AUTOMATIC PROCESSING OF DOCUMENTS AND BANK CHEQUES

JOSEPH NASSIF SAID

A THESIS
IN
THE DEPARTMENT
OF
COMPUTER SCIENCE

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY AT
CONCORDIA UNIVERSITY
MONTREAL, QUEBEC, CANADA

November, 1997

0-612-40313-0

Abstract

Automatic Processing of Documents and Bank Cheques

Joseph N. Said, Ph.D.
Concordia University, 1997

Automatic processing of documents with the purpose to scan different documents, recognize them, extract, and process different data items obtained from them could be achieved by top-down and bottom-up approaches. The former processes documents starting from the document class and ends with the pixel representation of the different items that should be extracted. The latter, however, processes documents in a reversed manner. In this thesis, a top-down formal approach for automatic processing of documents and bank cheques is proposed. This approach will view a document as a hierarchy of related items: (a) the background which contains simple or complex scenes that should be eliminated, and (b) the foreground which contains (i) base lines that must be removed and (ii) handwritten data, such as the date, the legal amount, and the courtesy amount, that should be extracted with minimum distortion.

The novelty of this new approach is to eliminate the background, first, by introducing a new recursive dynamic thresholding technique that could be used globally or locally on a given cheque image. As a second step, base lines that intersect the handwritten data are recognized and removed with the challenge of minimizing the distortion on the extracted items. Two methods are proposed to tackle this difficulty. The first method detects the handwritten data that intersects with the base lines that should be eliminated and uses morphological and topological processing to identify and fill the gaps resulting from the elimination of the detected base lines. The second method proposed a new dynamic morphological processing technique which acts as a detector and a preserver of the handwritten data that intersect with the base lines. The second method highly increased the efficiency of item extraction by more than 80% and enhanced the quality of the extracted items when combined with local processing techniques.

In a step to study the reliability of the proposed top-down automatic item extraction system, a quantitative analysis technique is investigated and an experimental study is performed comparing the top-down formal approach with another newly developed bottom-up approach using the same training set of 500 real-life bank cheques and two testing sets of 200 bank cheques obtained from the CENPARMI database. The purpose of the quantitative performance analysis technique is to subject the extracted items of the top-down and the bottom-up approaches to the same item processing system that is able to recognize these corresponding items and provide quantitative results to indicate the reliability of both approaches. The experimental results showed that the reliability of the top-down approach on the training set, first testing set, and second testing set are 89.20%, 87.91%, and 90.10% respectively while those on the bottom-up approach are 91.35%, 91.30%, and 93.10% respectively.

Finally, in a step towards the construction of a highly reliable system, a feasibility study has been conducted by combining both approaches. The result is quite encouraging and a reliability of 97.09% has been achieved when these two systems are combined.

Acknowledgements

This thesis would have been impossible without the encouragement and support of My two supervisors Dr. C. Y. Suen and Dr. M. Cheriet. I would like to thank them for their helpful discussion and guidance during this work.

I am grateful to all the people who helped me one way or another during those years at CENPARMI. Special thanks go to Stan Swiercz, Michael Assels and William Wong for their technical competence, for maintaining our computer network, and their great help in mastering the Unix operating system. Thanks to Christine Nadal for her help in collecting the database of real life cheque images from Bell Quebec.

I am thankful to Dr. Liu Ke for the helpful exchange of ideas while working together.

I am very thankful to Nick Strathy for many helpful and fruitful discussions. His help in all the aspects of conducting research, including structuring and managing to have a useful C library that has a significant importance of supporting this research.

I'd like to thank my father Nassif, my mother Salma, and my beautiful sister Sandra for always believing in me and encouraging me to achieve high goals. A very special thanks to my brother Fady and his adorable wife Rita for many great discussions and for providing me a place in their warm great heart and in their great soul and mind.

Dedication

This thesis is dedicated

to

My Great Father

Nassif

My Faithful Mother

Salma

My Truthful Brother

Fady

His Adorable Wife

Rita

and

My Lovely Sister

Sandra

Contents

List of Figures

Chapter 1

Introduction

Description of the problem, its significance, the novelty of the approach, research objectives, and thesis organization.

1.1 The Problem

Document image analysis is concerned with the automatic interpretation of images of printed and handwritten documents, including texts, engineering drawings, maps, music scores, business forms, [23, 24, 26, 28, 29, 40, 42, 181, 182, 124, 158, 159] etc. Research in this field descends in an unbroken tradition from the earliest experiments in computer vision, and remains distinguished by close and productive ties between the academic and industrial communities. While the difficulty of its characteristic problems continues to stimulate basic research, general agreement on quantifiable performance standards has encouraged the evolution of sound engineering methods. As a result, research in this area supports a rapidly growing industry.

Document images offer computer vision researchers unique opportunities. The early stages of processing such as feature extraction are relatively tractable, allowing rapid access to extraordinarily challenging later stages requiring the construction of full interpretation of complex scenes. As a result, a major focus of research is the architecture of complete, integrated vision systems, often exhibiting extremely high competence.

1.2 Motivation

Document analysis aims at the translation of data presented on different document images that is addressed to human comprehension into a computer-revisable form. In other words, the pixel representation of a digital document image must be converted into a formal and structured set of symbolic entities describing valuable features, which are appropriate for the intended type of computerized automatic information processing. It is clear that the achieved symbolic description level resembles the degree of understanding acquired by a document analysis system. This interpretation of the term "understanding" will be clarified in depth in this thesis. An attempt shall be made to clarify the important question: "Up to what level can a machine really understand a given document?" In fact, some of the different symbolic descriptions achieved can be used to produce an automatic processing system that can understand different types of document images. So, it is important at this level to ask some other interesting research questions: Could a machine understand which language is written or typed on these documents? Could a machine understand and differentiate between different layouts of various documents? Could a machine understand the relationship between one document and another one or the relationship between one item and another of the same document? Could a machine automatically process a given document without previous knowledge about its signature or it is necessary that the machine should have previous knowledge about the signature of a particular document in order to process it and understand it? Could a machine realize the difference between the relevant and the irrelevant information on a document image? Could a machine clearly extract and preserve the quality of any targeted information on documents? Could a machine have knowledge about the different topics being addressed in a given document? Of course, these and other challenging questions have created a very active field in document image analysis and recognition.

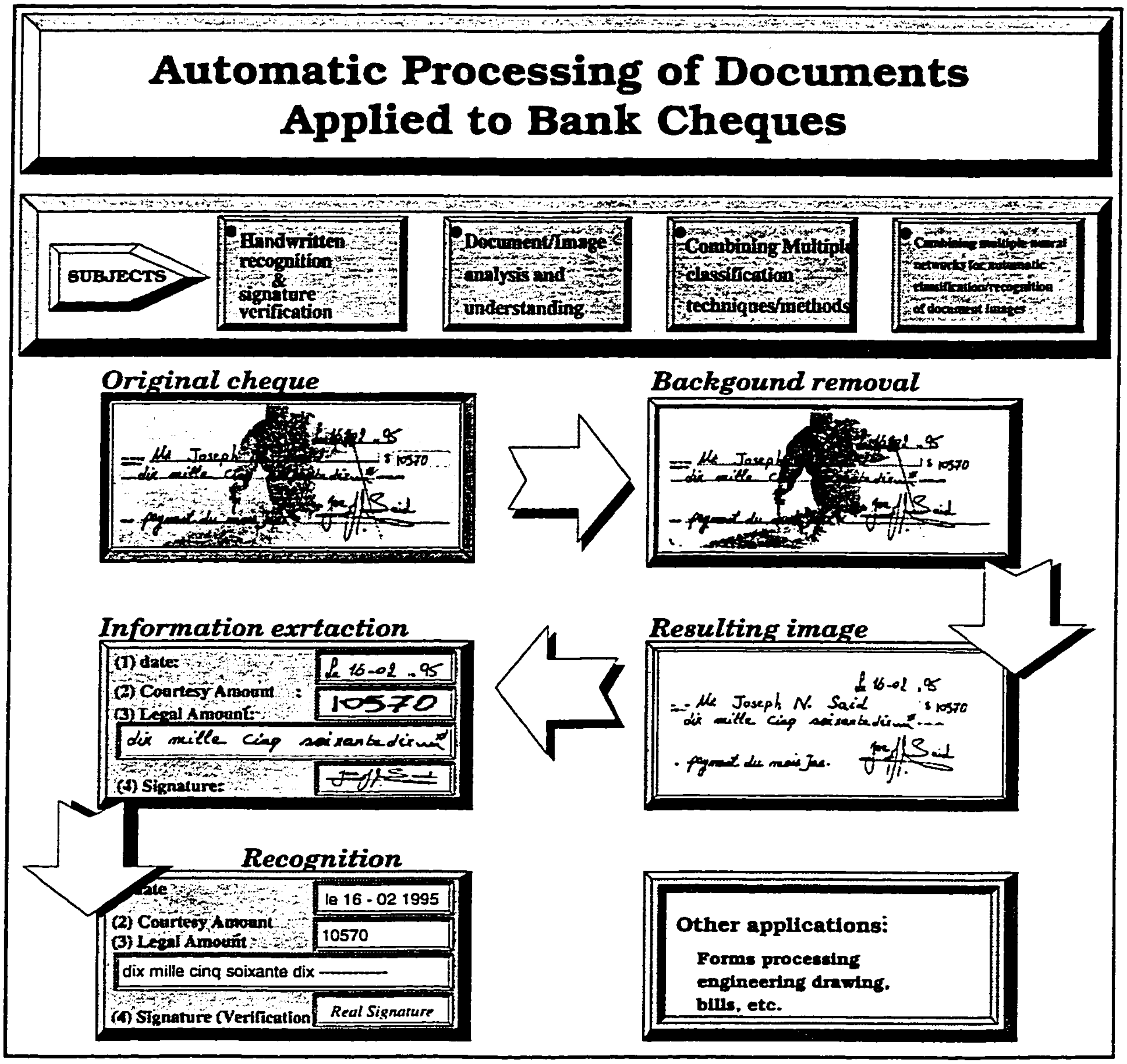

Figure 1.1: Automatic Processing of cheque images.

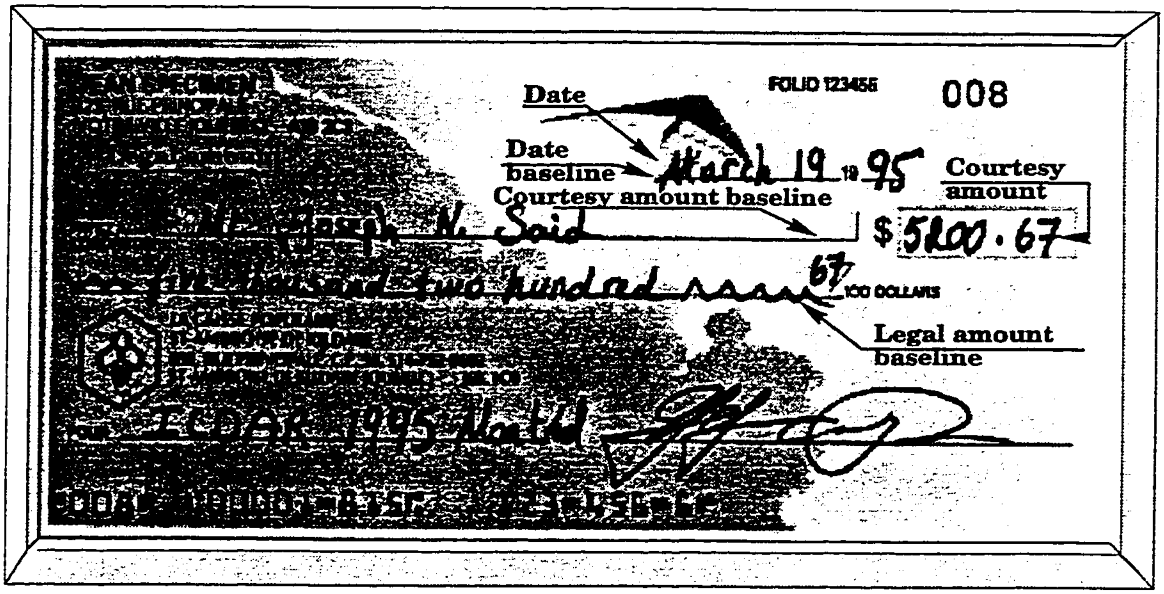

Figure 1.2: Research objectives: locating and extracting different data fields in bank cheques such as the date, the legal amount, and the courtesy amount.

1.3 Research Objectives

Enabling computers to automatically process specific information contained in bank cheques [182] is the major objective of this research. This specific information will be presented to specialized recognition systems for further processing and recognition [180]. For this reason the objectives in this research are to:

1. Analyze, identify, and understand different types of documents such as bank cheques.

2. Locate and extract different data fields in bank cheques such as the courtesy (or numeric) amount, the legal amount (generally written in words), and the date of the cheque. Figure 1.2 illustrates.

3. Suppress visual noise and enhance the efficiency and quality of the extracted data [28, 181] that will be presented later to specialized recognition systems such as [180] or those used in [124] and [124]. Figure 1.3 illustrates.

1.4 Novelty of the New Approach

In a step to develop a reliable and an efficient system to automatically identify, extract, and help to recognize information written or printed on document images, a top-down formal system is developed that processes documents starting from the document class and ends with the pixel representation of the different items that should be extracted. This approach will view a document as a hierarchy of related items that are (a) the background which contains simple or complex scenes that should be eliminated and (b) the foreground which contains baselines that must be removed and handwritten data, such as the date, the legal amount, and the courtesy amount, that should be extracted with minimum distortion.

The novelty of this new approach is to eliminate the background, first, by introducing a new recursive dynamic thresholding technique that could be used globally or locally on a given cheque image. As a second step, baselines that could intersect the handwritten data are recognized and removed with the challenge of minimizing the distortion of the extracted items. Two methods are proposed to tackle this difficulty. The first method acts as a detector of the handwritten data that intersects with the baselines that should be eliminated and uses morphological and topological processing to identify and fill the gaps that resulted after eliminating the detected baselines. The second method proposed a new dynamic morphological processing technique which acts as a detector and a preserver of the handwritten data that intersects with the baselines that should be eliminated. The second method highly advances the efficiency of item extraction by more than 80% and enhances the quality of the extracted items when combined with local processing techniques.

Figure 1.1 illustrates a sample cheque image process as viewed by the top-down approach.

1.5 Thesis Organization

In a step to answer the questions as presented in Section 1.2, it is important to introduce the field of document processing and consider various approaches, as will be

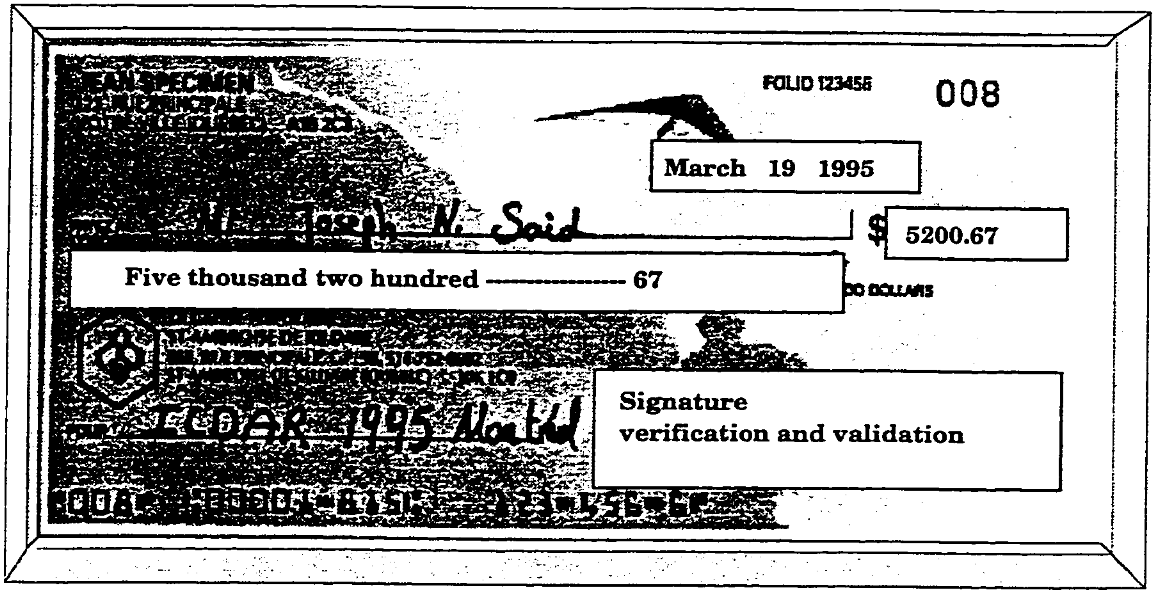

Figure 1.3: Recognition of extracted information.

presented in Chapter 2. As a second step, we will go many steps beyond presenting some of these approaches that aimed at understanding and extracting information from various types of document images to present a new formal, complete, and efficient system that automatically processes and extracts different types of information from document images, such as bank cheques. The new system will be compared with other systems to study its performance analysis. Figure 1.4 illustrates the thesis organization.

Chapter 2 will introduce the field of document processing, its major components, its different approaches, its difficulties and motivations, and its future trends. Moreover, similar works that tackled the same problem will be presented. These works could be classified under two main approaches: a top down approach or a bottom up approach. Moreover, Chapter 2 will motivate the need for a new approach.

Chapter 3 will present a new and an extended approach for image segmentation to eliminate the background of a cheque image. The new approach extended Otsu's approach [161] from dynamically extracting a single object to dynamically and recursively thresholding multiple objects from a cheque image.

Chapter 4 will present the basic top-down formal system for processing cheque

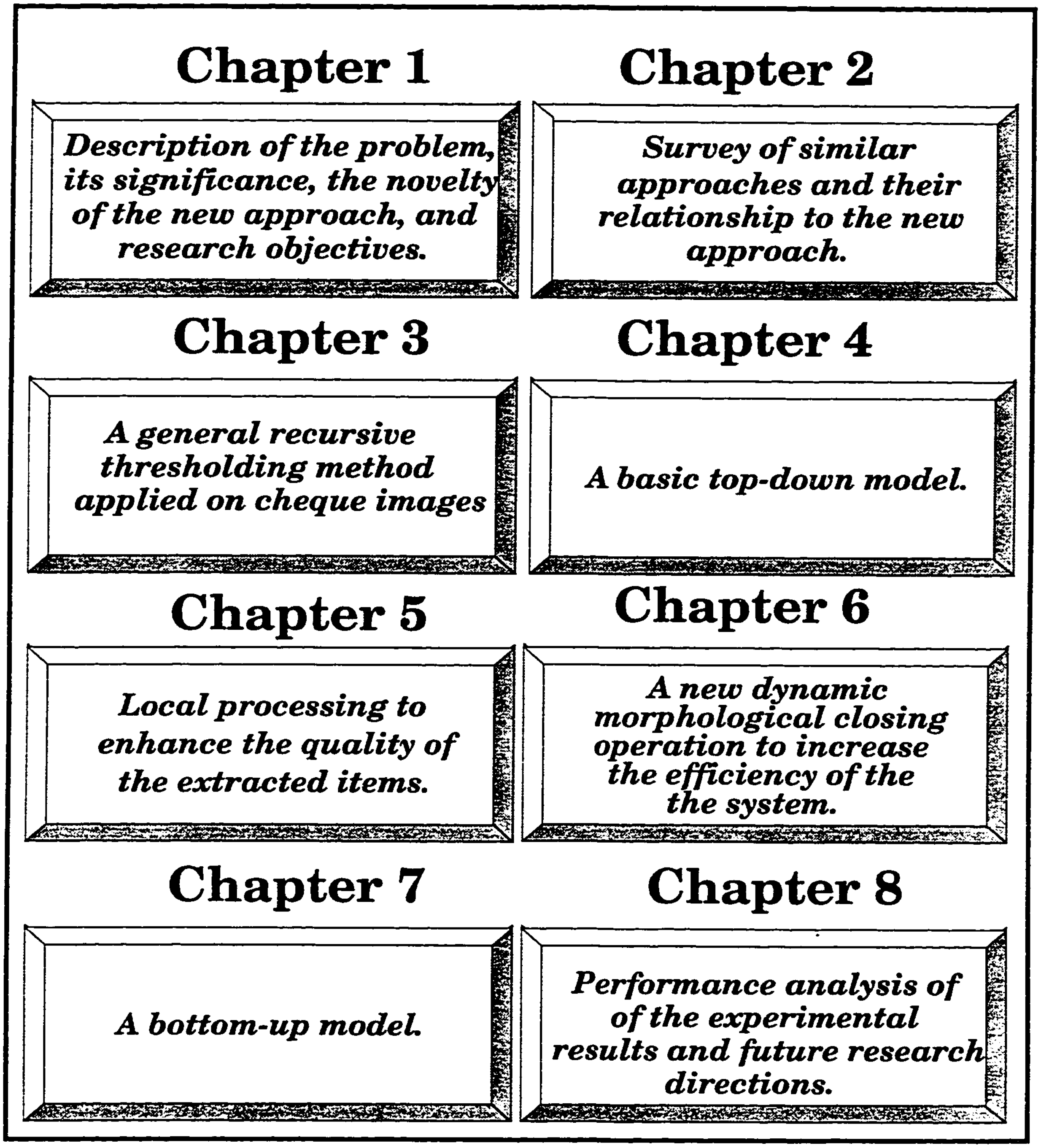

Figure 1.4: Organization of the Thesis.

images. Moreover, the architectural design of this approach will be introduced highlighting the major and most important aspects of gray-scale image processing. In this approach, the morphological processing of gray-scale images will be presented to point out the important usage of morphological processing in our model. Moreover, the concept of topological processing will be presented as a solution to the problem left out by morphological processing.

