left(\mathcal{V} \begin{bmatrix} d(|\nabla \mathbf{f}_\sigma|) & 0 \\ 0 & 1 \end{bmatrix} \mathcal{V}^t \nabla \mathbf{f} \right) = \mathbf{0}.$$

Thus $\nabla \mathbf{f}$ is decomposed along

$$\frac{\nabla \mathbf{f}_\sigma}{|\nabla \mathbf{f}_\sigma|} \quad \text{and} \quad \frac{\nabla \mathbf{f}_\sigma}{|\nabla \mathbf{f}_\sigma|}^\perp$$

and diffuses according to $d(|\nabla \mathbf{f}_\sigma|)$ and 1 respectively.

Haar Wavelet Shrinkage

Consider the Haar wavelet with filter:

$$h_0 = \frac{1}{2}[1, 1] \quad \text{and} \quad h_1 = \frac{1}{2}[1, -1].$$

Form the filter matrices H_0 and H_1 similarly.

Important observation:

$$\nabla \mathbf{f} = 2 \begin{pmatrix} H_0 \otimes H_1 \\ H_1 \otimes H_0 \end{pmatrix} \mathbf{f}.$$

Haar Wavelet Shrinkage

First level analysis operator:

$$\begin{pmatrix} H_0 \otimes H_0 \\ H_0 \otimes H_1 \\ H_1 \otimes H_0 \\ H_1 \otimes H_1 \end{pmatrix} \equiv \begin{pmatrix} H_{00} \\ H_{01} \\ H_{10} \\ H_{11} \end{pmatrix} \equiv \begin{pmatrix} H_{00} \\ H \\ H_{11} \end{pmatrix}.$$

Then form the m-level analysis operator $\mathcal{A}$ using the filters: $h_i^{(k)} = \frac{1}{2}(1, \underbrace{0, \ldots, 0}_{2^{k-1}-1}, (-1)^i)$.

We have $\mathcal{A}^t \mathcal{A} = \mathcal{A}\mathcal{A}^t = \mathcal{I}$.

Multi-level Decomposition

$H_{00}^{(3)}$

$$H_{00}^{(3)} H_{00}^{(2)} H_{00}^{(1)} \mathbf{f} = \mathbf{c}_{00}^{(3)}$$

$H_{00}^{(2)}$ $\quad H^{(3)}$

$$H^{(3)} H_{00}^{(2)} H_{00}^{(1)} \mathbf{f} = \mathbf{c}^{(3)}$$

$H_{11}^{(3)}$

$$H_{11}^{(3)} H_{00}^{(2)} H_{00}^{(1)} \mathbf{f} = \mathbf{c}_{11}^{(3)}$$

$H_{00}^{(1)}$ $\quad H^{(2)}$

$$H^{(2)} H_{0}^{(1)} \mathbf{f} = \mathbf{c}^{(2)}$$

$H_{11}^{(2)}$

$$H_{11}^{(2)} H_{0}^{(1)} \mathbf{f} = \mathbf{c}_{11}^{(2)}$$

$H^{(1)}$

$\mathbf{f}$

$$H^{(1)} \mathbf{f} = \mathbf{c}^{(1)} \quad = \frac{1}{2}\nabla \mathbf{f}$$

$H_{11}^{(1)}$

$$H_{11}^{(1)} \mathbf{f} = \mathbf{c}_{11}^{(1)}$$

Anisotropic Haar Shrinkage

Our anisotropic wavelet method is

$$f^{(r+1)} = \mathcal{P}_\Lambda y + (\mathcal{I} - \mathcal{P}_\Lambda)\mathcal{A}^* \mathcal{S}_\lambda(\mathcal{A} f^{(r)}).$$

Here the shrinkage is:

$$\begin{aligned} \mathcal{S}_\lambda(c_{00}^{(m)}) &= \mathcal{T}_\lambda(c_{00}^{(m)}), \\ \mathcal{S}_\lambda(c^{(k)}) &= \mathcal{V}^{(k)} \mathcal{T}_\lambda\left((\mathcal{V}^{(k)})^t c^{(k)}\right), \quad k = 1, \ldots, m, \\ \mathcal{S}_\lambda(c_{11}^{(k)}) &= \mathcal{T}_\lambda(c_{11}^{(k)}), \quad k = 1, \ldots, m. \end{aligned}$$

with $\mathbf{u} = [H_{01}^{(k)} \hat{f}_\sigma, H_{10}^{(k)} \hat{f}_\sigma]^t$, $\mathbf{n} = \mathbf{u}/|\mathbf{u}|$ and $\mathcal{V}^{(k)} = [\mathbf{n} \;\; \mathbf{n}^\perp]$.

