Raster to Object Conversion Aided by Knowledge Based Image Processing[§]

Enrico Nardelli[1,3], Michelangelo Fossa[2(*)], Guido Proietti[1,3]

(1) IASI, National Research Council, Viale Manzoni 30, 00185 Roma, Italy, Tel.: +39-6-77.16.439, Fax.: +39-6-77.16.461, E-Mail: {nardelli,guido}@iasi.rm.cnr.it
(2) Algotech, Via Appia Nuova 310, 00183 Rome, Italy, tel. +39-6-702.6781, fax +39-6-702.3607
(3)Univ. of L'Aquila, Dept. of Pure and Applied Mathematics, Via Vetoio, Loc. Coppito, 67100 L'Aquila, Italy

Abstract

This paper describes the conceptual framework and the design of the prototype of a common toolkit for maps and office documents interpretation developed in the ROCKI project[1]. The conceptual framework features a novel three steps knowledge-based approach to the interpretation process. The first is the primitive interpretation which identifies elementary graphical objects. These are further processed (object interpretation) to obtain basic structured objects, which are finally (document interpretation) grouped into objects with associated semantics, obtaining a full description of the document. The design takes into account the need for feedback and cooperation among the different steps.

1: Introduction

Recent years have seen an increasing and increasing use of computers for office automation and document management. Nevertheless estimates show that about 90 to 95% of information is still on paper. The automated analysis of image documents is now a hot research topic and the conversion of information on paper into a structural high level description is becoming essential.

So far, emphasis has been laid more on the algorithmic part of the problem and less on the document understanding and recognition using knowledge [KAST88, SRIH86]. Concerning the specific context of map interpretation, the conversion of analog maps into a digital representation has been obstructed thus far by the high cost of the manual conversion. Although the prime consideration to automate such a task would be cost-effective reasons, the automation open the way to:

- reducing the errors in the conversion process
- verification and validation of the entered data
- increased speed of conversion and thus a higher throughput of the number of analog to digital maps.

Existing systems to help in the conversion from raster to digital representation described in literature so far require for the interpretation many run-time information and therefore need a continuous user assistance. A general problem suffered by these systems is that the conversion is restricted to a vectorization process [SUZU90]. Therefore, there is a lack of object attribute and contextual information that can be employed in the analysis process.

Other programs devised for the analysis of line-drawings or technical documents concentrate more on the algorithmic aspects, e.g. to separate text and lines [KAST90]. Although these systems make use of intelligent algorithms, the knowledge employed is not made explicit nor mentioned.

Last, there exist few programs claim to use knowledge base interpretation of technical documents, i.e. line drawings, maps etc. [ILG90, PULU90]. Hovewer, the conversion is at primitive level still requiring substantial interaction.

In the ROCKI project the problem of defining an architecture of a Common Toolkit that uses algorithmical and knowledge-based techniques to support interpretation of complex documents has been tackled and solved.

This paper describes the conceptual framework used for the definition of the interpretation process and the conceptual design of the ROCKI Common Toolkit. Implementation and experimentations are described in [FOSS93, NARD93a, NARD93b].

2: The Interpretation Framework

The conceptual framework for complex document interpretation is based on a three stages approach. The whole process can be described by showing (see figure 1) how data are transformed during the different phases.

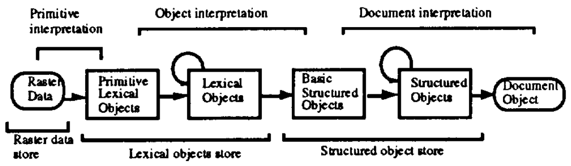

Fig.1 Data transformation during the interpretation process

Raster data are processed by the process "Primitive interpretation" which identifies the objects to be considered as graphical primitives for the application. Such "Primitive Lexical Objects" are elementary graphical objects, without any structure or meaning whatsoever. They can be seen as lexical elements of the structured information to be extracted by the system. Examples are characters, continuous lines, dotted lines, and so on.

§ Work here described was supported by the CEC under the ESPRIT project n.5376 ROCKI.