Chapter 5 will present a local processing approach that extended the approach presented in Chapter 4 to increase the quality of the extracted information from cheque images.

Chapter 6 will present a new and extended approach to morphological closing operation using a dynamic structuring element $\mathcal{V}$ as opposed to a fixed structuring element used in Chapters 4 and 5. The purpose of the new approach is to take into consideration the variability of the slanted handwritten information that intersects with the baseline that should be thresholded from the morphological image. In other words, the new approach does not cause any loss in the handwritten information that intersects with the baselines that should be thresholded from the morphological image. The new morphological closing operation highly improved our algorithm by decreasing the processing time and increasing the productivity of processing the cheque images.

Chapter 7, as a step to increase the throughput of extracting the date, the legal amount, and the courtesy amount, a new bottom-up method is introduced to process documents that have a light foreground (handwriting) against a dark background. This approach viewed document images starting by the pixel value in a step to distinguish between the pixels that belong to the background and the pixels that belong to the foreground. Having recognized the foreground objects, the system recognizes the baselines that must be eliminated and extracts the required items.

Finally, Chapter 8 will present the analysis of the experimental results in order to evaluate the efficiency and reliability of the developed system. A comparative study between the top-down approach and the bottom-up approach showing the strengths and weaknesses of each will be presented. Moreover, for future research directions, a quantitative study on the possibility of combining the two approaches will be presented.

Chapter 2

A Survey of Similar Works and Their Relationship to This Work

Survey of similar approaches and
their relationship to the new approach.

2.1 Document Processing

Nowadays the most common means of presenting and distributing inf
papers. Documents are made up of a series of logically ordered piece
taining information. Sometimes this information, whether writter
be constrained or structured according to a given logical or phys
mat and some other times it can be unconstrained or unstruct
form the primary means to present information in documents
journals, newspapers, magazines, books, letters, and many
ulating, and processing most of these forms require a lot
Adding to this are the privacy and security issues that
pecially in critical working environments where the pr
highly important (government, banks, business firr
recognition is a growing and productive field that is
lems by developing intelligent computer programs that a
analyze, understand, and recognize the contents of these docume
this, document analysis and recognition systems will automatically extra

information from these documents and store them in appropriate database systems for further retrieval and processing. In this context, privacy and security of the document's content is maintained, speed of interpreting and manipulating the document's content is increased, and most importantly, the document could be stored in an electronic medium where it could be shared and accessed by various parties at the same time. Fortunately, the decreasing cost and increasing performance of hardware will eventually make the storage and distribution of documents by electronic means the predominant medium as Wong *et al.* [235] pointed out.

As a branch of pattern recognition and artificial intelligence, document analysis and recognition has grown rapidly during the past few years. Numerous techniques and methods have been explored and some document processing systems have been proposed as in [86, 128, 206]. However, most of the studies are concentrated on non-financial types of documents such as newspapers, magazines, technical reports, bibliographical documents, envelopes, and so on. Only a few are related to the processing of business documents [24, 184, 212, 208, 240] in spite of the fact that millions of business transactions take place every day. Associated with them are documents such as bank cheques, payment slips and bills which have to be processed. A great deal of time, effort and money will be saved if such documents can be entered into the computer and processed automatically. However, not withstanding major advances in computer technology and intensive research efforts, the degree of automation in processing such documents is still very limited and a lot of manual labour is currently spent in this area.

The above situation indicates that any system which can speed up the processing of such documents will make a significant contribution. One of the most significant is [24] which has implemented a method of entering forms that contain machine-printed data produced in business environments. A system called IFP (Intelligent Form Processing) has been constructed. The IFP system uses a setup phase to create a model of each form to be read. Scanned forms containing data are compared against the matching form model. Special algorithms are employed to extract data fields and remove background printing (e.g., lines) which intersects the data. Another way was introduced in [153] where an empty form is first registered with the system before processing. But it takes up a large amount of memory. Areas of desired images in the form are then entered into the computer via man-machine interaction. Subsequent form processing simply involves the matching of the filled form with the empty one to produce the images which have to be processed later.

According to [208], the above systems are rather complicated. For instance, the IFP system [24] is a more general form document processing system. It has a complex architecture which contains form library, image database, coded database, form classification, different image subsystems, etc. This system can treat a great variety of form documents. It could be wasteful if the IFP is used to process only business documents.

In [208], a fixed document processing subsystem and a flexible document processing subsystem are separately used to process different types of financial documents according to their structural complexity. The purpose of the fixed subsystem is to process documents that have common properties in their geometrical and their logical structures, such as bank cheques. The purpose of the flexible subsystem is to process documents that do not necessarily have common properties in their geometrical or their logical structures, such as different types of income tax forms. Both subsystems are robust and efficient in describing the structural properties of a binarized document. However, to improve the quality of the information to be extracted from a given document by preserving its topological properties and to increase the recognition rate of the extracted information from the documents, binarized documents should be replaced by grey-scale documents.

2.2 Major Components and Major Goals

As a graphical illustration, a basic model for document processing is shown in Figure 2.1. The input to this model is a document and the output is the knowledge obtained about the given document using the logical structure. Figure 2.1, shows that there are different principal concepts through which a given document is processed.

Tang *et al.* in [207] and Tsujimoto *et al.* in [219] pointed out that dealing with various kinds of document layout is an important and major property of any document structure analysis system. Two approaches are essential to this capability: document analysis and document understanding.

The following principal concepts for document processing are proposed:

- Document processing is divided into two phases: *document analysis* and *document understanding.*
- A document is considered to have two structures: *geometric (layout) structure* and *logical structure.*
- Extraction of the geometric structure from a given document refers to document analysis. It breaks down a document image into several blocks, which represent coherent components of a document, such as text lines, headlines, graphics, etc, with or without the knowledge regarding the specific format. Mapping the geometrical structure into a logical structure is defined as document understanding. Normally, it maps the layout structure into logical structures considering logical relationship between the objects in specific documents.

The basic model of document processing can be formally described as:

Definition 1 A document Ω is specified by a quintuple

$$\Omega = (\Im, \Phi, \alpha, \beta, \delta) \tag{2.1}$$

such that

$$\begin{bmatrix} \Im \\ \Phi \\ \alpha \\ \beta \\ \delta \end{bmatrix} = \begin{bmatrix} \{\Theta^1, \Theta^2, \ldots, \Theta^m\} \\ \{\varphi_l, \varphi_r\} \\ \{\alpha^1, \alpha^2, \ldots, \alpha^p\} \\ \{\beta^1, \beta^2, \ldots, \beta^q\} \\ \{\Im \times \Phi \longrightarrow 2^{\Im}\} \end{bmatrix} \tag{2.2}$$

and

$$\Theta = \{\Theta_j^i\}^*$$

$$\alpha \subseteq \Im$$

$$\beta \subseteq \Im$$

where

- $\Im$ is a finite set of document objects which are sets of blocks Θ^i (i = 1,2, ..., m).
- $\{\Theta_j^i\}^*$ denotes repeated sub-division, since an object may be subdivided into several smaller objects.
- Φ is a finite set of logical linking functions, which indicate logical linking of the document objects. φ_l, φ_r stand for leading linking and repetition linking respectively.
- α is a finite set of heading objects.
- β is a finite set of ending objects.
- δ is a finite set of logical linking functions which indicate logical linking of the document objects.

Definition 2 Document processing is a process to construct the quintuple represented by Equations 2.1 and 2.2. Document analysis refers to extracting elements $\Im$, Θ^i, and $\{\Theta_j^i\}$, i.e., extracting the geometric structure of Ω. Document understanding deals with finding Φ, α, β, and δ by considering the logical structure of Ω.

An example of a document is illustrated in Figure 2.2.

$$\Im = \{\Theta^1, \Theta^2, \Theta^3, \Theta^4, \Theta^5, \}$$

$$\Theta^4 = \{\Theta_1^4, \Theta_2^4\}, \Theta^5 = \{\Theta_1^5, \Theta_2^5, \Theta_3^5\}$$

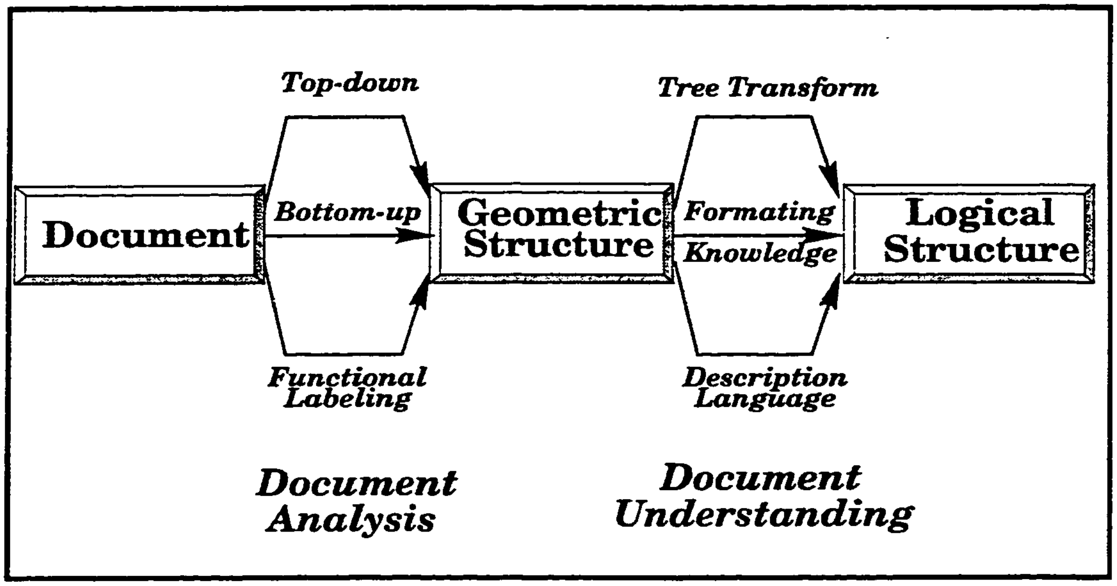

Figure 2.1: Document Processing Model.

$$\alpha = \{\Theta^1, \Theta^2\}, \beta = \{\Theta^4, \Theta^5\}$$

$$\delta(\Theta^1, A) = \Theta^3, \quad \delta(\Theta^2, R) = \Theta^1, \quad \delta(\Theta^3, L) = \Theta^5, \quad \delta(\Theta^4, L) = \Theta^5_i, \quad \delta(\Theta^5, B) = \Theta^2_i$$

where A, B, L, R represent logical linking *above, below, left,* and *right* respectively of the document objects.

2.2.1 Document Structures

The key concept in document processing is that of structures. Document structure is the division and subdivision of the contents of a document into increasingly smaller parts which are called *objects.* An object which can not be subdivided into smaller objects is called a *basic object.* All other objects are called *composite objects.* Structure can be realised as a geometric (layout) structure in terms of its geometric characteristics, or a logical structure due to its semantic properties.

2.2.2 Geometric Structure

Wilcox and Spitz in [233] pointed out that extraction of the geometric (layout) structure from the page image of a document is the minimum requirement for a document analysis system. Ingold in [88], Tsujimoto and Asada in [219] stated that based on this structure and some specific formatting knowledge, the partial logical structure of several document types can often be obtained.

The logical structure of a document can be extracted if more knowledge about the document is known, such as the syntactical knowledge about the relation among logical entities of the document and the semantic knowledge about possible keywords or letters inside each logical entity. Further, it is required to recognize the logical structure of different documents in order to separate the logical structure of a document into levels to gain a finer level of understanding. Section 2.2.3 will elaborate more on the logical structure of a document image.

Geometric structure represents the objects of a document based on the presentation, and connection among these objects. Figure 2.2 illustrates an example of the geometrical structure of a given document image. According to the International Standard ISO 8613-1:1989(E) [89], the geometric or layout structure can be defined as follows:

Definition 3 Geometric Structure is the result of dividing and subdividing the contents of a document into increasingly smaller parts, on the basis of the presentation.

Geometric (layout) Object is an element of the specific geometric structure of a document. The following types of geometric objects are defined:

- *Block* is a basic geometric object corresponding to a rectangular area on the presentation medium containing a portion of the document content.
- *Frame* is a composite geometric object corresponding to a rectangular area on the presentation medium either one or more blocks or other frames.
- *Page* is a basic or composite geometric object corresponding to a rectangular area. If it is a composite object, containing either one or more frame(s) or one more block(s).
- *Page set* is a set of one or more pages.

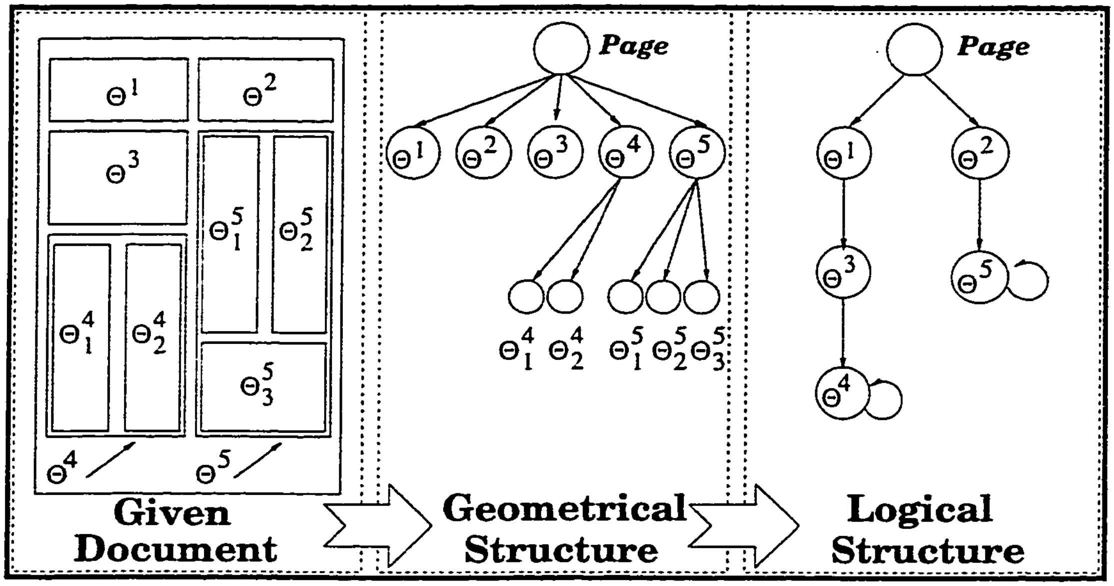

Figure 2.2: A Simple Example of Document Processing.

- *Document Geometric (Layout) Root* is the object at the highest level in the hierarchy of the specific geometric structure. The root node in the above example represents a page.

The geometric structure of a document can be formally defined as a *document geometric model (DGM)* according to the basic model as defined in Definition 1. To facilitate document analysis, an entropy function will be used in the following definition.

Definition 4 A DGM is described by a document space $\Omega = (\Im, \beta_U, H_\Im)$, where $\Im$ is the set of geometric objects which are either basic objects or composite objects; β_U is a set of operations which are performed in $\Im$, and $H_\Im$ stands for entropy function, such that:

$$\Im = \{\Im_T, \Im_G\} \tag{2.3}$$

$$\beta_U = \{\cup, \cap\}$$

$$\forall i \neq j (\Im_i \cup \Im_j) \subseteq \Omega$$

$$\forall i \neq j(\Im_i \cap \Im_j) = \phi$$

$$H_{Im} = -\sum P(\Im_T, \Im_G) log P(\Im_T, \Im_G)$$

where

$\Im_T$ represents *Text Area*, and $\Im_G$ stands for *Graphic Area*. Each area can be defined as follows:

$$\Im_T = \{\Theta^1, \Theta^2, ..., \Theta^m\}$$

$$\Im_G = \{\Theta^G, \Theta^C, \Theta^R\}$$

where

- $\Theta^1, \Theta^2, \ldots, \Theta^m$ show different *text Blocks*, e.g. Θ^1 shows a *Headline block*, Θ^2 shows a *Text line block*.
- Θ^G indicates *Geometric Graphics Blocks* such as points, area, and lines.
- Θ^C presents *Graphics Characters Blocks* such as accented letters and special symbols.
- Θ^R presents *Raster Graphics (Picture) Blocks*.

and

$$\Theta^j = \{\Theta_i^j\}^*, \quad \Theta_i^j = \begin{cases} \{\sigma_1^C, \sigma_2^C, \ldots, \sigma_r^C\} & \Theta_i^j \subseteq \Im_T \\ \{\sigma_1^G, \sigma_2^G, \ldots, \sigma_s^G\} & \Theta_i^j \subseteq \Im_G \end{cases} \tag{2.4}$$

where

- σ_i^C indicates a *Character*.
- σ_i^G represents a *Graphical Element* including geometric graphics, graphic character elements and raster graphics elements.

2.2.3 Logical Structure

Document understanding emphasizes the finding of logical relations between the objects of a document. In other words, the logical structure represents the objects of a document based on the human perceptible meaning, and connection among these objects. Figure 2.2 illustrates an example of the logical structure of a given document image. According to the International Standard ISO 8613-11989(E), the logical structure can be defined as follows:

Definition 5 Logical structure is the result of dividing and subdividing the content of a document into increasingly smaller parts, on the basis of *human-perceptible meaning* of the content, for example, into chapters, sections, subsections, and paragraphs.

Logical Object is an element of the specific logical structure of a document. For logical object, no classification other than *Basic logical object, Composite logical object* and *Document logical root* is defined. Logical object categories such as *Chapter, Section* and *Paragraph* are application-dependent and can be defined using the *Object class* mechanism [89].

According to the basic model of Equation 1, a formal description of the logical structure termed *document logical model (DLM)* is presented as follows:

Definition 6 A DLM is described by a tree, $\mho = (\Theta, \Re)$. Θ denotes a nonempty set of nodes which represent the logical objects. $\Re$ expresses a set of edges representing relations between the logical objects.

$$\Theta = \{\Theta^1, \Theta^2, \ldots, \Theta^n\}$$

$$\Re = \{\Im \times \Im\}$$

$$\Re = \{\Re_1, \Re_2, \ldots, \Re_m\}$$

$\mho$ can be represented as an *Incidence Matrix* with size $m \times n$:

$$\mho = \begin{cases} 1 & \text{when } \Re_j \text{ is incident with } \Theta^i \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

2.2.4 Document Analysis

Document Structure Analysis is often defined as a process, which automatically transforms the document page image into its structural representations for computer processing. In other words, document analysis is defined as the extraction of the geometric structure of a document. In this way, a document image is broken down into several blocks, which represent coherent components of a document, such as text lines, headlines, graphics, etc, with or without the knowledge regarding the specific format [207, 219]. Unfortunately, in a document preparation system a document may exist in different forms, Jolobof [96] Quint [174], and we are interested only in durable representations: i.e. those which can be stored on devices or transmitted across communication networks.

To start with, two representations are of practical interest: The layout structure and the logical structure of a document. The first describes the formatted form of the document that is to be sent to output devices; it is normally called layout structure. The second is manipulated by a system's user in specifying the structure and contents of the document: it is normally called logical structure.

Brown in [18] stated that maintaining the underlying logical structure of a document can have many advantages for document processing systems. Ideally the structure reflects the way the author/editor thinks about the document, and allows the processing system to provide intelligent help with section numbering, cross-references and the provision of such useful adjuncts as a table of contents and an index. It will also provide a suitable framework for intelligent finding of text, graphics and other types of content.

Top-down, Bottom-up, and Functional labelling have been used in document analysis. Each has its advantages and disadvantages. The top-down approach is fast and very effective for processing documents that have a specific format. On the other hand, the bottom-up approach is time consuming. But it is possible to develop algorithms which are applicable to a variety of documents. A better result may be achieved by combining the two approaches [147].