Convergence

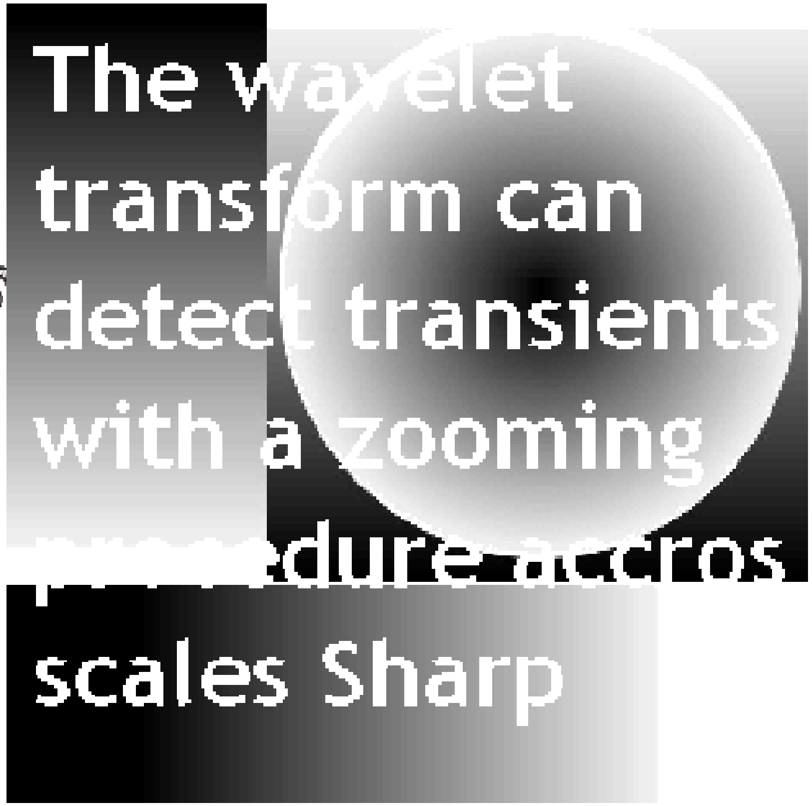
$f^{(r)}$ converges to the minimizer of

$$\min_{\mathcal{P}_\Lambda f = \mathcal{P}_\Lambda y} \left\{ \min_c \left[\frac{1}{2}\|\mathcal{A}f - c\|_2^2 + \sum_{k=1}^m \|\mathrm{diag}(\lambda^{(k)})(\mathcal{V}^{(k)})^t c^{(k)}\|_1 + \sum_{k=1}^m \|\mathrm{diag}(\lambda_{11}^{(k)}) c_{11}^{(k)}\|_1 \right] \right\}$$

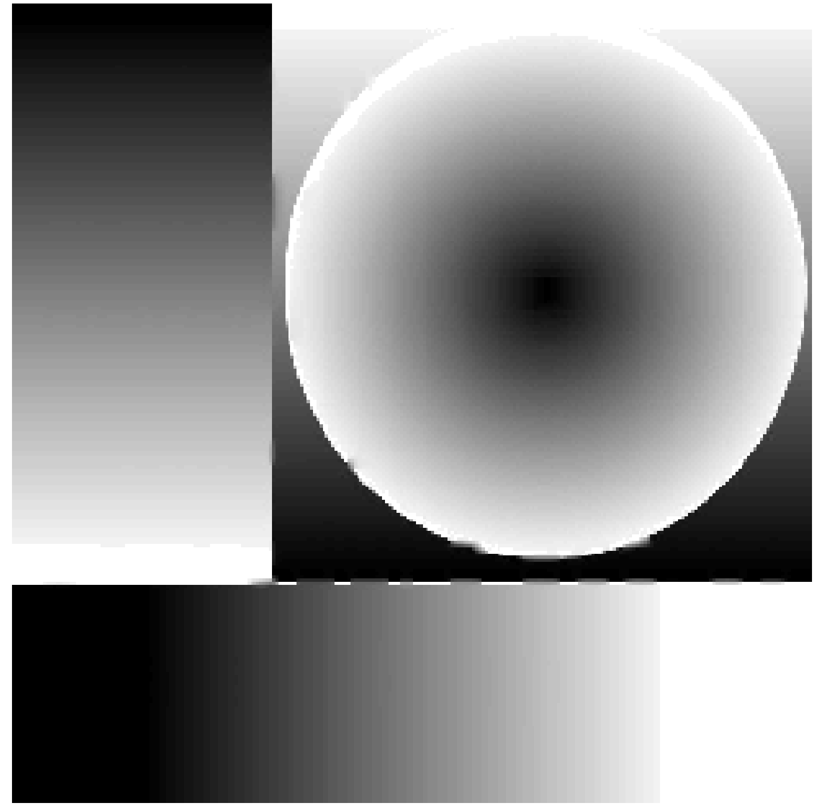
Proof follows the lines for framelet algorithm.