* M.Fossa current address is: SARITEL S.p.A, S.S. Pontina Km.29,100, Pomezia (Rome), Italy, tel.+39-6-91197.463, fax +39 -6-91197.600

[1]The partners of the ROCKI Consortium are Algotech (I), IASI - Istituto di Analisi dei Sistemi ed Informatica (I), IPA - Fraunhofer Gesellschaft (D), OCE Nederland B.V.(NL) and TNO Institute of Applied Physics (NL).

0-8186-4960-7/93 $3.00 © 1993 IEEE

Primitive Lexical Objects are processed by the process "Object interpretation", which groups and give structure to primitive objects. The resulting items can be either more complex lexical objects, which only represent graphical elements with no semantic value for the application, or higher level aggregations, called "Basic Structured Objects", which carry a meaning for the application. Basic Structured Objects are defined as the less complex objects which carry out a meaning from the point of view of the application. That is, they are composed only of an arbitrary amount of primitive objects, grouped together according to some hierarchy defined in the Structured Object definition. They are named "Basic" structured objects because all other structured objects are defined only as aggregations of simpler structured objects, without any reference to lexical elements. Examples of Basic Structured Objects are 'the set of characters representing a specific paragraph', 'the set of lines representing a given house', 'the set of lines representing a particular power supply network', and so on. Note that the Object Interpretation process groups Lexical Objects in more and more structured objects until a semantically defined item has been extracted. The whole process substitutes the set of Lexical Objects in the corresponding store with a set of Structured Objects: no semantically meaningless item is left for the subsequent phases.

As an example, let us consider recognizing a valve. First, the elementary graphical objects, that is the lines which make up the drawing of the valve, are grouped together to result in a predefined graphical pattern, which is still a Lexical Object. Then, this pattern is identified as the symbol of a valve. Note that from this point on, the process of recognizing more and more complex Structured Objects will make only reference to the concept of "valve symbol" instead of to its graphical pattern.

Structured Objects are finally processed by the process "Document Interpretation", which groups Structured objects in more and more complex ones and associates semantics to them, until a completely structured document is obtained. Many different levels of aggregation/abstraction may be present here, corresponding to the different logical views that may be required.

This general description may be completed and described by means of the Data Flow Diagram shown in figure 2, to take into account also the information which is needed to carry out the interpretation process.

At the level of Primitive interpretation, two more data stores are required, one being a base of algorithms for the recognition of graphics primitives, the other containing control information, specific to the considered type of document, to drive the process of primitive interpretation. Process "Primitive interpretation" accesses the algorithms base to download the algorithm(s) more suited to the logical type and physical characteristics of the document currently being processed and to the required type of interpretation. Examples of things found in the algorithms data store are 'an algorithm for identifying and following dotted lines', 'an algorithm for character recognition', 'an algorithm for continuous curved lines following', and so on. As well, specific control information is required to make both algorithm selection and primitive interpretation more efficient (e.g. 'patents are on 2 columns', 'street maps have no dotted lines', 'textual documents have no intersecting lines', and so on).

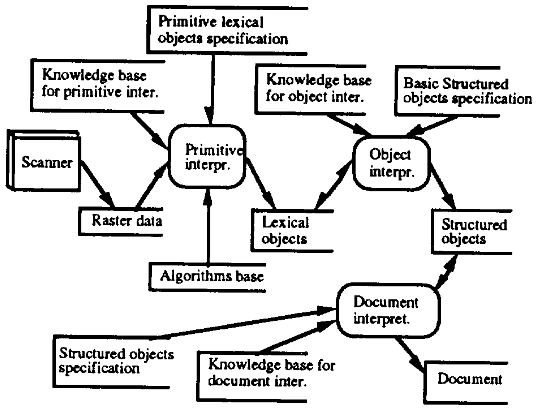
Primitive lexical objects specification
Knowledge base for primitive inter.
Knowledge base for object inter.
Basic Structured objects specification
Scanner
Primitive interpr.
Object interpr.
Raster data
Lexical objects
Structured objects
Algorithms base
Document interpret.
Structured objects specification
Knowledge base for document inter.
Document

Fig.2 DFD of the interpretation process.