2.2.5 Document Understanding

As document analysis extracts geometric structures from a document image by using the knowledge about the general document and/or the specific document format, document understanding maps the geometric structures into logical structures considering the logical relationship between the objects in specific documents. There are several kinds of mapping methods in document understanding: [219] proposed a tree transformation method for understanding multi-article documents. Toyoda *et al.* in [214] discussed the extraction of Japanese newspaper articles using a domain specific knowledge. Inagaki *et al.* [87] constructed a special purpose machine for understanding Japanese documents. Higashino *et al.* [80] proposed a flexible format understanding method, using a form definition language. Tang *et al.* and YAN [211, 209, 240] have led to the development of a form description language for understanding financial documents. These mapping methods are based on specific rules applied to different documents with different formats. A series of document formatting rules are explicitly or implicitly used in all these understanding techniques. In this section, document understanding based on tree transformation, document formatting knowledge and document description language will be discussed.

Document Understanding Based on Tree Transformation

This method defines document understanding as the transformation of a geometric structure tree into a logical structure tree [219].

A document has an obvious hierarchical geometric structure, represented by a tree. Moreover, the logical structure of a document is also represented by a tree. In this context, three kinds of blocks are defined: H (head), B (body) and S (either body or head). During the transformation, a label is attached to each node. Labels include title, abstract, sub-title, paragraph, header, footnote, page number, and caption. The transformation, which moves the nodes in the tree, is based on four transformation rules. These rules are created according to a layout designed according to the manner in which humans read. Two rules (Rule 1 and Rule 2) are based on the observation that a title should have a single set of paragraphs as a child in the logical structure. The paragraph body in another node is moved to the node under the body title by these rules. Another rule (Rule 3) is mainly for the extraction of characters or sections headed by a sub-title. In another rule (Rule 4), a unique class is attached to each node.

Document Understanding Based on Formating Knowledge

Since a logical structure can correspond to a variety of geometric structures. the generation of logical structure from the geometric structure is difficult. One of the promising solutions for this problem is use of formatting knowledge. The formatting rules may differ from each other because of the type of document and language to be used in it. However, for a specific kind of documents, once the formatting knowledge is acquired, its logical structure can be deduced. An example can be found in [210] where a method of extracting articles from Japanese newspapers has been proposed. In this method, six formatting rules of Japanese newspaper layout are summarized. An algorithm for extracting articles from Japanese newspaper bas been designed based on the formatting knowledge.

Another example can be found in [43] where a business letter processing approach has been developed. Because business letters are normally established in a single-column representation, letter understanding is mainly the identification of the logical objects, like sender, receiver, date, etc. In this approach. the logical objects of the letter are identified according to a statistical Database (SDB). As the author reported. the SDB consists of about 71 rule packages derived from the statistical evaluation of a few hundred business letters. Other knowledge, like the shape. size and pixel density etc. of the image block can also be used for document understanding. References [48, 241] use statistical features of connected components to identify the address blocks on envelopes.

Document Understanding Based on Description Language

One of the most effective ways to describe the structures of a document is the use of a description language. [80] detects the logical structure of a document and makes use of the knowledge rules represented by a form definition language (FDL). The basic concept of the form definition language is that both the geometric and logical structures of a document can be described in terms of a set of rectangular regions. For example, a part of a program in form definition language coded for the United Nations' (UN) documents is listed below:

```
(defform UN-DOC#
        (width 210) (height 297)
        (if (box (? ? ? ?)
                (mode IN Y LESS)
                (area (0 210 60 100))
                (include (160 210 1 5)))
            (form UN-DOC-A
                (0 210 0 297))
            (form UN-DOC-B
                (0 210 0 297))))
(defform UN-DOC-A ...)
(defform UN-DOC-B ...)
```

It means that the UN documents have a width of 210mm and a height of 297mm. The "if" predicate is one of the control structures. If the box predicate succeeds, the document named UN-DOC# is compared with UN-DOC-A and UN-DOC-B, and analyzed as UN-DOC-A. Otherwise, it is analyzed as UN-DOC-B. The box states that a rule line should exist inside the region (0 210 60 100) and satisfy the conditions that the width of the ruled line is between 160 mm and 210 mm and the height is between 1 mm and 5 mm (defform UN-DOC-A ...) and (defform UNDOC-B ...) will give the definition of the UN documents with and without a ruled line with these properties stated above.

According to the definition, a form dividing engine will analyze the document and produce the images of some logical objects, such as the organization which issued the document, document number, and section, etc. More details about this method can be found in [80].

As stated in the previous paragraph, the concept and purpose of document analysis and recognition systems are helpful when they exist, however, putting these concepts into computers is a very difficult and challenging task to accomplish. In fact, to integrate existing documents, an input system is required in many fields, such as desktop publishing, machine translation, database management, electronic communication, and so on. Such a system could assist a user in the encoding of printed documents for computer processing.

Among such systems, a document structure analysis system must be developed to extract structural information from printed documents to create a database that stores the extracted structure of each document.

2.3 Document Processing: Two Main Approaches

In general, there are two main types of analysis that are applied to text in documents. The first aims at recognizing characters and words from their bit-mapped images. The second aims at analyzing the page layout (structure of the text) to discover formatting of the text and, from that, to derive meaning associated with the positional and functional blocks in which the text is located. The first type of analysis is usually distinguished as being applicable to either machine-printed or handwritten character recognition [180]. The second type of analysis is applied to formatted and machine-printed pages and a special type of layout analysis (forms recognition), where machine-printed or handwritten text occurs within structured and predefined blocks on a printed form.

There are two types of structural analysis, viz top-down or bottom up. In top-down analysis, a page is segmented from large components to smaller ones. In this approach, the page is split into various column blocks of text. Moreover, each column is split into paragraph blocks, each paragraph is split into text lines, each text line is split into words, and words are split into characters. On the contrary, in bottom-up analysis connected components are merged into characters, then words, then text lines, then paragraphs, then columns.

2.3.1 Top-down Analysis

Top-down (knowledge-based) approach proceeds with expectation of the nature of the document. It divides the document into major regions which are further divided into sub-regions, etc. [62, 80, 88, 107, 108, 153, 156]. The top-down approach is fast and very effective for processing documents that have a specific format.

For layout analysis, horizontal and vertical projection profiles can be used. The vertical projection profile provides information that helps in detecting and locating text columns. In a similar manner, the horizontal projection profile provides information that helps in locating the paragraph breaks within each column[1]. Moreover, the horizontal projection profile within paragraphs provides information to locate text lines. In addition to this, some differentiation among detected text blocks can be made using this approach. In fact, the horizontal projection profile of a title indicates characters in a larger size text than that of the body of a text. A footnote may have smaller text line spacing than that of the body of text. In this top-down approach, much information can be determined relating to the layout of the page.

In practice, when the projection profile approach to layout analysis is used, it is not necessary to perform processing on all pixels of the original resolution image. Instead, to improve efficiency, the image may be reduced in size. One approach in tackling this problem is to smooth characters into smaller and unrecognizable (as characters) blobs. These blobs can be of a single character, a word, or a text line based on the smoothing used. As opposed to the previous projection profiles that used the whole image, now, projection profiles are constructed on the basis of fewer pixels. Of course, this reduces the processing time and improves the efficiency of layout analysis. An

[1] When the paragraphs are separated by some extra blank space that is greater than the inter-line spacing.

example of smoothing is the run-length smoothing algorithm which is a very good and popular method for performing this smoothing process. By "smearing" the text to join characters into blobs, the runlength smoothing algorithm merges characters into words, words into text lines, and (sometimes) text lines into paragraphs. In short, the runlength smoothing algorithm detects white spaces between black pixels on the same lines and sets them to black if the length is less than a given threshold. Subsequent merging between characters of the same word and words of the same line is accomplished with appropriately selected thresholds. Besides reducing the number of pixels, this method can lead to better results, especially for sparse text that otherwise yields poor or noisy projection profiles. This use of horizontal and vertical projection profiles requires that the image be first skew corrected and that spacing is known and uniform within the image.

Projection profiles could be used in a more structured top-down method that successively splits the document into smaller rectangular blocks by alternatively performing horizontal and vertical "cuts" along white space, starting with a full page and continuing with each sub-block. This approach is successfully used by Nagy [146, 148]. From the horizontal and vertical projection profiles of each block, the locations of these cuts are recognized. Later, segmentation is aided by functional labelling being performed at the same time. Such a functional labelling is based on apriori knowledge of features of the expected blocks and on the characteristics of their profiles described in a document syntax. For instance, a title block with a large-font appears in the horizontal projection profile as having wider peaks. On the contrary, the text body will appear in the horizontal profile as narrower, periodic peaks and valleys. Segmentation results are represented in an X-Y tree, where the top-level node represents the page, each lower node represents a block, and each level alternatively represents the results of horizontal (X-cut) and vertical (Y-cut) segmentation. In such an approach, segmentation can be performed down to individual paragraphs, text lines, words, and characters. One good advantage here is that the combination of functional labelling in structural processing enables the process to be directed and corrected. However, this requires the initial specification of blocks syntax, which depends upon knowledge of block features. It should be noted that it is critical that the image has no background, no skew, and that any noise has already been removed first. These two conditions are required by most segmentation methods.

Baird in [10] used another top-down layout technique that analyzes white space (that is, background area versus foreground text) to isolate blocks and then uses projection profiles to find lines. In this approach, first, all locally maximal white rectangular structures among the black connected components are enumerated from largest to smaller ones. The largest of these white rectangular structures forms a "covering" of the white background on the page, thus forming a partition of the foreground into structural blocks of text. A good point of this approach is that different sizes of rectangles may be chosen where segmentation is performed down to a chosen level; for example, down to columns, paragraphs, or even individual characters. However, the page must have a Manhattan layout[2]. As pointed out by Baird in [10], one advantage of using background white space versus foreground text for layout analysis is the language independence that results, because white spaces are used as a layout delimiter in similar ways in many languages. Another advantage is that few parameters need be specified. However, a drawback is that the choice of what constitutes a "maximal" rectangle, for example, the longest, or the maximal area, -may be nonintuitive as well as different for differently formatted documents.

A primary advantage of top-down methods is that they use global page structure to their benefit to perform layout analysis quickly. For most page formats, this is a very effective approach. However, these methods may be inappropriate for other pages where text does not have linear bounds and where figures are intermixed both in and around text.

Consider for example, a magazine which crops text around an inset figure so that the text follows a curve of an object in the figure rather than a straight line. A more complicated case is when tables of contents in magazines and journals are often formatted with inset figures, centred column entries (rather than justified), and other

[2]that is, it must have only one skew angle and must be separable into blocks by horizontal and vertical cuts.

non-Manhattan layout. For these formats, the bottom-up techniques that will be described below are more appropriate with the trade-off that they are usually more expensive to compute.

2.3.2 Bottom-up Analysis

Bottom-up (data-driven) approach progressively refines the data by layered grouping operation. the bottom-up approach is time consuming. But it is possible to develop algorithms which can be applied to a variety of documents [4, 32, 46, 54, 87, 75, 90, 235].

As mentioned above, bottom-up layout analysis locates and identifies small components and groups them into successively larger components until all blocks are found on the page. Unfortunately, there is no single, general method that typifies all bottom-up techniques. In this report, we will describe a number of approaches that can all be classified as bottom-up but use very different intermediate methods to achieve the same purpose.

One approach combines a number of the techniques as described by Fisher in [55]. First, using Hough transform a skew is found and appropriate skewing is performed. Then, the one-dimensional Fourier transform of the projection profile for θ fixed at the computed skew angle determines the between-line spacing. Runlength smoothing is performed, and then within-line spacing is determined by finding the peak on a histogram of these within-line lengths of white spaces (that is, inter-character and inter-word spacing) and of black lengths (that is, words). Next, by a sequence of run-length smoothing operation in the direction of the skew for words and text lines perpendicular to the skew for paragraphs and text columns, bottom-up merging of the text components is done. The results are on regions, upon which connected-components analysis is performed. These connected-components provide many statistics: for example, ranges of words height, area, and length. This feature information helps to discern text blocks and to discriminate text and non-text. Esposito *et al.* [50] used a similar approach, but they first determined bounding boxes of individual characters and then operated with respect to these bounding boxes, instead of operating on individual pixels, to reduce the amount of computation.

O'Gorman in [158] employed bottom-up $k-$ nearest-neighbour clustering to group from characters into text lines and structural blocks. First, for each document component, k nearest-neighbour connections to neighbouring components are found (where k is usually taken to be 4 or 5). The distance and angles of these connections are compiled in histograms. Since most connections will be made between characters on the same text lines, the peak distance will indicate the inter-character spacing and the peak angle will indicate the skew. Based on these estimates, text lines are found as groups of characters and words along the page orientation. Now, using the document characteristic that text lines of the same block are usually spaced more closely than text lines of different blocks, text lines are grouped into blocks.

Nagy in [146] used a combination of both top-down and bottom-up approaches. Akiyama in [4] used field separators (lines between text regions) and then blank delimiters to segment blocks. Then, using generic properties of documents, global and local text features are determined, and blocks are found and classified. The text features determined here are measures on horizontal and vertical projection profiles, horizontal and vertical crossing counts (the number of black-white and white-black transitions along a line), and bounding boxes for each character. For example, headline text is identified by the property that the larger characters have smaller crossing counts. Bounding box size for individual characters relative to that of other characters in the same region are used to identify these characters. Such a combination of structural layout segmentation and functional labelling technique has both advantages and disadvantages. Obviously, this combination requires more from the user (earlier in the processing stages) to supply information on the expected features of labelled fields. In fact, such approaches will take the trend that proceeding as far as possible in early processing without application-specific knowledge being needed is advantageous because this practice encourages robust techniques. However, other approaches will take the trend that the presence of application-specific knowledge is useful, and should be used to facilitate solving the problem. In the following subsection, functional labelling is discussed independently (as a following process) of structural analysis.

2.3.3 Functional Labelling

As discussed above, some of the layout analysis methods [145, 4] perform functional labelling in the course of structural blocking. Some methods perform blocking only while others perform these steps sequentially, first obtaining structural blocks and then applying functional labels. In either case, functional labelling is performed based on document-specific rules and using many features derived from the document. These features may include the font size styles of the text, general typesetting rules of spacing, the relative and absolute positions of a block on a page, and the relative and absolute position of a field within a block. The formatting rules used to produce the page are those that are used in the reverse process of functional labelling. One problem remains which is the wide variations of formatting style across different documents. However, despite this problem, most humans can easily locate blocks of interest using their acquired knowledge of formatting as well as by using text recognition.

Most of the work in functional labelling has been restricted to particular domains because of the large variations of formats and because of common image-processing problems such as noise, erroneous feature detection, and imperfect text recognition results. In the domain of postal recognition, for example, much work has been done on the problem of locating addresses and address fields Palumbo [163, 223]. For US mail, the objective usually is to locate the address block, then the ZIP code on the last line, and then other parts of the address block from there. This is a good example of a familiar format in which blocks in different locations and fields in different lines can note different functional information. Work has been done in the domain of forms recognition, where the problem of structural blocking is facilitated by the lines of the form delimiting blocks [24, 212]. Functional labelling must be performed to recognize, for example, that a block at the bottom of a column of blocks is a "total" field and that indented blocks are intermediate values. In Dengel [40], functional labelling is used in the domain of business letters. The Office Document Architecture (ODA) standard for document formatting is solely used to hierarchically represent the document components. In Amano [6], the domain of interest is Japanese journals and patents, where labelling is done in part using rules of relative horizontal and vertical placement as well as ranges on the numbers of lines per block. Even within each of these restricted domains, variations of formatting are large, thus presenting many challenging problems to be solved in this field.

Since the processes of functional and structural layout analysis are the reverse of the document-formating process, it is logical to ask if there is some standard for producing and describing document contents that can be used to perform layout analysis and to test its results. Two of the internationally recognized standards are ODA [82] and Standard Generalized Markup Language (SGML) Goldfarb [66]. Although both describe documents, they are very different in their description. ODA has an object-oriented document architecture that describes both structural and functional components of a complete document. These ODA descriptions are hierarchical, so a layout may be defined as a document containing pages on which there are columns, paragraphs, and so on. The functional hierarchy describes a document, which may contain sections or chapters, and each chapter has headings, sub-headings, and so on. The original purpose of ODA was to create a common standard for the interchange of office documents, SGML is a language for "marking up" (or annotating) text with information that describes it as functional components. It was created for publishing documents. In the publishing process, these functional components are used together with publication-specific rules to format the document in a chosen style. In contrast to ODA, SGML can be thought of more as a bottom-up description of a document, and this definition is in keeping with its purpose. Both of these standards have grown beyond their original purposes; for example, both are used for document layout analysis. However, while both are used for publishing and in document analysis, they are by no means used exclusively. Many publications use appropriate systems that are compatible with neither ODA nor SGML, and most current document recognition research today still does not produce results that conform to any standard.

The use of page layout analysis is currently very limited in commercial document systems. The only capability provided by most general systems is specifying absolute block coordinates on which to scan or perform OCR; that is most general systems do not do automatic page layout analysis. Two exceptions to this limitations are systems for special-purpose applications: postal reading.

2.4 Document and Form Processing Systems

A document processing system embodies many of the image and document processing techniques that will be described later in this thesis. To reduce noise and to modify the pixel data to produce a form (such as binary or thinned image) that best facilitates subsequent analysis, pixel-level processing is performed first. Moreover, to describe more succinctly the pixel region contained in the image, pertinent features are found. These features are used to recognize textual format and content. During this process and for recognition of tables and forms, graphics recognition techniques may also be employed, to obtain information from the delimiting lines. The previous process will become interesting and challenging when multiple pages are introduced. In fact, additional aspects of complexity with complete document representations for handling the production of multiple-page document introduced additional and higher complicated aspects of complexity for document representations. Up till now this is a problem that has usually not been handled by document analysis systems and requires more investigation.

Image based electronic libraries is one application of document processing techniques. Three recent systems for storage and display of technical journals include Nagy's [146], the Document Recognition System Amano [6], and The RightPages System [196]. Some of the technical work associated with these systems includes noise reduction, structural and functional page layout analysis, and OCR.

Document processing of postal mail is another application area that is probably the most economically important applications of current document systems implemented. The aims of such systems are to locate address blocks and to read at least some portion of the addresses for automatic routing of each piece. One very critical and important aspect of this problem is that recognition and routing must be performed very quickly to compete with human operations. Some of the work in this area includes that of Palumbo [163] on real-time address block location, Matan [135] on a neural-network system for reading ZIP codes, and Kimura [119] on segmentation and recognition of ZIP codes.

Opposed to the above mentioned document systems that process postal mail, the applications of form recognition entails document images whose formats are more restricted than those of mail or journal publications, because the required fields are in specified locations that are bounded by lines or highlighting. With the knowledge of location of blocks of interest, these blocks can be found and their contents recognized. Although this may seem simpler than non delineated-block location, there are problems inherent in forms reading that are distinct from those of other applications. Two of these problems are locating boxes, usually by graphics recognition techniques upon the lines, and dealing with text that is misaligned with or intersecting the boxes. Also, since forms are often filled out by hand, handwriting recognition may be involved. Some work in forms recognition includes that of [24] and [212]. These systems have been demonstrated on US tax forms, among others.

2.5 The State of the Art of Available Systems

During the last decade, many prototype systems or commercial tools have tackled the goal of document structure analysis. Few systems have been devoted to document understanding although a lot of work has been done on document analysis, see e.g. Wong *et al.* [235], Tsuji [217], Ciardiello [32], Wilcox and Spitz [233]; only Higashino *et al.* [80], Tsujimoto and Asada [219], Dengel [41], Chenevoy and Belaid [27], Nagy *et al.* [146], and Belaid *et al.* [13]. Most of these systems, such as, Wong *et al.* [235], Tsuji [217], Ciardiello [32], Wilcox and Spitz [233], are connected only with OCR and layout structure recognition.

Several more advanced systems execute the bottom-up approach without the aid of OCR; or the result of OCR is not used for the top-down approach.

Nagy *et al.* [146] has introduced a rather complete document image analysis system for technical journals. Based on a page grammar, a syntactic parsing is conducted to assign logical labels to every block on a page, and OCR is used for these; but the obtained results are not used to verify these given logical labels.

Fujisawa [62] simplified the form definition language (FDL) by introducing a SFDL to avoid recursive expressions, and the extracted images containing bibliographic items are sent to OCR; but these results are not used for assisting the logical structure extraction.

Higashino *et al.* [80] proposed a knowledge-base segmentation for document understanding. A form definition language (FDL) is used for representing the layout rules of a document; then the regions containing bibliographic items, such as the title, authors and other important items can be extracted. In the layout rules, only position information is used.