C., Setzer, Steidl, SIAM J. Imaging Sciences (2008)

Numerical Results

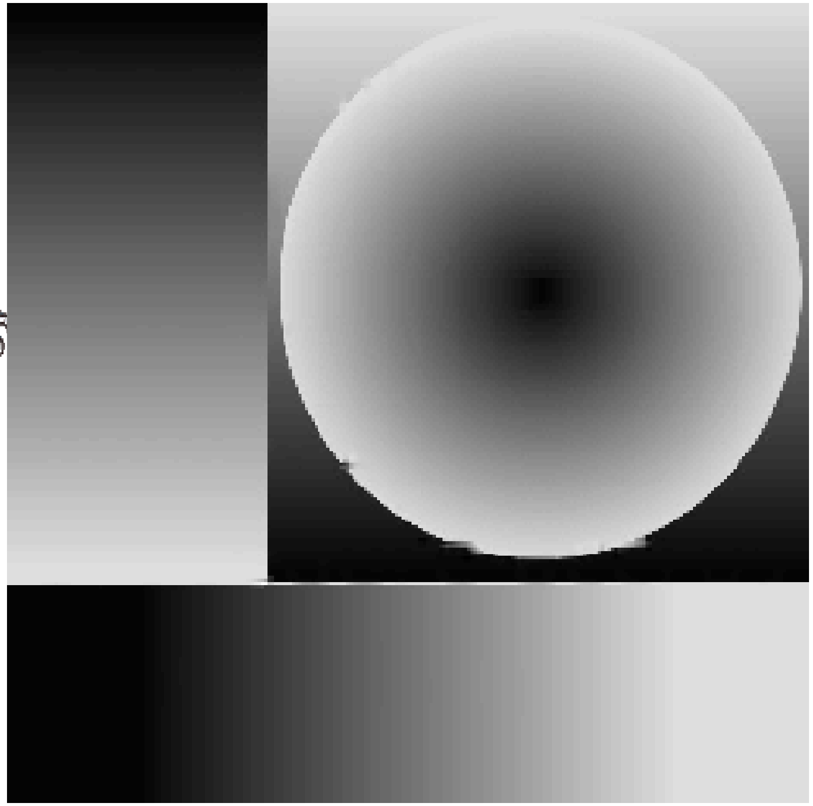


Corrupted

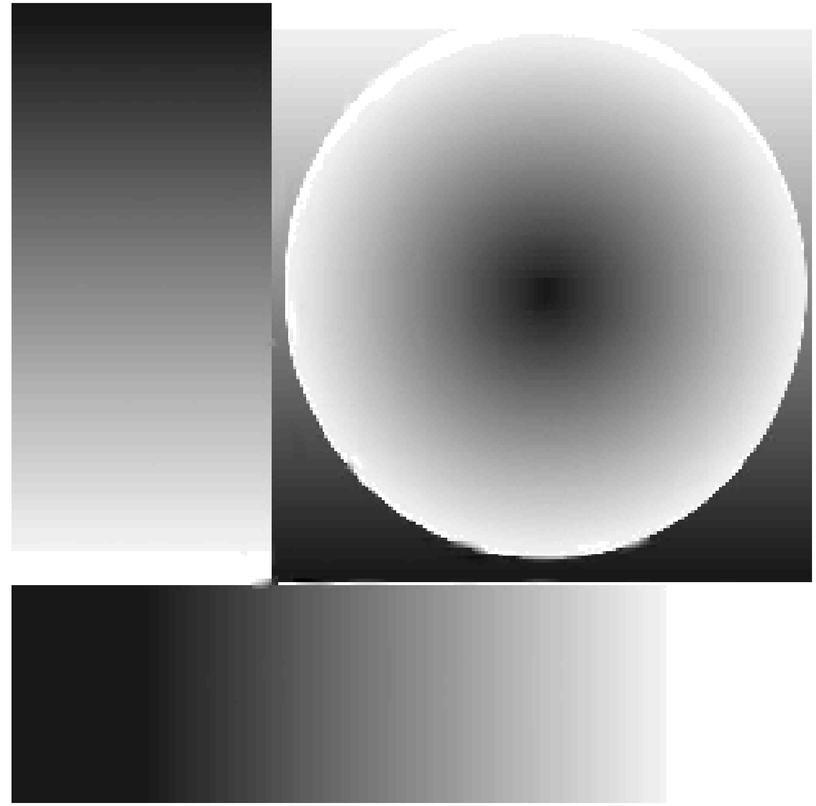


Framelet (33.27dB)

Numerical Results



Anisotropic Haar
$\mathcal{V}^{(k)}$ (36.60dB)



Anisotropic Haar
$\mathcal{V}^{(k,r)}$ (38.58dB)

Thank you !

www.math.cuhk.edu.hk/~rchan