At the level of Object interpretation, three more data stores are required, one containing <u>declarative</u> knowledge specifying types of lexical objects to be identified, in terms of aggregation of simpler lexical objects, the other two containing <u>procedural</u> knowledge, that is knowledge used to restrict the search space during the simpler lexical objects recognition phase. Example of the declarative knowledge are '<the PDL description of a patent>', 'a house is made up by four continuous lines in the shape of a rectangle and has an associated number'. Example of the procedural knowledge are 'the number associated with the house must be searched within 2 cm from the house border'. In object oriented terms, declarative knowledge is a kind of class definition whose instances correspond to the identified structured objects.

Declarative knowledge not only specifies the types (classes) of lexical objects to be identified by aggregation of simpler lexical objects but also specifies relations holding between the simpler lexical objects that contribute to the definition of more complex lexical objects (e.g. 'a private house is made up by four continous lines in the shape of a square adjacent to a continous double line'). Process "Object interpretation" accesses the declarative knowledge base to know how the object structures to be identified are defined.

The procedural knowledge base contains both general and application-dependent control strategies for driving the object interpretation. The general control strategy is part of the kernel of the toolkit (application independent part), while the application-dependent control strategy will be instantiated for each kind of application the toolkit will be specialized for (e.g. 'patent processing', 'road network management', and so on). An example of a general control strategy is simply to deal with the objects in the order they appear, while for a given application the control strategy could be tailored to deal with easy to find objects first and later to construct more complex ones. Also process "Object interpretation" accesses the procedural knowledge base to download the set of rules more suited to the type of document currently being processed and to the required type of interpretation.

At the level of Document interpretation, three additional data stores are also required, one containing declarative knowledge specifying types of structured objects to be identified, in terms of aggregation of simpler structured objects, the other two containing procedural knowledge, that is knowledge used to restrict the search space during the simpler structured objects recognition phase. Examples of the declarative knowledge are '<the DDL description of a patent>', 'a power supply network map is made up by continous primary and secondary distribution segments intersecting only at derivation points'.

Declarative knowledge not only specifies the types (classes) of higher level structured objects relevant for the application but also specifies relations holding between the lower level structured objects that contribute to the definition higher level ones (e.g. 'a private house is a house adjacent to a garden'). Process "Document interpretation" accesses the declarative knowledge base to know how the object structures to be identified are defined.

The procedural knowledge base contains both general and application dependent control strategies for driving the document interpretation. The general control strategy is part of the kernel of the toolkit (application independent part), while the application-dependent control strategy will be instanciated for each kind of application the toolkit will be specialized for (e.g. 'patent processing', 'road network management', and so on). Similar to the object interpretation the control strategy can be an exhaustive search. It could also be the case to specifically look for objects based on the context of already extracted objects. In addition, the process "Document interpretation" accesses the procedural knowledge base to download the set of rules more suited to the type of document currently being processed and to the required type of interpretation.

As it can be seen from the above description, process "Document interpretation" is analogous to "Object interpretation", since they both produce more complex objects from simpler ones. The reason for having them as two different processes is that the latter always starts from lexical objects, which carry no meaning, while the former always starts from structured objects, which are meaningful to the application.

3: The Conceptual Design

The Conceptual Design is given in terms of Objects and Actions: objects are the conceptual components of the application, while actions may be seen as the messages exchanged among objects. A sketch of the architecture of the Toolkit is given in fig.3.
An asynchronous communication mechanism in the form of an object called Event Base is introduced to take into account the need for feedback and re-iteration of part of the interpretation process to solve complex and misleading cases. Events are created by higher stages of the interpretation process when a dead-end is found in understanding objects and ask to lower stages to reconsider previous interpretations of objects and to revise them on the basis of different hypothesis.