Tsujimoto and Asada [219] presented a document understanding method based on tree structure transformation. Both the geometric structure and logical structure of a document can be represented by tree structures, and can be transformed into a logical structure tree that does not need the assistance of OCR results. Therefore, in these systems, the obtained macro structure is often not accurate because of the ambiguous boundaries of some typographical attributes.

A more recent system Porter and Rainero [168] intended to recognize both the logical structure levels, but the implemented part can only receive a postscript file as input.

Recently, a few systems have addressed some aspects of micro structure analysis. Chenevoy Belaid [27] and Belaid [13] described a knowledgebase system for structured document recognition: multi-specialists have been used for different tasks, and the micro structure of bibliography references was analyzed.

Dengel [41] has presented a document analysis system IODA (Paper Interface to ODA), which includes both tasks: text recognition and text analysis: the former is to explore the captured text of logical objects: and the latter syntactically checks some simple structure objects, such as the sender, recipient and date in a letter, and generates links between content portions and logical objects.

However, the methods used for both analysis levels are completely different; therefore, their system architecture is rather complex, and several studies, such as the integration between both analysis levels as well as the robustness and efficiency of their systems have to be done.

Many attempts in document image understanding have been made from early 80's. Early research on document image understanding can be classified into several classes. The first one is to analyze complex page layout such as newspapers, to extract articles consisting of a title, text and photographs for instance [80, 87]. The second one is to read printed documents with unknown layouts by enhancing character segmentation methods [235, 3]. Another domain of document understanding is for analyzing complex tabular forms to recognize hand-printed Kanji's to enter information to computer systems [154]. Some recent attempts also include address analysis of small pieces [227].

One of the important applications of such technology is automatic information extraction for automatic document filing [63, 155]. Document image filing systems using optical disks, which are being used in many offices, can store tens of thousands of document pages in terms of digital images [134]. In such a system, ease of information retrieval is very important.

Higashino *et al.* proposed a top-down document analysis method aiming at such applications [80, 225]. The feature of the approach is document layout knowledge used to parse two-dimensional physical document structures. They devised a knowledge representation language FDL, Form Definition Language, to describe generic layout structures of documents. The structures are represented in terms of rectangular regions. These generic descriptions are then matched against input document images to determine specific layouts. As a result, index information such as titles, authors' names, etc, can be extracted from front pages effectively. Although this top-down method is powerful, it is rather complex because of recursion.

Up to now, several techniques for document analysis and document understanding [226] - [195] have been proposed. Wahl *et al.* presented a smearing algorithm which is widely used for document analysis. Toyoda *et al.* [214] extracted Japanese newspapers articles using domain-specific knowledge. Okamoto *et al.* [160] analyzed papers containing mathematical expressions. Baird *et al.* coped with the extraction of chess games from the Chess Information. Esposito *et al.* [50] classified document types (patents, scientific papers, etc.) Higashino *et al.* [80] proposed a flexible format understanding method using FDL, i.e., a form definition language. Tsuji *et al* represented a document structure with a tree. Nakano *et al.* [153] built a business form understanding system incorporating character recognition. Inagati *et al.* [87] built a special purpose machine for Japanese document understanding. Masuda *et al.* [133] presented a prototype of a Japanese text reader. Baird [11] proposed a versatile text reader whose layout analysis components were designed to be language-dependent. Srihari *et al.* [195] surveyed the current status of document analysis. At the same time many approaches have been proposed for character segmentation/recognition especially for the segmentation of characters which touch each other. Kahan *et al.* discussed the segmentation of touching characters. Casey *et al.* [22] built a recursive segmentation and classification method for touching characters. Kooi *et al.* [122] analyzed the contour of an image of touching characters.

2.6 A Bottom-up Approach

Tsujimoto and Asada in [219, 220] presented a document image processing system used in a newly developed text reading system consisting of three major components: document analysis, document understanding, and character segmentation and recognition.

2.6.1 The Method in Brief

Lines of text are extracted from a page for recognition by the document analysis component. This procedure finds document constituents such as paragraphs, graphics, and text. As a result of this procedure, the geometric structure as a hierarchy of items on the page is extracted for modelling the relationships between characters, lines, columns and the page.

Logical relationship between the document constituents is extracted by the document understanding component. In fact, a tree can represent the geometric structure of a document, obtained in the document analysis phase. On the other hand, another tree can represent the logical structure of the same document using a small number of geometric rules introduced to transform the geometric structure into the logical structure.

Characters are extracted from a text line and recognized by the character segmentation and recognition component. Characters which touch each other may have several candidates for their break position, and any segmentation area might possibly fit several alternative characters. Therefore an efficient resolution of ambiguities at each stage is a crucial issue in practical text reading. The authors' approach in [219, 220] to do this is based on the heuristics of character composition as well as on recognition results for omnifonts.

More than a hundred documents used in experiments have proven that the proposed approaches to document analysis and document understanding are robust even for multicolumned and multiarticle documents containing graphics and photographs. Moreover, Experiments have also shown that the proposed segmentation and recognition method is robust enough to cope with omnifont characters which frequently touch each other.

Logical Structure

A document is normally composed of several articles, each of which consists of many different components connected to each other logically in a hierarchical structure. These components are a title, an abstract, subtitles, and paragraphs. For example, browsing the logical hierarchical structure one can infer that the title dominates the abstract, chapter, and sections, while subtitles dominate paragraphs. Thus, a document has a logical hierarchy that could be represented as a tree.

Geometric Structure

A document image is composed of several blocks, each of which represents a coherent component. Each coherent component corresponds to a set of lines with the same typeface and a consistent line spacing. The geometric structure, which means the relationship between blocks could also be described in a tree. Of course, the geometrical tree must be constructed from the tree that represents the logical structure of the same document else one can immediately construct the tree that represents the geometrical structure directly. To do this one needs some rules that define how to construct both trees. These rules are related to a previous understanding of a document.

The approach as in [219, 220] is, first, to extract the words from a document image, which are then merged into text lines. Text lines are then combined into blocks which usually correspond to paragraphs. This approach as indicated in Section 2.3.2 is called bottom-up.

According to the parent-child relationship between blocks, a geometric structure is generated. These relationships are established by examining the column a block belongs to, and its vertical position.

Since efficiency of processing is an important requirement of document processing system, a run-length representation of an image is used instead of a bit-map representation. A text line extraction algorithm is employed and is divided into four parts. The first is to extract adjacent components as a segment. The second part classifies the segments into text lines, figures, graphics, and so on. The third is a merging process for adjacent segments which are classified into text lines. The last is another merging process for the segments in the same column defined by the column boundaries. Words are usually extracted by both the first and second subprocesses, and text lines are obtained in the third subprocess. The fourth process is added to cope with cases where a long blank between words prevent the words from merging into a text line.

Block Extraction and Generation of The Geometric Structure Tree

In order to describe how each segment is related to the column position, a group number is given to a segment. The group number g is given by the following equation:

$$g_n = \sum_{i=1}^{N} cp[i] * 2^{(i-1)},$$

where $cp[i] = 1$ if the segment is in the Di^{th} column, $cp[i] = 0$ otherwise, and N is the number of columns.

Now, a block will contain all vertically adjacent text lines with the same group. Here, vertical adjacency is determined by a threshold defined by the line interval. Lastly, a geometric structure tree is generated where each node of the tree represents a set of adjacent blocks with the same *group number*. Parent-child relationship between nodes are established by examining group numbers and their vertical locations. Nodes not dominated by others are directly connected to the root node. These nodes are ordered by examining their group numbers and vertical locations so that a block to the left and to the top precedes the other.

2.6.2 Document Understanding

The algorithm that transforms the geometrical structure into a logical structure, as presented in Figure 2.3 is composed of four rules that define the conditions under which an element in a node is moved from the geometrical structure to the logical structure.

2.6.3 Basic Algorithm for Tree Transformation

The algorithm is composed of four transformation rules that define the condition under which an element in a node list is moved. These rules are illustrated in Figures 2.4, 2.5, where **H** indicates a *head* block, **B** indicates a *body* block, and **S** indicates that a block can be either *body* or *head*. In the tree, each node is sequentially numbered in the depth-first order. This is called depth-first indexing.

Four transformation rules are described below. Through this transformation process, a node which becomes *NULL* is deleted:

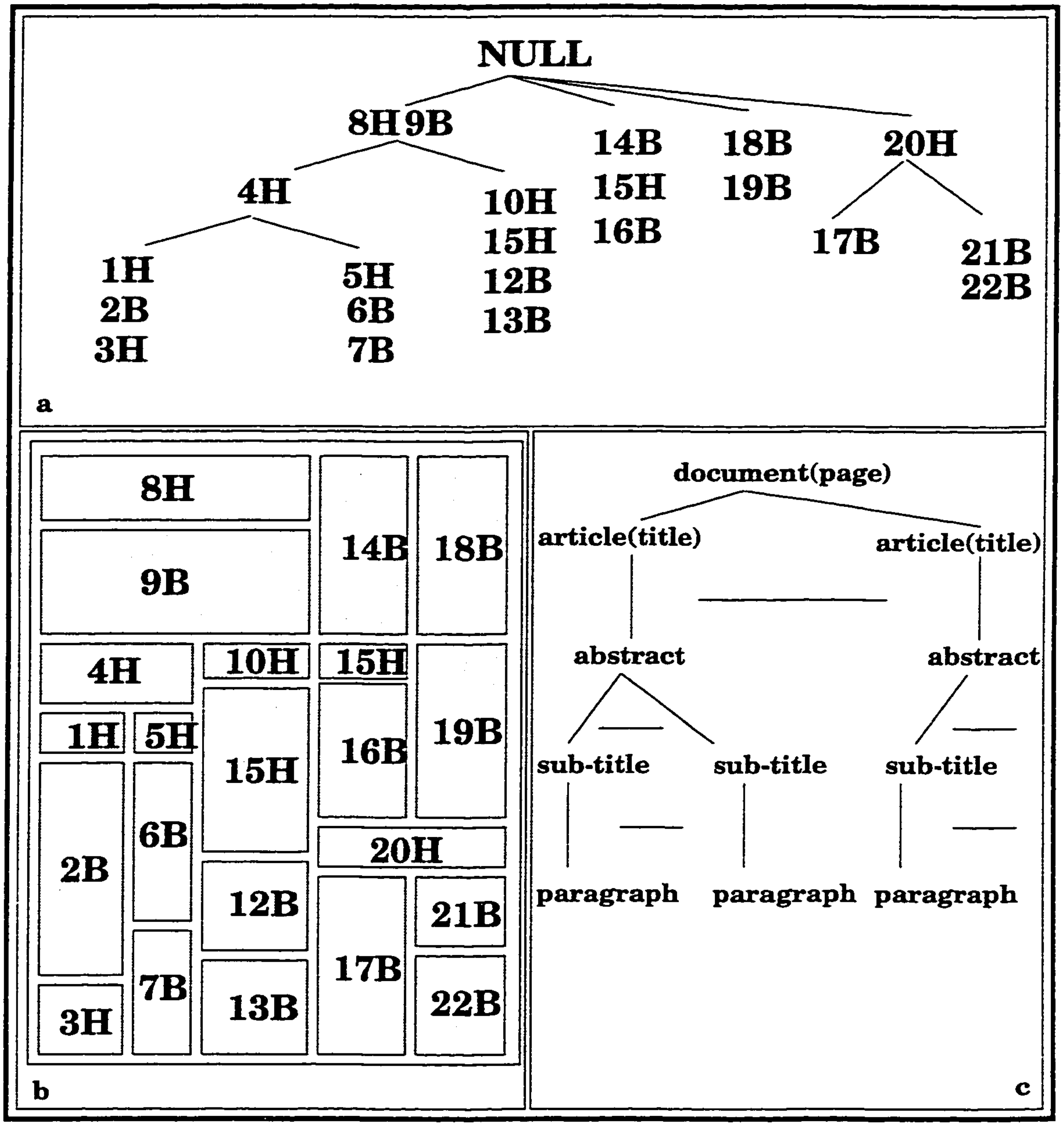

Figure 2.3: (a) Geometrical tree that corresponds to the sample document in (b). (b) Document divided into various blocks, (c) Logical structure tree.

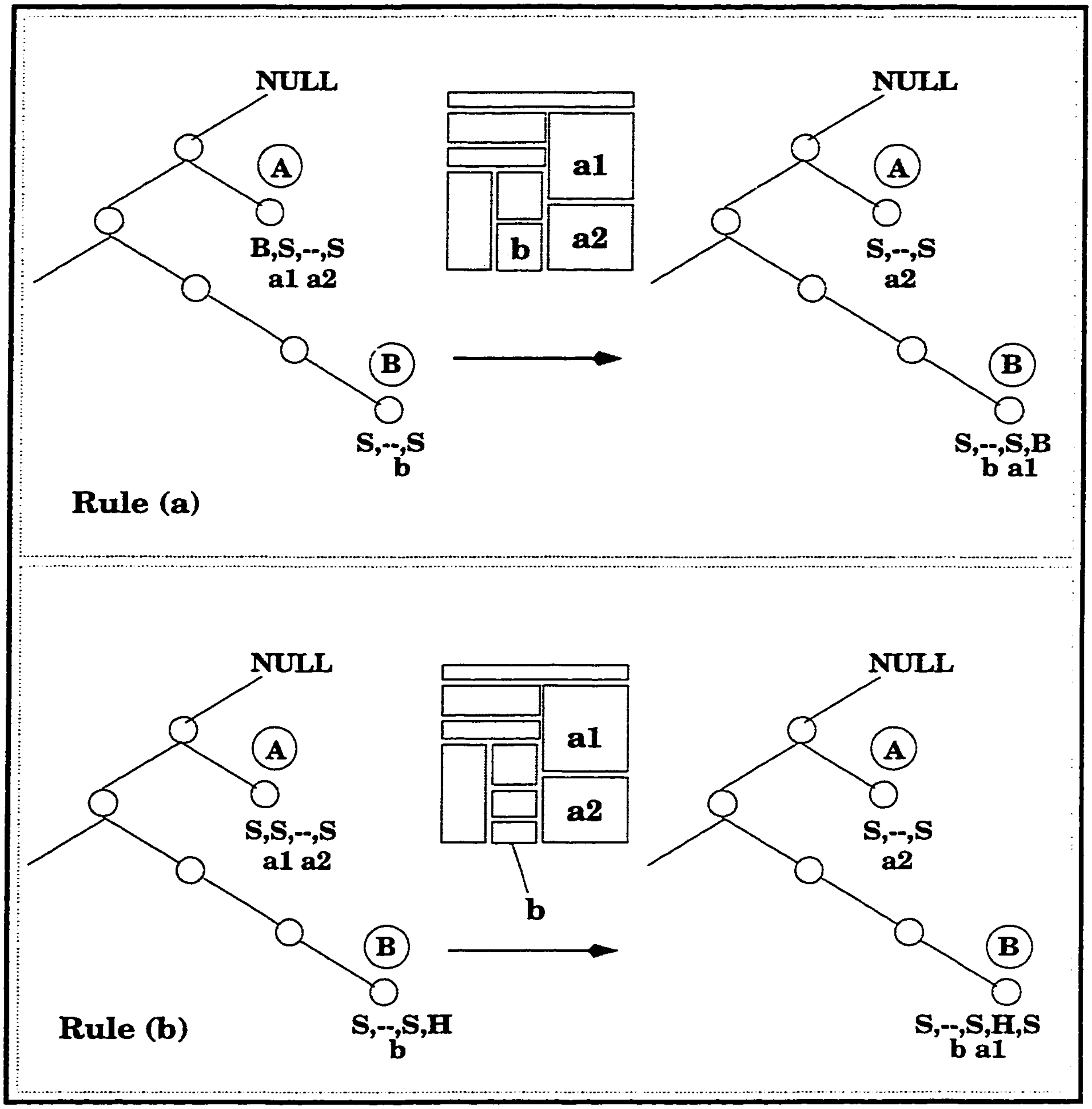

Figure 2.4: Rule (a): This rule is based on the observation that each title has a single set of paragraphs. Rule (b): Similar to rule (a).

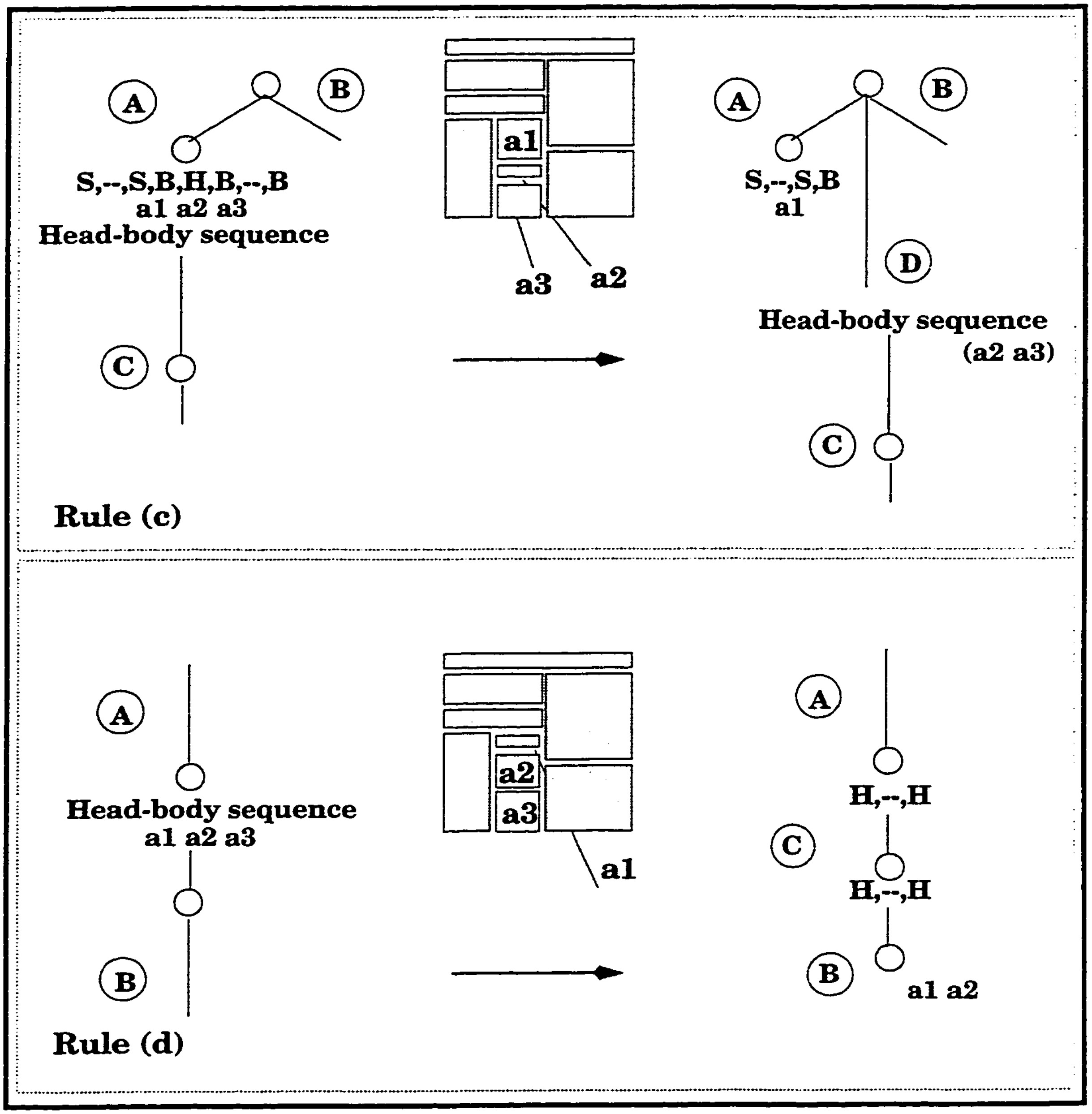

Figure 2.5: Rule (c): Extracts chapters of sections for a subtitle. Rule (d): attaches a unique class *(head/body)* to each node.

Rule (a):

- *If*

  a node (say **A**) is a terminal node and the first element of node **A** is a *body* and the preceding node (say **B**) in the *depth-first indexing* is a terminal node,

  then

  remove the first element from node **A**, and append it to the last element of node **B**.

This rule is based on the observation that a title has a single set of paragraphs as child in the logical structure. Therefore, if the parent of a terminal node containing *bodies* has several children, then only one of them can be the true child of the parent. It is reasonable that the eldest child represents the text dominated by the parent and that the others should be merged to it.

Rule (b):

- *If*

  a node (say **A**) is a terminal node and that is not connected to the root node, and the preceding node (say **B**) in the *depth-first indexing* is a terminal node, and the first element of node **A** is not *NULL*, and last element of node **B** is a head,

  then

  remove the first element from node **A**, and append it to the last element of node **B**.

This transformation is the same as that for Rule (a). The difference is that the first element of node **A** does not need to be a *body* if the last element of node **B** is a *head*.

Rule (c):

- *If*

  a node (say **A**) contains a head block, and it is not the first element of the node, *then*

generate a younger node (say **D**), and remove the *head-body* sequence that begins with that *head* block and ends with the last element of node **A**, with child of node **A**, if any, and attach them to the younger node **D**.

When a node includes more than one *head-body* sequence, a new node is generated for each *head-body* sequence by applying this rule recursively. This rule is mainly for extracting chapters and sections headed by a subtitle.

Rule (c):

- *If*

  there is a *head* block sequence in a node, and it is the first part of the node,

  then

  generate a child node, and move the *body* sequence that follows the *head* sequence to the child node.

Figure 2.5 shows a case in which node **A** has a single *head* sequence and a single *body* sequence. In this case, the *body* sequence is separated from node **A** and moved to a new node **C**, which is a newly generated child of node **A**. By this rule, each node has either a *head* or *body* sequence. This rule is applied after rules (a), (b), and (c) have been completed.

The next step is an interpretation process, where a label is attached to each node. Here, the labels include *title, abstract, subtitle, paragraph, header, footer, page number*, and *caption*. If a child of the root node has children and is a head sequence, it represents a *title*. If it has no children and it is a *head* sequence, one of the labels *header, footer, page number*, or *caption* is attached to it according to its location on the page. For example, a block which is centred and located at the bottom of a page is a *page number*. Any *head* blocks other than children of the root node are *subtitles*. *Body* blocks in terminal nodes are normally *paragraphs*. A *body* which is the eldest and whose next child is a *subtitle* represents an *abstract*. A *body* block with children also represents an *abstract*.

Figure 2.6 shows an example of the transformation process which generates a logical structure from the geometric structure of the document shown in Figure 2.3.

2.6.4 Experimental Results

The methods proposed in this work [221] were implemented on a recognition board consisting of a RISC processor and memory. A scanner with a resolution of 300dpi was directly connected to this recognition board. All procedures were realized in software alone.

Experiments on document analysis and document understanding were carried out on 106 documents taken from magazines, journals, newspapers, books, manuals, letters, and scientific papers. There were 12 documents whose layouts were not correctly interpreted. This was attributed to three reasons. One was that the geometric structure was not correctly constructed because of errors in segment and/or block extraction in the document analysis process. Another reason was because the proposed transformation rules did not cover all actual layouts. Four documents fell into this category. A document whose title or abstract was located in the middle of the text blocks belong to this category. The last reason was that documents did not have geometrically and logically defined hierarchical structures. These types of documents, however, are usually in the minority. One of the test documents happened to belong to this category.

2.7 A Top-down Approach

In this section, we will present a top-down method for document image understanding to extract information from documents. First, a language FDL , Form Definition Language, is introduced which can represent generic layout structures as a set of rectangular regions, each of which is recursively defined in terms of inclusive rectangular regions. The generic descriptions are matched against input document images to identify concrete layout structures where specific items on a page are extracted. Although this approach is powerful, it is rather complex and a simplification is possible. A lan-

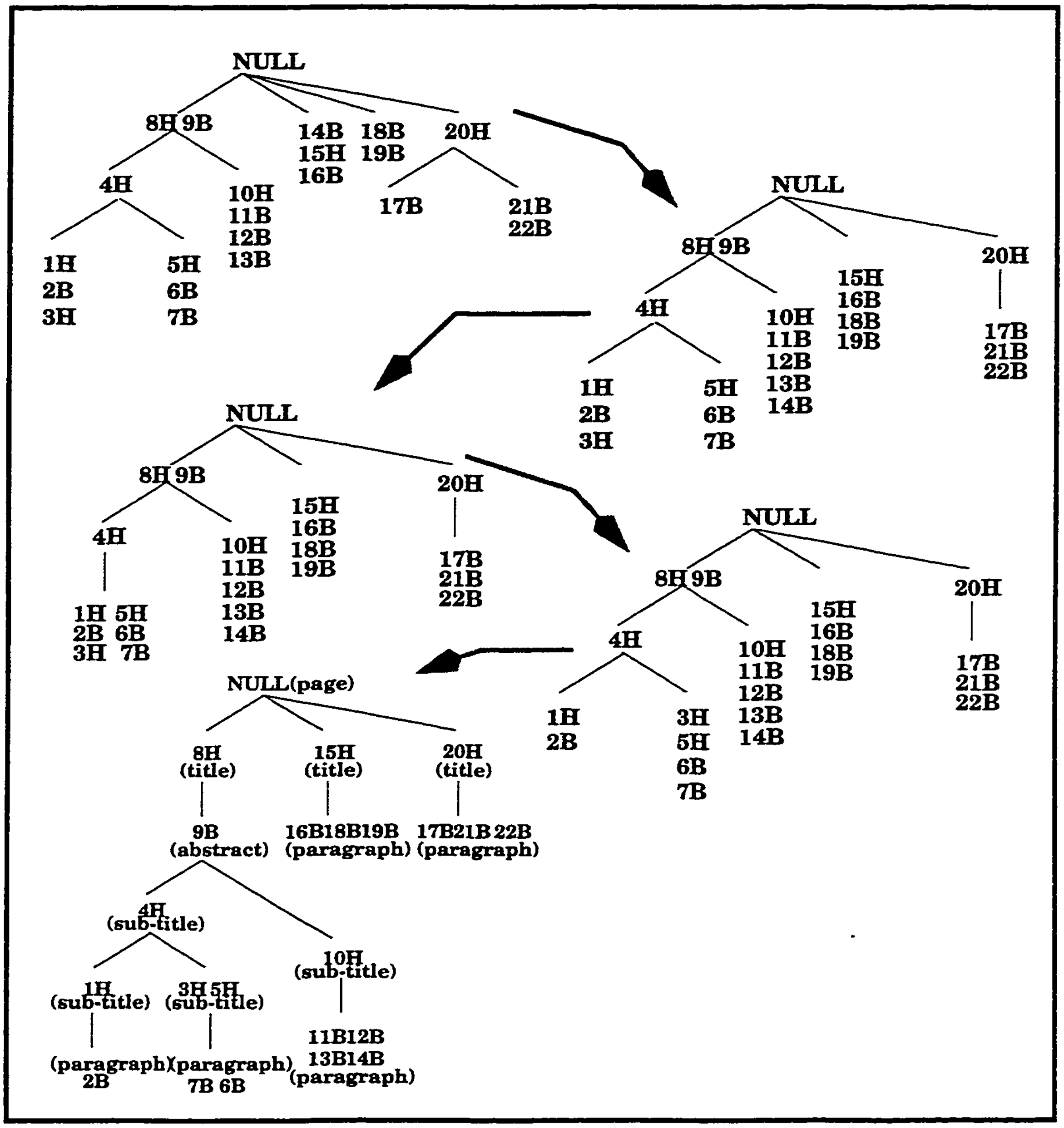

Figure 2.6: A sample process of tree transformation of a given form.

guage SFDL, a simplified version of FDL, can be used for rather regular document forms to describe characteristic patterns of the document in terms of templates. Templates are rectangular regions among which spatial constraints are defined without recursions. By matching such templates, input document images can be classified, and bibliographic items such as title and author names can be extracted. Experiments have shown promising results and it can be applied to an automatic document filing system.

There are three major steps in the proposed method: preprocessing of document images, extraction of connected pattern components, and matching with layout descriptions written in SFDL. The system can be considered as document class recognition since in the matching step, connected pattern components are matched to a plurality of SFDL description, each of which is defined for a class of document pages. As a result, one description that matches to the input is identified.

2.7.1 The Grammar of SFDL

Classification of different pages of documents is an important function of an automatic filing system, where many different documents will be stacked for automatic entry. Classes include those of a front page (header page), title page, text page, figure page, etc. In the case of such documents that require automated filing, pages have typical features in their layout styles. Especially, front pages have characteristic features, in other words, some regularities. For instance, pages have fixed areas and variable areas. Such regular forms include patent application forms.

The main purpose is document classification which is achieved by matching input features to the pre-registered features. Based on the existence of typical areas and their consistent geometrical (spatial) interrelationship matching is performed. In this context, document classification is made easier. Moreover, matching enables the identification of important areas of titles, author's names, etc. The purpose of the SFDL language is to describe such layout features. In SFDL, templates can be defined that specific pattern areas which are selected as characteristic patterns on a page. Normally, such a pattern is a string of characters. Spatial relations can, then, be defined to constrain some geometrical interrelationships among these pattern areas. There should be as many templates as defined as to correctly identify a document class. A template for a characteristic pattern can be defined by the following TEMPLATE statement:

$$\mathit{label}\ TEMPLATE[n]$$

where *label* represents an arbitrary name attached to the template, and n is an optional identification number. In SFDL, upper case and lower case letters represent reserved words of the SFDL language and parameters, respectively.

Some spatial constraints on characteristic patterns are defined in terms of absolute and relative constraints. In fact, SFDL specified four kinds of constraints that can be defined. These constraints, can be specified by three kinds of modifier statements, or AREA, SIZE, and EDGE statements which are as follows:

$$AREA\ l\ r\ t\ b, \quad SIZE\ w_1\ w_2\ h_1\ h_2, \quad EDGE\ \pi\ a_1\ a_2$$

Here, α and β specify one of the four edges of a template as in the EDGE statement, and a_1 and a_2 are bounds of the difference between the two specified edges. For instance, the following example statement:

$$REL\ U\$ABC_1 = B\$\ ABC_2\ 150\ 230$$

specifies that two patterns included in templates ABC_1 and ABC_2 align themselves such that the vertical distance between the upper edge of ABC_1 and the lower edge of ABC_2 should be between 150 and 230 mm.

2.7.2 The Template Matching Process

Connected pattern components that are extracted from a preprocessed document image are matched against form description which are represented in terms of templates extracted from document images as pointed out in the previous section. The image is considered to match the form description, and is assigned the corresponding page

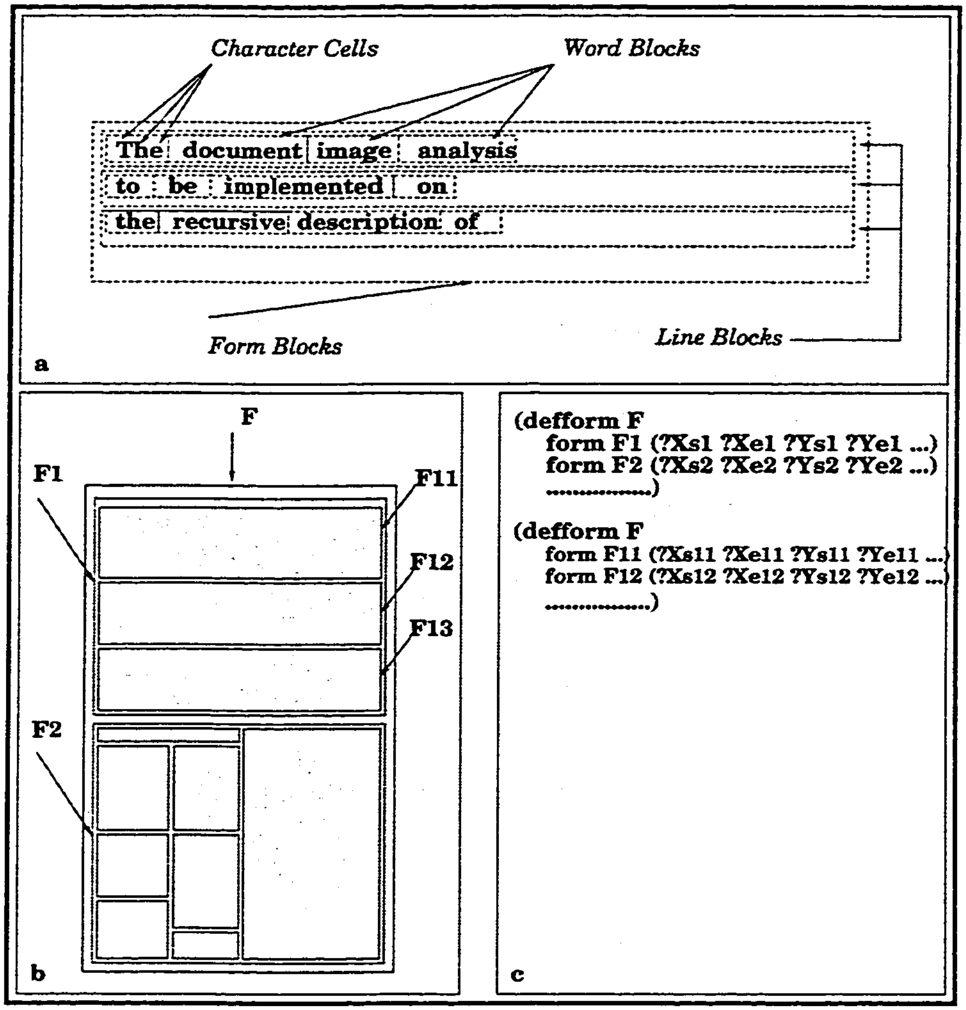

Figure 2.7: A Top-down Approach for Document Processing.

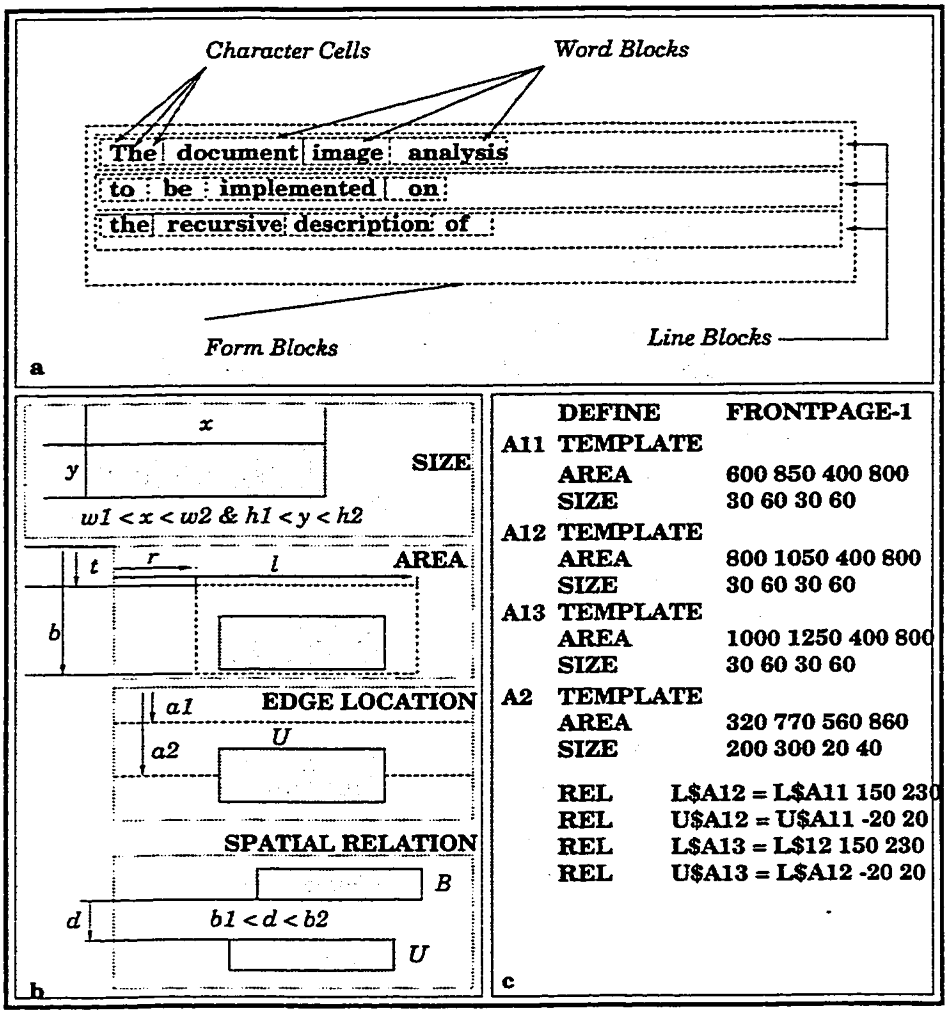

Figure 2.8: A Simplified Form Description Language SFDL.

class, if each template in a form description finds a text line pattern(s) (connected pattern component) enclosed in the specified area, and all templates in the same form description specifies the geometrical constraints. In this case, the image is assigned the corresponding page class. If such a match does not occur, the next form description is tried to match. The image is rejected as having unknown page class, if it matches none of the descriptions.

Connected pattern components extracted from the preprocessed image are listed in a set P of candidate line patterns p_i at the start of the matching process. A set of templates, $T_k = \{t_{k_j}\}$, is read out from form description file F which has as many sets, T_s, as the number of forms defined. Candidate line patterns that are not located in any of the template areas are excluded from the candidate list P, by using the SIZE, AREA and EDGE statements as constraints. During this step, each line pattern is identified by a template t_{k_j} that includes it. In the next step, every line pattern p_i is tested if there exists other line pattern q that needs the geometrical constraints defined in the REL statement for template t_{k_j}. If there exists one, it remains as it is. Otherwise, the line pattern is excluded from the candidate list. And this elimination process is repeated until no change in the candidate list occurs.

After the elimination process is performed, it is tested if there remains at least one candidate line pattern for each template. If it is true, the matching succeeds and the form T_k is recorded as candidate of the page class. On the contrary, if there is any template that includes no candidate line pattern, the matching is a failure. Sometimes, there remains templates that have more than one pattern. This case is also a failure as a definition.

The corresponding page class is accepted, if the matched description T_k is unique after matching all the form descriptions in the file F. If the uniquely matched description is not the right one, then it means a misclassification. If there are no descriptions or more than one description matched to the input, on the other hand, it is rejected.

2.7.3 Form Description for Identifying Page Class

A very important property of a document classification approach is to automate the classification problem. In fact, the system should be able to scan forms and documents and extract knowledge from these forms automatically. Unfortunately, this is not the case here. In a step to automate document filing, documents to be stored automatically should be studied in advance so that classes and their features are identified by the system designer. Then, form descriptions are written manually from detailed observations on typical sample images. By carrying out several test runs, misclassified and rejected images are examined, and inappropriate form descriptions are corrected. This elaboration process must be repeated until satisfactory results are expected.

The form description process corresponds to a training process for pattern recognition. However, what is different from pattern recognition is that structural features are to be extracted from samples and be represented in a language like SFDL. After identifying structural features, parameters should be trained.

As a future direction, we recommend that one must think of automatic learning rather learning that is based on the user's observation. It will be learning from samples with some abstraction process.

2.7.4 Experiments

Two series of experiments for page class recognition have been carried out to evaluate the SFDL method introduced in Section 2.7. In the first series, a set of sample documents of 106 pages from patent application documents was used. In the second series, a set of 81 pages from Japanese Patent Disclosure Bulletin JPDB was used. These documents were scanned with the resolution of 200 dot/inch and converted to binary digital images, each page consisting of 1728×2287 pixels. The recognition experiments for the two sets were carried out independently with two separate sets of form descriptions.

2.7.5 Summary

A simple but effective method for document analysis has been proposed in [62]. A language SFDL, a simplified version of FDL, has been developed. By using SFDL, the form of a document can be described in a simple form without recursive expressions.

The run-length filtering is applied to input images as preprocessing in order to treat lines directly in the form matching. In the image analysis, form descriptions are matched against the preprocessed patterns, and the page classes of input are identified. Bibliographic items such as a title and author's names can be extracted as a result of this matching process. Images in the extracted regions are sent to a character recognition processor and then converted into code information.

Two sets of patent related printed documents were used as experiments of the proposed method. For the patent application documents, 104 pages out of the 106 pages were correctly recognized and two were unrecognized because of noise. For the second set of patent disclosure documents, all 81 pages were correctly recognized. If we average the results of two series of experiments, the recognition accuracy becomes 99%. The effectiveness of this approach has been proved in spite of the simplified capacity in form description.

Future research should be directed to a development of automatic form description generation, which is machine learning of structures from samples. Such an approach will include automatic structural feature extraction and abstraction process. By incorporating the learning capacity, this approach will realize a full automatic filing function for the optical document filing system.

2.8 Difficulties and Motivation

There are a number of problems in a practical document analysis system. Some of them will be listed here:

- Noisy inputs, for instance, image skewness, noise resulting from scanners, deformation of scanned documents, decay of document caused by time, and many others as shown in Figure 2.9.

- Different types of background images printed on different documents, such as bank cheques, that complicate the process of segmenting the background and the foreground data. Figure 2.10 illustrates this.