3.1: Objects related to Primitive Interpretation

The most relevant Objects concerning the Primitive Interpretation (PI onwards) are:
- The Primitive Interpreter itself that extracts Primitive Lexical Objects from the Raster DataSet. The Interpreter is seen as an inference engine driven by a KB.
- The KB for PI that provides to Primitive Interpreter the strategies to run and control Alghoritms taken by a suited Algorithms Library.
- The Algorithms Library for PI: it is a set of methods issued by the Primitive Interpreter to extract the Primitive Lexical Objects applying the Algorithms to the Raster Dataset, and to extract specific Knowledge about the current interpretation process to be inserted in the Facts Base.
- The Facts Relevant to PI that are updated by the algorithms of the library and are read by the Interpreter.
- The Primitive Lexical Objects Specifications that are a collection of elementary graphical symbols meaningful for an application used by the algorithms of the base .
- The Raster Data Archive that is a set of fields representing the different raster datasets that is used as input for the Interpretation Process.
- The Lexical Objects Base which is created and updated by the Interpretation Process and it is intended as an active object that may return specific information about the stored lexical objects.

3.2: Objects related to Object Interpretation

The objects involved in the object interpretation process (OI onwards) are:
- The Object Interpreter which extracts the Basic Structured Objects from the Primitive Lexical Objects. The strategies and techniques followed for the extraction are defined in the OI Techniques Library. The Object Interpreter updates the Lexical Object Base (by aggregating simpler lexical objects) and creates the Structured Objects Base.
- The OI Techniques Library that contains the methods for the extraction of Basic Structured Objects; it reads the Basic Structured Objects Specifications and the Lexical Objects Specifications and it updates the base of the Facts Relevant to the Object Interpreter.
- The Facts Relevant to the Object Interpreter which is a base of facts that can be used by the interpreter. Such facts correspond to properties of the processed document that could make the OI process itself more efficient. This base of facts is updated by both the Object Interpreter as well as by the Primitive Interpreter.
- The KB for OI which provides the Object Interpreter with the strategies to control the application of the techiques

extracted from the Techniques Library. The KB contains both general and application-dependent strategies.

The Lexical Objects Base that is used as input by the interpretation process that also updates the base itself.

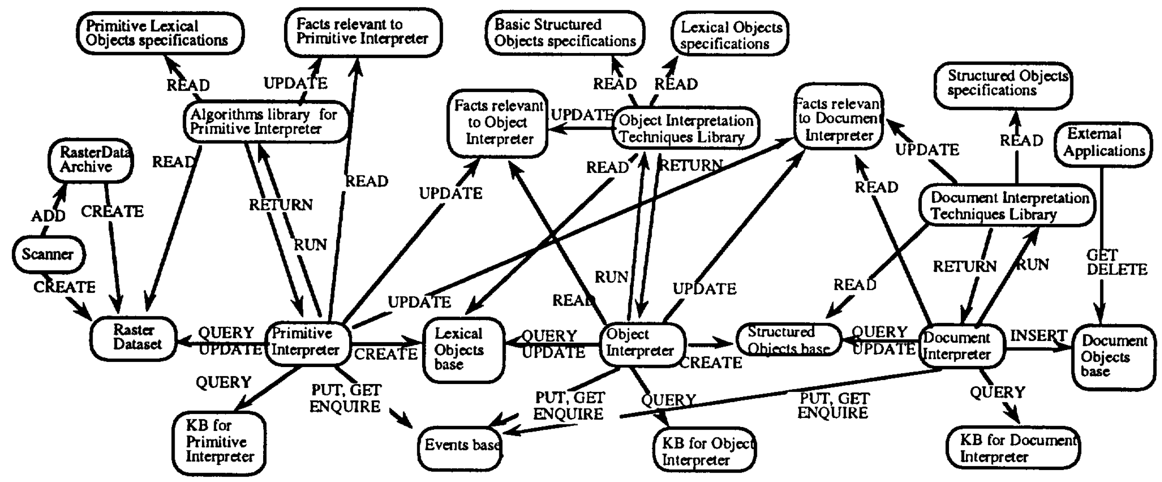
Primitive Lexical Objects specifications
Facts relevant to Primitive Interpreter
Basic Structured Objects specifications
Lexical Objects specifications
Structured Objects specifications
External Applications
Algorithms library for Primitive Interpreter
Facts relevant to Object Interpreter
Object Interpretation Techniques Library
Facts relevant to Document Interpreter
Document Interpretation Techniques Library
RasterData Archive
Scanner
Raster Dataset
Primitive Interpreter
Lexical Objects base
Object Interpreter
Structured Objects base
Document Interpreter
Document Objects base
KB for Primitive Interpreter
Events base
KB for Object Interpreter
KB for Document Interpreter
READ
UPDATE
RETURN
RUN
ADD
CREATE
QUERY
PUT, GET
ENQUIRE
INSERT
GET
DELETE

Fig.3 Main objects and their interactions

3.3: Objects releted to Document Intepretation

Finally the objects involved within the Document Interpretation Process (DI onwards) are:

- The Document Interpreter which extracts Structured Objects from the Basic Structured Objects applying suitable strategies that control the execution of methods and algorithms stored in the DI Techniques Library. The Document Interpreter reads and updates the Structured Objects Base and it updates also the Document Objects Base (which can be used by external applications).
- The DI Techniques Library which contains a set of techniques for interpretation from Structured Objects to Document Objects. It is intended as an active object that runs the techniques and algorithms which read the Structured Objects Specifications to perform their tasks.
- The Facts Relevant to DI that are a base of facts used by the interpreter to support the interpretation process. Facts can be added to the base by any of the three interpreters and consist of information contained in the processed document that can make the DI easier or more efficient.
- Finally, the KB for DI which supports the Document Interpreter with strategies to choose and to control the application of techniques for the aggregation of structured objects into document objects.

4: Conclusion

In this paper the conceptual approach of a toolkit for the interpretation and analysis of complex documents has been presented. Problems arising when designing a knowledge based system for document interpretation (due to the variety of typology of documents) have been discussed. The general approach based on three interpretation subprocesses and the conceptual design of the toolkit (based on objects and actions among the objects) have been also described.

Acknowledgments

We thank for their contribution to the project: Franco Arcieri, Thomas Bantel, Jan Hendrik Burger, Giorgio Gambosi, Jan Gerritsen, Jan den Hartog, Karen Hanse, Edwin Jansen, Ton ten Kate, Dick Schmidt.

References

[FOSS93] M.Fossa, E.Nardelli, G.Proietti, *A multistage approach to complex document interpretation*, submitted to 2nd Workshop on Document Analysis and Understanding.

[ILG90] Ilg M., *Knowledge-based understanding of road maps and other line images*; Tenth Int Conf Pattern Recognition: 282-284 (1990)

[KAST88] Kasturi R., Bow S.T., El-Masri W., Shah J., Gattiker J.R., Mokate U.B., *A system for recognition and description of graphics*, Ninth Inter Conf Pattern Recognition; Rome, Italy:255-259 (1988)

[KAST90] Kasturi R., Bow S.T., El-Masri W., Shah J., Gattiker J.R., Mokate U.B., Honnenanahalli S., *A system for interpretation of line drawings*; IEEE Trans Pattern Anal Mach Intel; 12:978-991 (1990)

[NARD93a] E.Nardelli, G.Proietti, *An algorithmic approach to the interpretation of floorplan maps*, IASI Tech. Rep, Sept.93.

[NARD93b] E.Nardelli, G.Proietti, *Advanced techniques for cadastral map interpretation*, IASI Tech. Rep, Sept.93.

[PULU90] Puluti P., Tascini G., *Interpreting Technical Drawings*, Comp J; 33:417-423 (1990)

[SRIH86] Srihari S.N., Zack G.W., *Document Image Analysis*, Eighth Inter Conf Pattern Recognition; Paris, France: 434-436 (1986)

[SUZU90] Suzuki S., Yamada T., *MARIS: map recognition input system*, Pattern Recognition; (1990)