- Ambiguous boundaries of different typographical attributes; for example, the possible attribute values of some overlapping logical entities, which often lead to many solutions for a document, as illustrated in Figure 2.9.

- The inborn difficulty of a top-down analysis in error handling mentioned by Ingold and Armabgil [88], which makes it non-trivial to deal with documents and often lead to no solutions.

- The expensive time and space requirements for parsing. In syntactic pattern recognition Bunke and Sanfeliu [20], Gonzalez and Wintz [67], parsing often refers to the process of searching a path through a directed graph for an input string; most of the parsing algorithms have an exponential computing complexity.

The above challenging difficulties affect the robustness and efficiency of a document analysis system. Because the available document analysis systems are mostly at their prototype stage, these problems have not yet been solved well. Until now, few of them have paid sufficient attention to these problems. The IODA system of Dengel [41] did make considerable efforts: probability theory was used to handle uncertainty, a large statistical database (SDB) was built to assign possible attribute values and their probabilities to all possible logical entities, and a best-first search algorithm was applied by Korf in [123]. However, this probability theory has strict requirements for the independence among many factors and for the combining rules among factors, which are hard to be satisfied in practice. In order to build a document model, lots of statistical samples are needed, which makes it impossible for a user to write a document description by himself. Furthermore, the best-first search has still an exponential computing complexity in worst case as pointed out by Korf in [123]; therefore, it could be used only for some small scale applications.

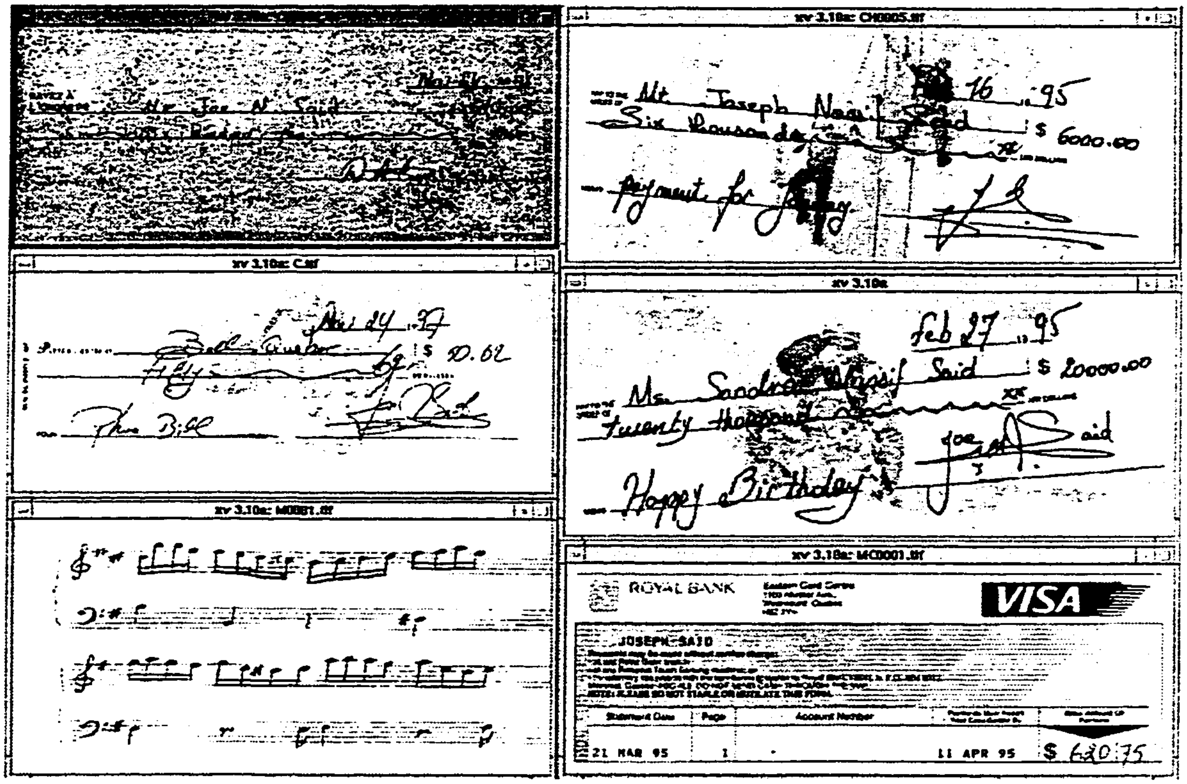

Figure 2.9: Difficulties in document processing.

In order to deal with some input noise and small errors, an error-correction parser is often used for evaluating string distances as mentioned by Bunke and Sanfeliu [20]; however, this algorithm is not suitable when too many possible document patterns exist.

2.9 Documents and Forms Considerations

This section will discuss form considerations which include the form's background, design, identification and indexing. Addressing the form issues posed in this section will save storage space, speed up the performance and assist the processing and recognition engines.

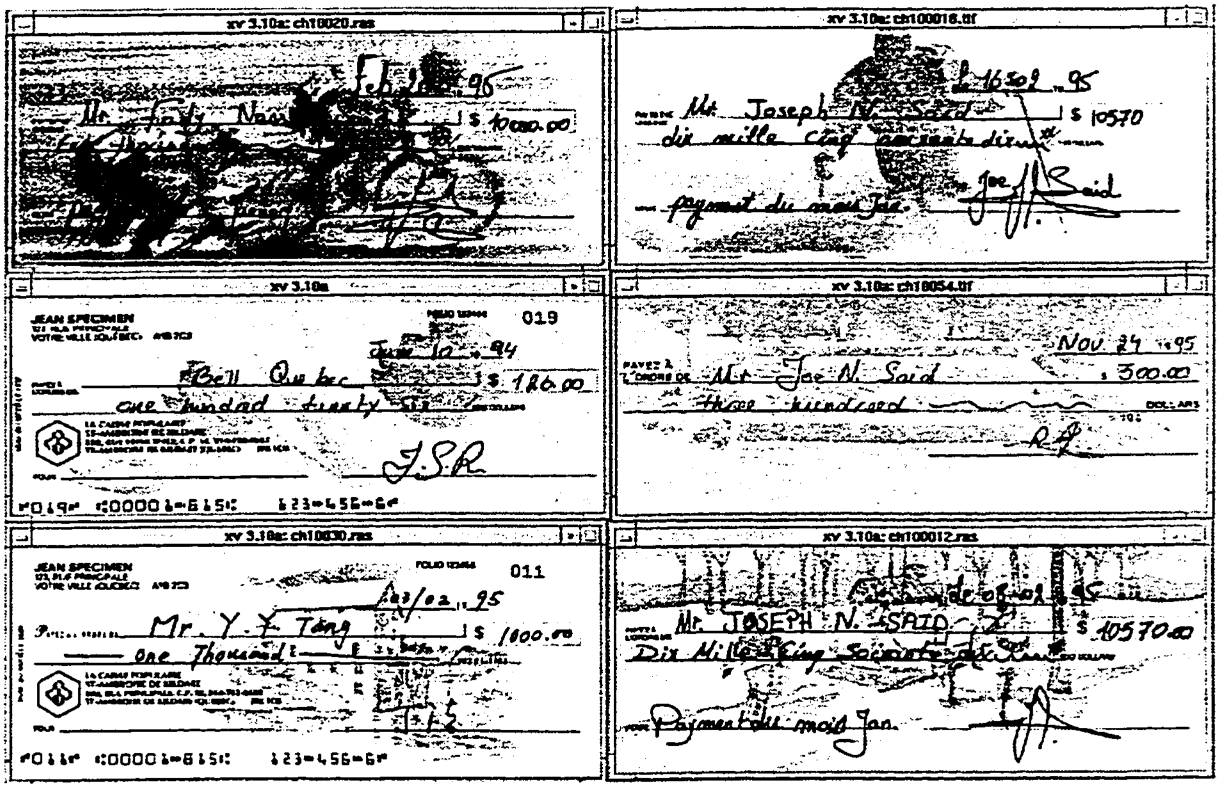

Figure 2.10: Complicated background printed on cheque images.

2.9.1 Characteristics of Form Documents

Specific characteristics of form documents have been identified and analyzed in [211, 210, 209, 240] which are listed below:

- In general, form documents may consist of straight lines which are oriented mostly in horizontal and vertical directions.
- The information that should be acquired from a form is usually the filled data. The filling positions can be determined by taking the above lines as references.
- Texts in form documents often contain a small set of known machine-printed, hand-printed and handwritten characters, such as legal and numeric amounts on cheques. They can be recognized with current recognition techniques.

2.9.2 Form Background

The form background is the same in every form of that type. If the repetitive form can be eliminated from an image and only the data remains, the size of that image could be as much as one tenth of the size of the complete image. Three possible ways to eliminate the form background are:

1. Print the form in drop-out ink, a non-carbonized ink that the scanner does not see at certain scanning thresholds.
2. Suppress the form image with form suppression software.
3. Utilize a filter on the camera lens that blocks out the form's background colour but does not block out the ink of the data input on the form.

2.9.3 Form Design

Form design becomes especially important when optical character recognition is employed. Optical character recognition machines will read the individual characters, but custom software must place meaning to those letters and prepare them for input into the processing system. The following are some form design suggestions:

1. For OCR readability provide quiet space around the data fields and barcodes. This will decrease the possibility of data running from one field to the next. Avoid shaded fields. They decrease contrast and make differentiation of data and background more difficult.

2. Use the form to create a structure for both optical character recognition and data entry. You can create a template to identify where to look for data and what to do with that data once it is read. Here, the quality of scanning is the key. If the form is skewed, data will not fall into the prescribed areas of the form for reading.

3. Avoid free form comments. Lead the user in filling in the desired values by providing codes, examples or instructions. Precode the form, if possible. This saves user time, examiner time and increases accuracy.

4. Make the input fields consistent with the processing system that will receive the data. Separate the fields on the form that are separate on the system. For example, first name, middle initial and last name are three different fields on a processing system yet are often merged into one field on a form. As another example, consider the city, province/state, and postal/zip code used as labels in most entry systems. Such fields should be separated in order for the processing algorithms to be able to correctly analyze, extract, and recognize these filled items.

2.10 Commercial State and Future Trends

Advancements in document analysis and recognition is very much affected by advancement in other fields such as computer hardware and technology. In fact, hardware advances will continue to provide easier computer input, faster computation, and greater storage, resulting in an improvement in the field of document analysis and recognition. It is thus with some sense of dissatisfaction that we see the field of document analysis seem to progress with less speed. However, this is due not so much to a slow improvement in other fields as it is to the distance still required to reach our ultimate goal of attaining recognition results similar to that of a human. This goal will not be attained soon.

Having stated that there is a long way to reach the ultimate goal, it is very important to look at the current status of the field of document processing and recognition, describe and spot the difficulties that research is facing, present practical solutions to critical problems that contributes to the advancement of a better technology in this research area. For this reason in this report, we will suggest below where progress should continue and where it is especially needed. In later sections, we will present our new formal model for visual data processing of business forms and bank cheques that used some techniques, extended others, and introduced many new and intelligent operators for document and image processing.

In what follows we will point out where progress should continue and where it is especially needed:

- Nowadays documents are scanned in binary form, although gray-scale and colour scanners are becoming increasingly prevalent. In the future, more gray-scale and colour scanning will be employed and more powerful techniques for scanning, smoothing, and segmenting will be integrated in these scanners. The primary advantage of these is simply more pleasing images, especially for those of gray-scale and colour pictures. However, this will also enable more sophisticated document and image analysis to be performed. Thresholding will be improved when more complex algorithms can be applied digitally rather than today's simpler optical methods being used. Multithresholding can be done to separate, for instance, black text from white background. Segmentation can be performed to distinguish among text, graphics, and pictures. The result is better image understanding that will positively affect further processing.

- Image enhancement to remove salt-and-pepper noise due to scanning is available on current systems. Future systems should employ more sophisticated filters on scanned document images that will use some knowledge of the document components to provide recognition-based cleaning. Of course, the furthest extent of this speculation, will be to perform OCR along with font and format recognition, and then to redisplay a page exactly as before. The challenging difficulty here is that the recognition must be flawless so as not to introduce more than the initial amount of noise that will disturb the recognition results.

- Today document scanners and systems can be purchased with special software to correct up to about 10 degrees for the small amount of skew often present when documents are placed on the scanner. Future systems should be able to reorient document from any skew angle, even from upside-down. Related to the increased ease and utility of scanning systems are the improvement and reduction of size in the hardware. In fact, portable and hand held scanners (usually packed with OCR software) are currently available. In the future, scanners will be incorporated into portable computers so the portable, office-computer, machines, copying machine, fax, and telephone will always be available.

- While rudimentary structural layout analysis is already performed in some document systems (mainly to separate text columns), more complex tasks, such as paragraphs and caption segmentation, should also be performed in the future. On a further horizon it should be determined if it is possible that structural-layout analysis is to be combined with functional labelling. Since the latter requires knowledge of the document, automatic training procedures will be developed to facilitate this task. Document description will be made in some standard language, such as ODA or SGML.

- Rice in [176] showed a system for printed text with 99.5 percent to 99.9 percent recognition accuracy of individual characters for high-quality text. This recognition drops off to 90 to 96% accuracy for full words. Although these may appear to be quite good at first glance, they correspond to one incorrectly recognized word for every few lines of machine-printed text. We recommend that by the use of word and document based information, major progress in recognition and system's accuracy should be improved. This improvement is not only reached by developing better techniques, but also by using knowledge from documents.

Currently, dictionary lookup is used to change unrecognized words into correct ones that have similar word features. In the future, increased computer power and memory (plus fast indexing methods) will enable practical searching of large spaces so corrections can be made using syntactic and even semantic knowledge of the text being analyzed. Even with these improvements, we do not expect to see flawless OCR in the near future. Near flawless OCR will be available only for constrained situations, such as a limited character set of 0 to 9. Erroneous OCR can be tolerated in some applications, such as one to search for keywords in a paper where pertinent keywords will likely be repeated. The evaluations are based on character and word recognition results for a range of document conditions, such as skew and document quality.

- Most applications for most current OCR apply to the standard character sets: in particular, numerals and the Roman, Kanji, and Arabic alphabets. However, there should be other character sets and symbols that must be recognized in future systems. An example is the larger symbol set used for mathematics. Character sequence along a text line in mathematical equations is not strictly linear which is not a trivial problem to pursue. Add to this, the difficulty of superscripts and subscripts and bounds of integrals and summations that are above and below other symbols. Without forgetting that superscripts and subscripts are usually in a smaller font size, this makes the problem of recognition even more difficult.

- If printed text is a difficult problem, then handwritten text is more difficult to model. In fact for handwritten text, recognition rate is lower than those for printed text. Current off-line recognition rates for untrained data are about 96.5 percent for digits (0-9), 95 percent for uppercase letters only, and 86.5 percent for lowercase letters, all for isolated characters Wilkinson [234]. The recognition rates of off-line handwritten texts and those of machine printed texts are usually highly dependent upon training and the size of the character set. Although the off-line recognition rates are unsatisfactory for many applications, there are those applications that do very well with them. For example, for the postal application, although ZIP code reading is currently not very high, this still reduces by a huge amount the mail that must be sorted by hand. Besides applications like this, the most predominant applications of handwriting recognition in the short term will be highly constrained. These include trained and untrained digit recognition, recognition of pre-segmentation block characters, signature storage and recognition, and trained, single user handwriting recognition. Many of the other comments made above for printed text OCR apply to handwritten texts also.

- Current systems use different techniques for machine-printed and handwritten text recognition. Also different approaches are often used for different alphabets (for example, Roman, Kanji, and Arabic). Even tables and mathematical equations are most often analyzed using different modules (where they are analyzed at all in current commercial systems). A goal in character recognition is seamless and automatic multilingual, multisymbol recognition, where language and character type are recognized first, and then characters are recognized quickly and correctly.

Chapter 3

Recursive Thresholding

A general recursive thresholding method applied on cheque images.

3.1 Introduction

The problem of *image segmentation* has received considerable attention in the literature [162, 183, 215, 216]. Several methods have been proposed to tackle this problem, and two approaches are widely used in this context: edge-based [79, 121, 228] and region-based [16, 21, 98, 118, 175, 183, 193]. In edge-based methods, the local discontinuities are detected first and then connected to form longer, hopefully complete, boundaries. In region-based methods, areas of an image with homogeneous properties are found, which in turn give the boundaries. The two methods are complementary to each other and one may be preferred to the other for some specific applications like document image analysis. In their remarkable work, Sahoo *et àl.* [183], Pal *et al.* [162], and Trier *et al.* [215, 216] presented an excellent survey on image thresholding that tested the performance of many thresholding techniques.

In fact, there is currently a substantial and growing interest in the field of document image understanding where several research groups are developing and designing systems to automatically process and extract relevant information from documents such as engineering drawings, maps, magazines, newspapers, forms, and mail envelopes [28, 100, 101, 171, 172, 173, 181, 182].

Our interest in this thesis is to develop a general recursive thresholding technique for gray-scale images and demonstrate the need for such a new technique in the field of document analysis and understanding. Of course, the new technique also aims at preserving the topological properties of the extracted information that will be used for further processing.

3.2 Problem Background and Motivation

The objective of image segmentation is to separate the components of an image into subsets that correspond to the physical objects in the scene. The segmented components are then used by higher-level processes for interpretation and recognition. The image segmentation methods assume that the objects have smooth homogeneous surfaces that correspond to regions of constant or smoothly varying intensity in the image and that the intensity changes abruptly at the boundaries. These assumptions are mostly, but not always, valid.

One of the most powerful techniques for image segmentation is thresholding. The application of the thresholding techniques is based on the assumption that object and background pixels in a digital image can be distinguished by their gray-level values [118]. In some cases, such as text images, it is a priori known that the image contains only two principal gray tones. The histogram of such an image may be considered as representing the distribution of the image brightness as in Figure 3.11 (b,d). Based on the shape of the histogram, it is possible to determine an optimal threshold value for segmenting the image into two brightness regions. The result of this processing is an image with only two gray-levels, which correspond to the background and objects. This approach is referred in the literature as the global (or bilevel) thresholding. Over the past years, several techniques have been proposed for automatic global threshold selection, see [183] for a survey of the thresholding technique.

Global thresholding methods can be applied only to some images, where a clear foreground-background relationship exists [121] as illustrated in Figures 3.1, 3.2 and

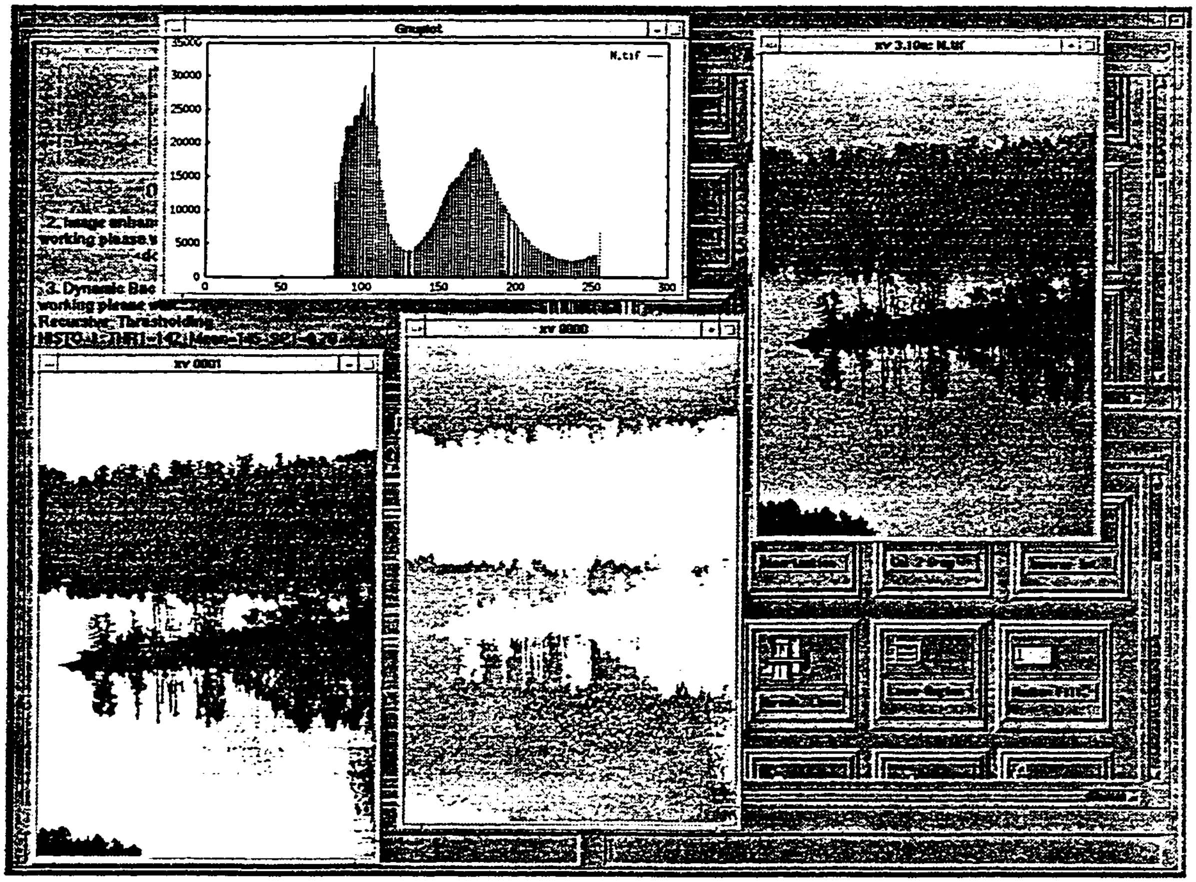

Figure 3.1: Image segmentation in the case of a very clear foreground-background relationship. The threshold value is 145 which is the deepest point in the valley as in the histogram. The background corresponds to the brightest object whose pixel values are greater than 145. The foreground corresponds to the darker object whose pixel values are less or equal to 145.

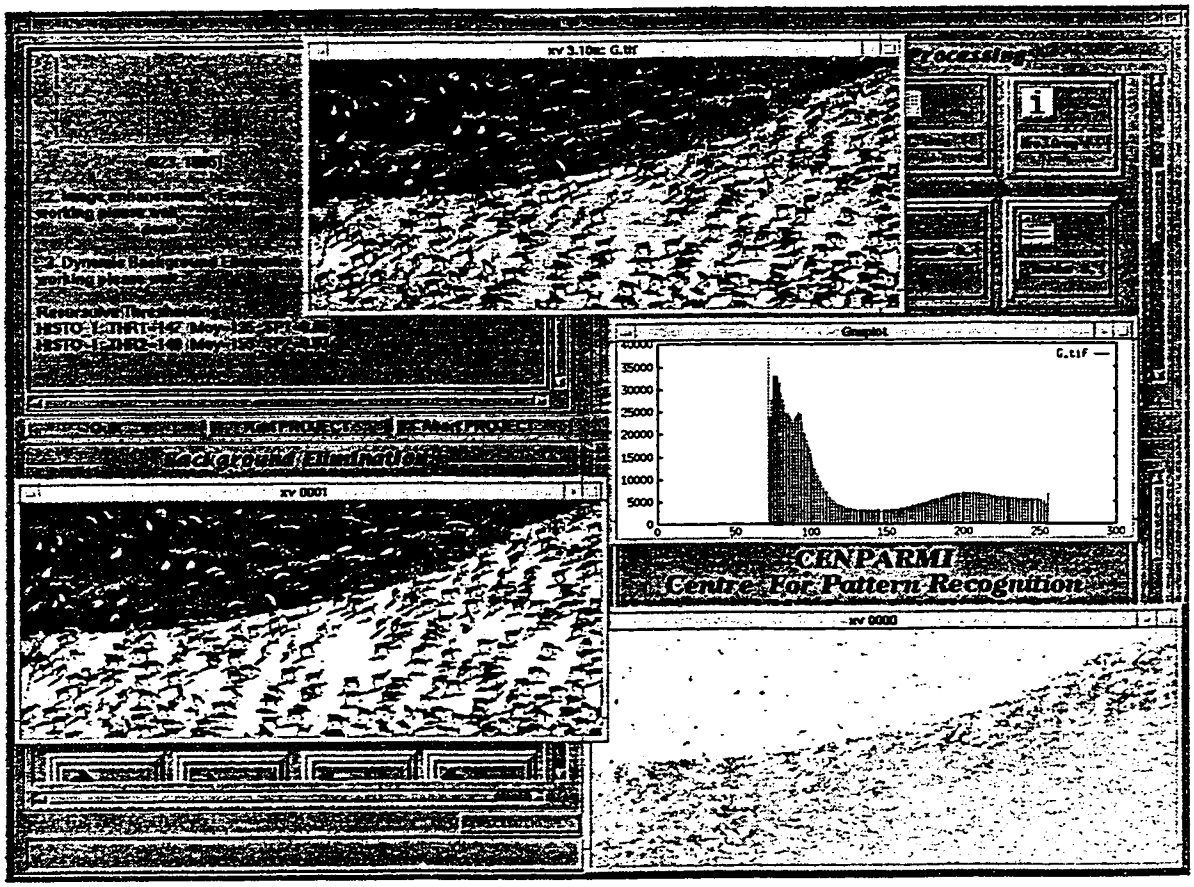

Figure 3.2: Image segmentation in the case of multiple objects with a clear foreground-background relationship between the brightest object in the image and the rest of the objects. The threshold value is 148. The background corresponds to the brightest object whose pixel values are greater than 148. The foreground corresponds to the darker object whose pixel values are less or equal to 148.

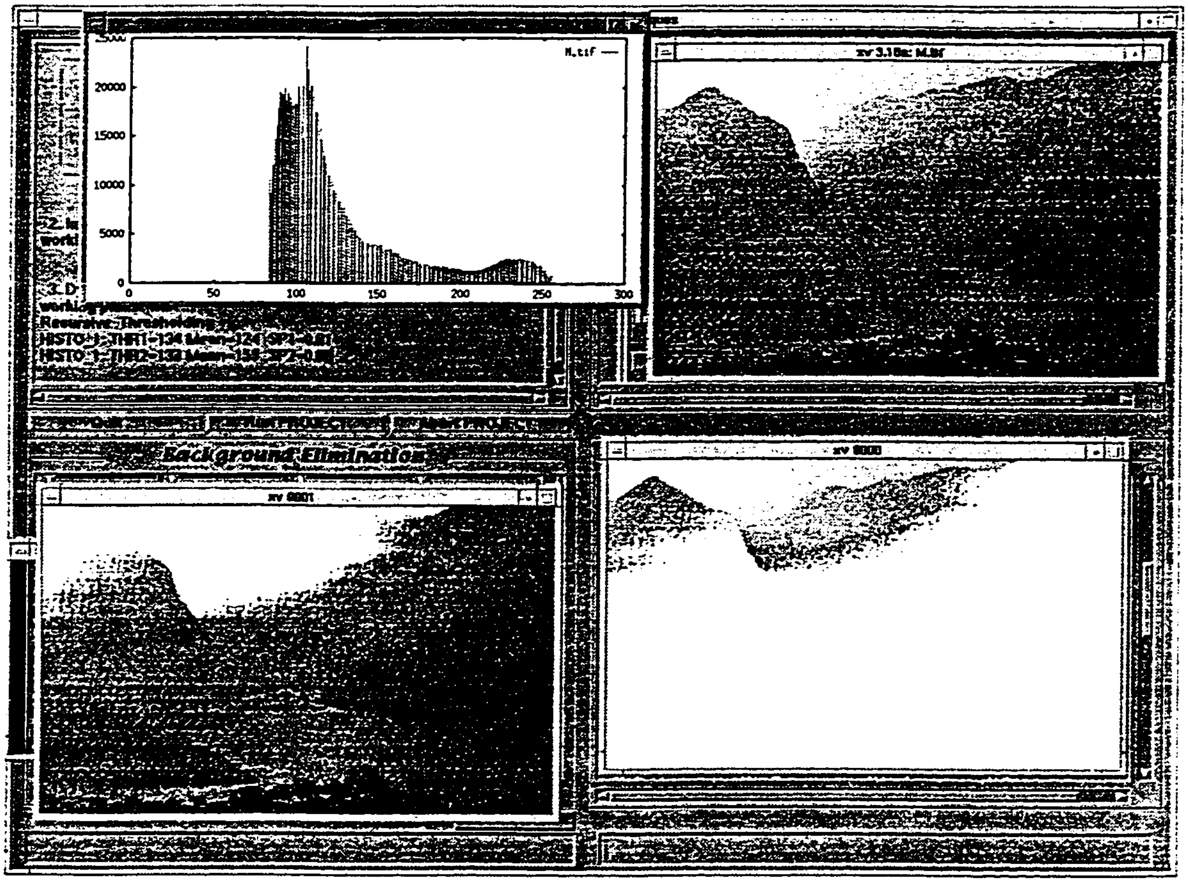

Figure 3.3: Image segmentation in the case of multiple objects with an acceptable foreground-background relationship between the brightest object in the image and the rest of the objects. The threshold value is 133. The background corresponds to the brightest object whose pixel values are greater than 133. The foreground corresponds to the darker object whose pixel values are less or equal to 133.

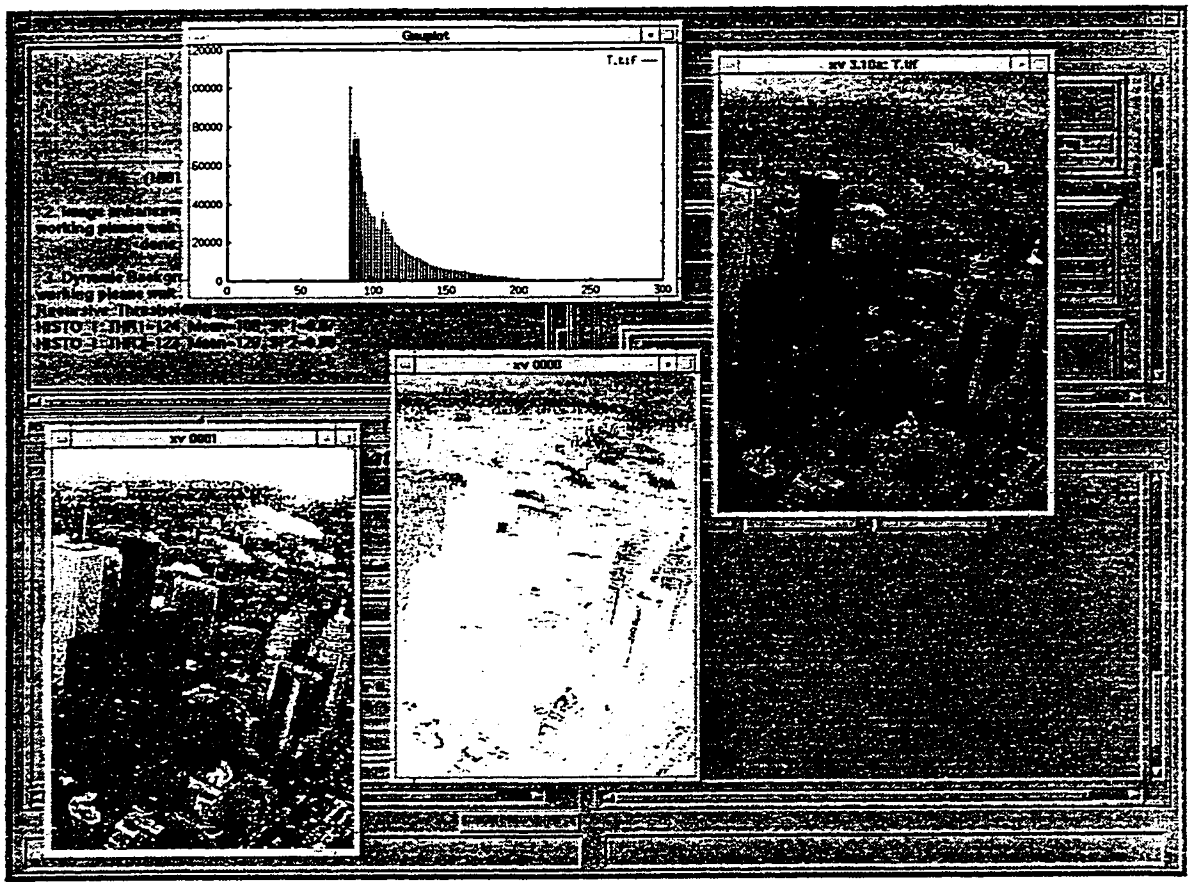

Figure 3.4: Image segmentation in the case of vague foreground-background relationship. The threshold value is 123. The background corresponds to the brightest object whose pixel values are greater than 123. The foreground corresponds to the darker object whose pixel values are less or equal to 123.

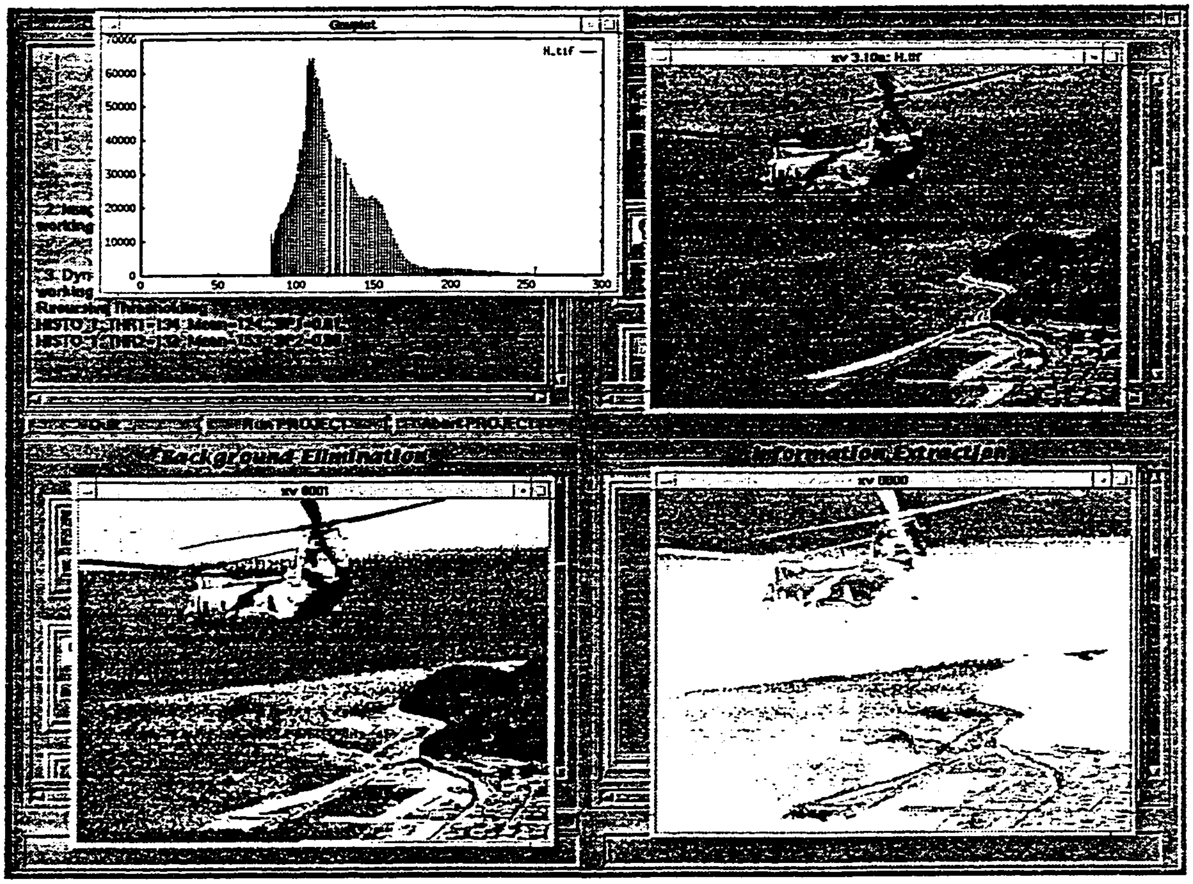

Figure 3.5: Image segmentation in the case of a vague foreground-background relationship. The threshold value is 133. The background corresponds to the brightest object whose pixel values are greater than 133. The foreground corresponds to the darker object whose pixel values are less or equal to 133.

3.3. For images where there is no clear foreground-background relationship, global thresholding methods do not produce good results as required. Figures 3.4 and 3.5 illustrate this point. In these figures, the histograms show that the distribution of pixel values of the corresponding images have very high peaks and no deep valleys between different objects. If we adopt the corresponding threshold value, as illustrated, the resulting thresholded objects will lose some of their pixels. Fortunately, this last case does not happen so often in cheque images.

For the segmentation of more complex images as in Figure 3.11 (c), however, it is necessary to resort to multilevel thresholding selection techniques. Generally, it is not a simple matter to determine multilevel threshold values. In images with multiple objects there are several difficulties for multilevel threshold selection that are associated with the gray-level distributions, small objects, and object overlapping. To overcome these difficulties, several techniques have been proposed. Baukharouba *et al.* [16] first determined a distribution rational function. Then, the multithreshold values are defined as the zeros of a curvature function derived from the distribution function. Many multithresholding approaches are based on edge matching and classification. The methods of Wang and Haralick [228], Hertz and Schafer [79], Kohler [121], and Spann and Wilson [193] belong to this category. These methods are applicable to images with good edges. Additionally, they are not based on histogram information for the determination of the threshold values. As a first step, in all these methods, the pixels of the initial image are first classified as edge and no edge pixels by using an edge extraction algorithm. Consequently, for the extraction of the best thresholds, computationally expensive recursive procedures are used. During each iteration, the threshold values are modified in order to satisfy some characteristics.

Spann and Wilson [193] proposed a hybrid multithreshold selection method, which is based on statistical and spatial information. Specifically, the method is a combination of a quad-tree smoothing technique, a local centroid clustering algorithm, and a boundary estimation approach. This method is applicable under some conditions, such as requiring that the histogram consists of only Gaussian distributions.

Another category of multithresholding selection approaches involves methods that are based on image histogram only. The methods of Reddi *et al.* [175], Kapur *et al.* [98], and Carlotto [21] belong to this category. The method of Reddi *et al.* is very fast and it is a version extended to multithresholding of the global threshold method of Otsu [161], which, according to [183], is probably the most powerful method for global thresholding when two threshold values are required for a given image. However, it is very important to observe that when more than two threshold values are required the method needs to be extended because it is constrained by the number of thresholds required and the previous knowledge about the number of objects in a given image. A second interesting and effective multithresholding approach is the method of Kapur *et al.*, which is based on the maximum entropy criterion. An interesting approach for multithresholding has been proposed by Carlotto. It determines the threshold value by processing information derived from the changes of zero-crossing in the second derivative. This method gives good results only for the histograms that satisfy the basic hypothesis that they consist of only univariate normal distribution.

Therefore, background reduction and elimination of an image entails the division or the separation of an image into regions of similar attributes. For background reduction and background elimination, the number of the objects that belong to the background and their intensity play an important role in selecting good thresholds.

In fact, many cheque images can be characterized as containing some objects of interest of reasonable uniform brightness placed against a background of different levels of brightness as illustrated in Figure 3.6. For such images, luminance is a distinguishing feature that can be utilized to segment the object of interest from its background. If an object of interest is white against a black background, or vice versa, it is a trivial task to set a mid-gray threshold to segment the object from the background. Difficult problems occur, however, when the observed image is subject to noise, and when both the object and the background assume some broad range of gray scales. As depicted in Figure 3.11 (a), another common difficulty is that the background may not be uniform as illustrated in Figure 3.6.

In the following sections, we will address the problem of eliminating such backgrounds using Otsu's [161] approach, show its limitations, and then present the re-

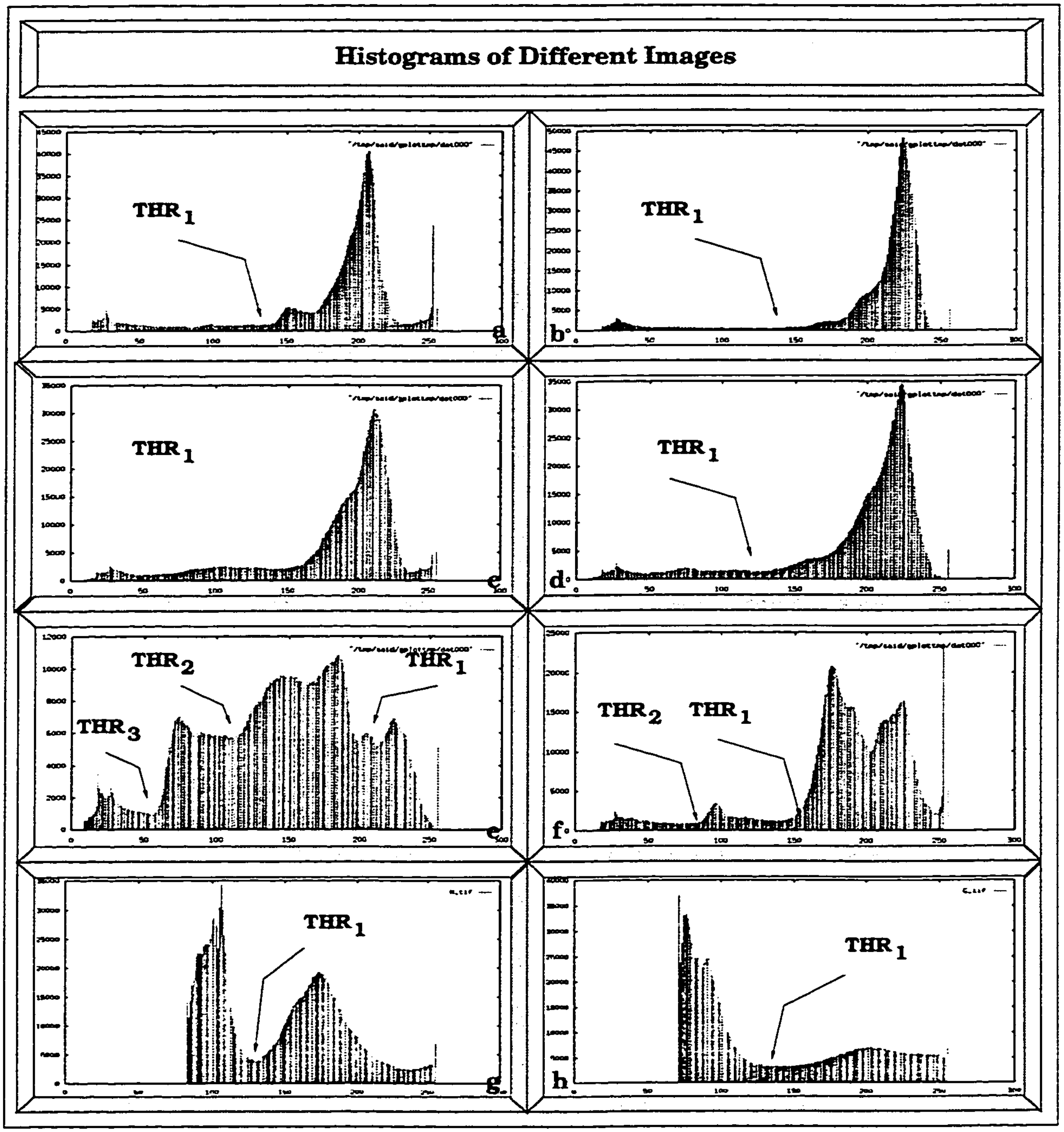

Figure 3.6: Histograms of different images with single or multiple objects.

cursive thresholding approach which is an extension to [161] and is not constrained by the number of objects in a given image as opposed to the methods discussed in the previous paragraphs.

3.3 Otsu's Approach

Let N be the set of natural numbers, (i, j) be the spatial coordinates of a digitized image, and $G = \{0, 1, \ldots, l-1\}$ be a set of positive integers representing the gray-levels[1]. Then, an image function can be defined as the mapping

$$f : N \times N :\rightarrow G$$

where $f(i, j)$ is the gray-level of a pixel whose coordinates are (i, j), $0 \leq i \leq h$ and $0 \leq j \leq w$, and h and w are the height and width of a digitized image f respectively. Also, by convention, the gray-level 0 is the darkest and the gray-level $l-1$ is the brightest.

Let $t \in G$ be a threshold and $B = \{b_0, b_1\}$ be a pair of binary gray-levels such that $b_0, b_1 \in G$. The result of thresholding an image $f(x, y)$ at gray-level t is a binary image function

$$f_B : N \times N :\rightarrow B,$$

such that

$$f_B(i, j) = \begin{cases} b_0 & \text{if } f(i, j) < t. \\ b_1 & \text{if } f(i, j) \geq t. \end{cases}$$

In general a thresholding method is one that determines the optimal value t^* of t based on a certain criterion.

Otsu's method, as proposed in [161], is based on discriminant analysis. The threshold operation is regarded as the partitioning of the pixels of an image into two classes C_0 and C_1 (e.g., objects and background) at gray-level t, i.e., $C_0 = \{0, 1, \ldots, t\}$ and $C_1 = \{t+1, t+2, \ldots, l-1\}$.

[1] In our model we assume $l = 256$.

In order to evaluate the "goodness" of the threshold (at level k) the discriminant criterion measures (or measures of class separability) used in the discriminant analysis [57] and defined in Equation 3.1. In this case, the problem is reduced to an optimization problem to search for a threshold t that maximizes one of the object functions (the criterion measures) of Equation 3.1.

This standpoint is motivated by a conjecture that well thresholded classes would be separated in gray-levels, and conversely, a threshold giving the best separation of classes in gray-levels would be the best threshold.

As stated in [161], let σ_W^2, σ_B^2, and σ_T^2 be the within-class variance, between-class variance, and the total variance, respectively. An optimal threshold can be determined by maximizing one of the following (equivalent) criterion functions with respect to t:

$$\lambda = \frac{\sigma_B^2}{\sigma_W^2}, \quad k = \frac{\sigma_T^2}{\sigma_W^2}, \quad \eta = \frac{\sigma_B^2}{\sigma_T^2}. \tag{3.1}$$

The discriminant criteria maximizing λ, k and η, respectively for t are, however, equivalent to one another: e.g., $k = \eta + 1$ and $\eta = \lambda/(\lambda + 1)$ in terms of λ, because the following basic relation always holds:

$$\sigma_W^2 + \sigma_B^2 = \sigma_T^2. \tag{3.2}$$

It is noticed that σ_W^2 and σ_B^2 are functions of threshold level t, but σ_T^2 is independent of t. It is also noted that σ_W^2 is based on the second-order statistics (class variance), while σ_B^2 is based on the first-order statistics (class means). Therefore, η is the simplest measure with respect to t. Thus η must be adopted as the criterion measure to evaluate the "goodness" (or separability) of the threshold at level t. Thus the optimal threshold t^* is defined as:

$$t^* = \underset{t \in G}{ArgMax}\ \eta, \tag{3.3}$$

where

$$\sigma_T^2 = \sum_{i=0}^{l-1} (i - \mu_T)^2 P_i, \quad \mu_T = \sum_{i=0}^{l-1} i P_i, \tag{3.4}$$

$$\sigma_B^2 = w_0 w_1 (\mu_1 \mu_0)^2, \quad w_0 = \sum_{i=0}^{t} P_i, \quad w_1 = 1 - w_0 \tag{3.5}$$

$$\mu_1 = \frac{\mu_T - \mu_t}{1 - \mu_0}, \quad \mu_0 = \frac{\mu_t}{w_0}, \quad \mu_t = \sum_{i=0}^{t} i P_i, \quad P_i = \frac{n_i}{n}. \tag{3.6}$$

n_i is the number of pixels with gray-level i and n is the total number of pixels in a given image defined as

$$n = \sum_{i=0}^{l-1} n_i. \tag{3.7}$$

P_i is the probability of occurrence of gray-level i defined as

$$P_i = \frac{n_i}{n}. \tag{3.8}$$

In digital images, the uniformity of objects plays a significant role in separating objects from the background. In fact, Otsu's method in thresholding gray-level images is good on the basis of uniformity measure between the two classes C_0 and C_1 that should be thresholded. Figure 3.7 illustrates the results after applying Otsu's method on two different images. Figure 3.7 (a) represents the original image, figure 3.7 (b) represents the histogram, Figure 3.7 (c) represents the final result (class C_1). and Figure 3.7 (d) represents the thresholded background (class C_0).

3.3.1 Analysis of Further Aspects

Otsu's method as proposed affords further means to analyze additional aspects other than the selection of the optimal threshold for a given image.

For a selected threshold t^* of a given image, the class probabilities w_0 and w_1 indicate the portions of the areas occupied by the classes C_0 and C_1 respectively in

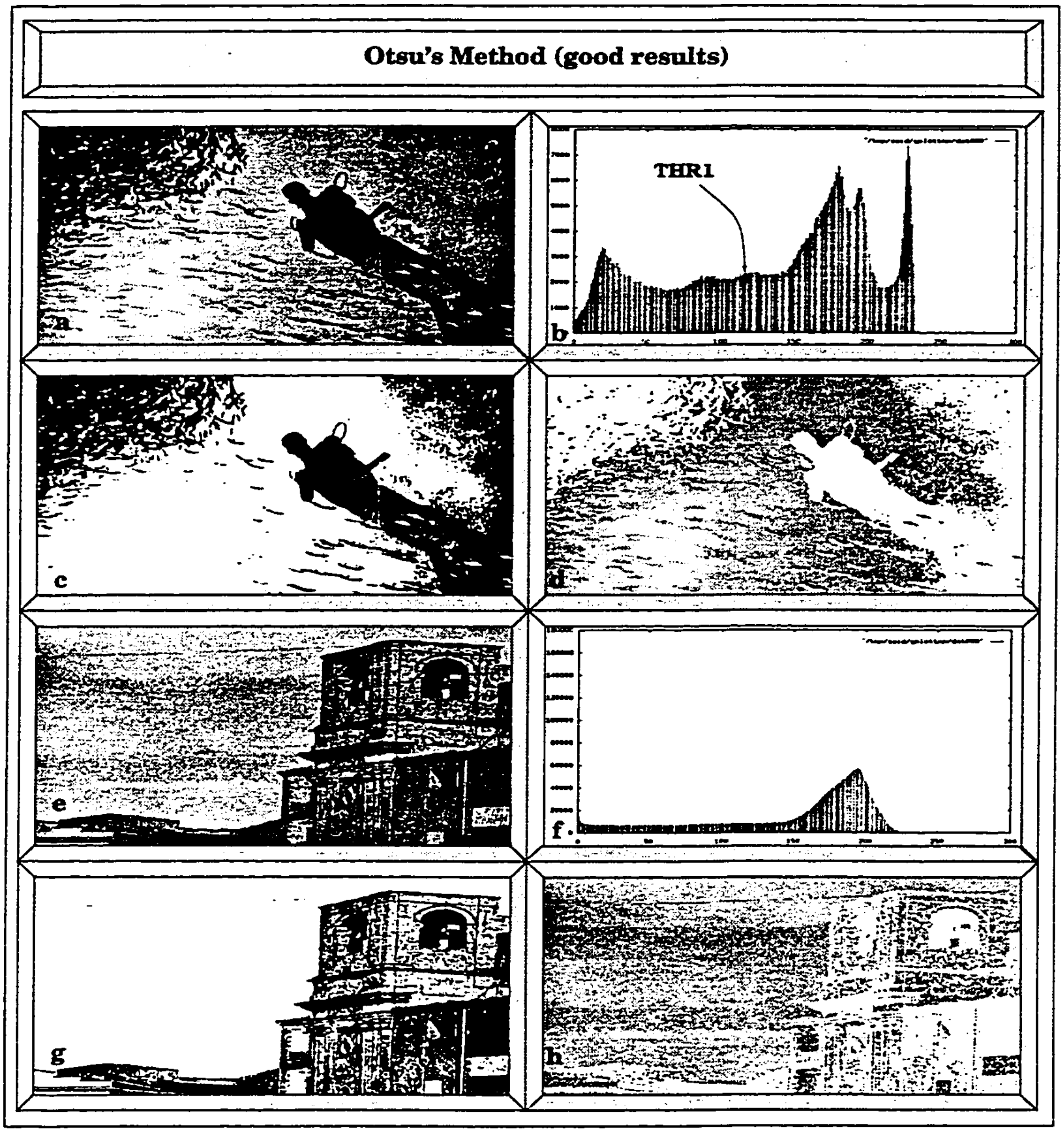

Figure 3.7: Image segmentation using Otsu's method.

the thresholded image. The class means μ_0 and μ_1 serve as estimates of the mean levels of the classes in the original gray-level image.

The maximum value of η, denoted by η^*, can be used as a measure to evaluate the separability of classes C_0 and C_1 in the original image or the bimodality of the histogram. This is a very significant measure because it is invariant under affine transformations of the gray-level scale. It is uniquely determined within the range

$$0 \leq \eta \leq 1.$$

The lower bound (zero) is obtained when, and only when, a given image has a single constant gray-level, and the upper bound (unity) is obtained when, and only when, two-valued images are given. This property is an important criterion that we will use in extending Otsu's approach.

3.3.2 Extension to Multithresholding

According to [161] the extension of the method to multithresholding problems is straightforward by virtue of discriminant criteria. For example in the case of three-thresholding, assume the two thresholds are: $1 \leq t_1 < t_2 \leq l$ for separating three classes, C_0 for $\{1, \ldots, t_1\}$, C_1 for $\{t_1 + 1, \ldots, t_2\}$, and C_2 for $\{t_2, \ldots, l\}$. The criterion measure σ_B^2 (or η) is then a function of two variables k_1 and k_2, and an optimal set of thresholds t_1^* and t_2^* is selected by maximizing σ_B^2:

$$\sigma_B^2(t_1^*, t_2^*) = \max_{1 \leq t_1 < t_2 \leq l} \sigma_B^2(t_1, t_2)$$

It should be noticed that the selected thresholds generally become less credible as the number of classes to be separated increases. This is because the criterion measure (σ_B^2), defined in one dimensional (gray-level) scale, may gradually lose its meaning as the number of classes increases. The expression of σ_B^2 and the maximization procedure also become more and more complicated. However, they are very simple for $M = 2$ and 3, which cover almost all practical applications, so that a special method to reduce the search processes is hardly needed. It should be emphasized that the parameters required in the present method for M-thresholds are $M - 1$

discrete thresholds themselves, while the parametric method, where the gray-level histogram is approximated by the sum of Gaussian distributions, require $3M - 1$ continuous parameters.

In this thesis, a new recursive method will be presented where the criterion σ_B^2 is always a function of one argument t. Moreover, the extension to multithresholding as proposed by [161] requires a previous knowledge of how many classes are there in a given image, however, the recursive method that will be developed in this research does not require any knowledge about the number of class objects in a given image.

3.4 General Scope of the Method

In segmenting several objects from a given image, the determination of multilevel threshold values should not be constrained by the number of objects in that particular image. Figure 3.8 illustrates the limitation of [161] when more than two objects exist in the image. For example, [175] is adequate when the image has exactly three or less objects. Moreover, [193] is based on statistical and spatial information requiring that the histogram consists of only Gaussian distributions.

Despite the fact that gray-level distributions, small objects, and object overlapping are some of the most complicated issues that create several challenging difficulties for multilevel threshold selection in images, a thresholding technique must be able to segment a digitized image into different objects which have similar properties. In this thesis, we will present an extension of Otsu's approach [161] as a general technique for image segmentation and limit the scope of the problem into document images. The new approach goes beyond segmenting only one bright object from an image to an approach that recursively segments the brightest object at each recursion leaving the darkest object in the given digitized image. The recursive method is developed without any constraints on the number of objects in the digitized image.

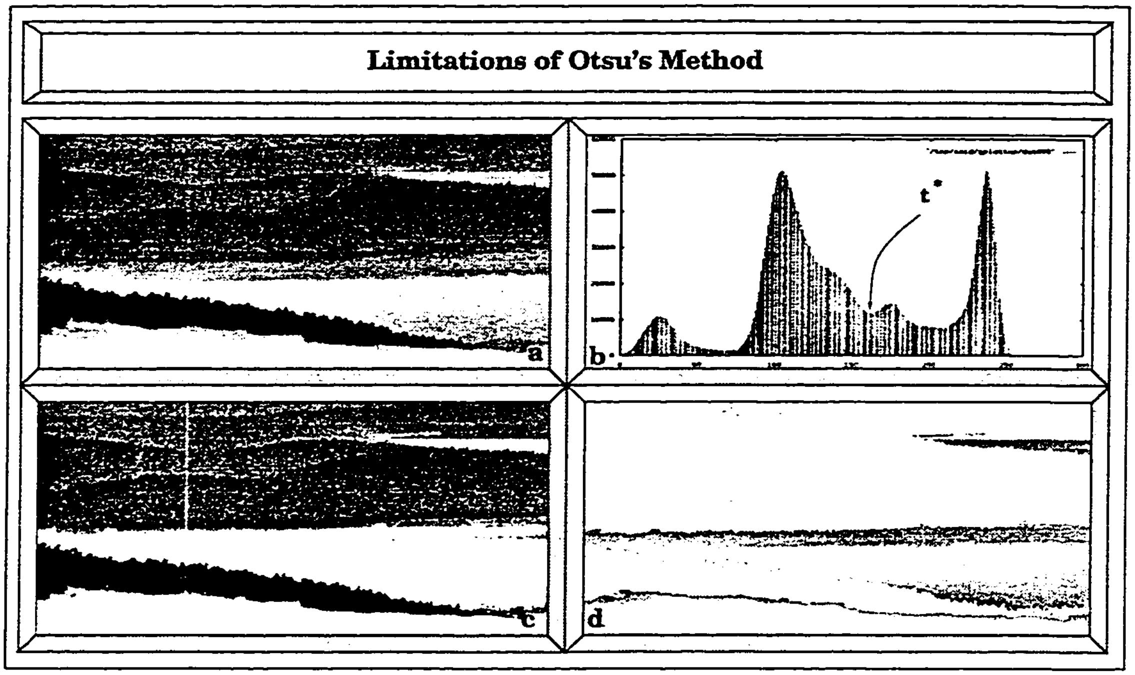

Figure 3.8: Image segmentation using Otsu's method.

3.5 Methodology

To apply thresholding recursively we proceed in the following way: according to [161], a global threshold value is evaluated for a given image and a new image with one thresholded object is produced. Now, the regions of the new image are considered as the given image and the process of histogramming, peak selection, and thresholding as defined in [161] is recursively repeated until no new peaks can be found or regions become too small. In this sense, an effective segmentation technique can be achieved that is not constrained by the number of objects in the original given image.

3.6 Image Enhancement

Now, given an image f, a smoothed image f_{filt} is generated whose intensity at every pixel (i, j) is obtained by averaging the intensity values of the pixels of f contained in a 3×3 neighbourhood of (i, j). Formally:

$$f_{filt}(i,j) = \frac{1}{K} \sum_{(n,m)\in S} f(n,m), \;\; \forall(i,j) \in W \times H \tag{3.9}$$

where K is the total number of points in S which is assumed as a 3×3 neighbourhood, and W and H are the width and height of the image f.

3.7 Recursive Thresholding

Now, given f_{filt}, thresholding as in [161] is applied recursively. Initially, a gray-level histogram is evaluated for the entire image f_{filt}. From this histogram, the method [161] automatically calculates the threshold THR_1 and the separability factor SP_1 for the entire image. SP_1 (η^* as in Section 3.3.1) is a number ranging from 0 to 1, indicates the likelihood of separating the class that has the lowest intensity in the image from the other classes in the same image. If the separability factor is less than S ($S = 95\%$ which is determined after training), which means that the two classes or objects to be separated are not 95% (or more) the same, the threshold value THR_1 will be used to separate the two classes or objects. This results in a new image f_1 from

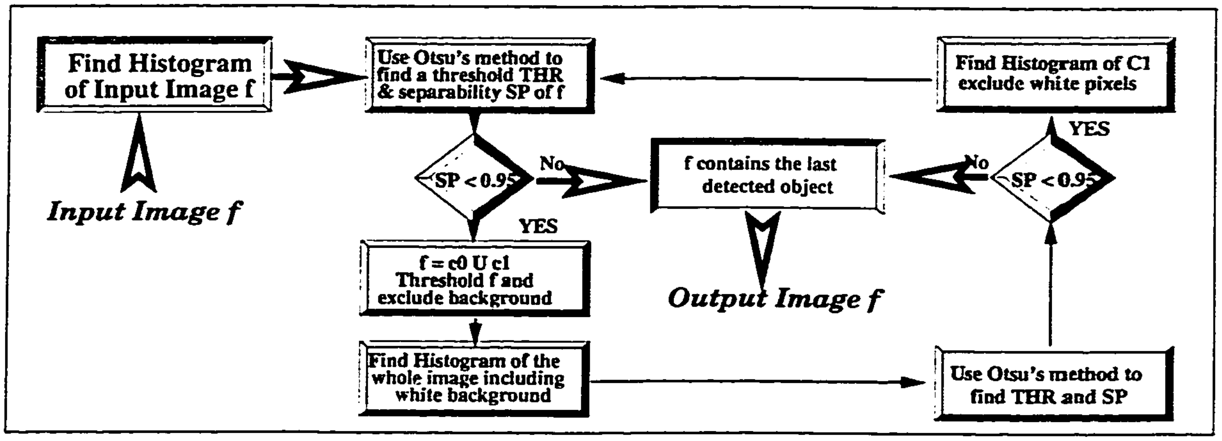

Figure 3.9: Diagram of Recursive Thresholding Technique.

which the object having the highest pixel values (denoted by C_0) has been segmented from the image f_{filt}.

At recursion t ($t \geq 1$), we will determine the gray-level histogram of the whole image including the white pixels of the background and use [161] to calculate the separability factor SP_t and the threshold value THR_t. If SP_t exceeds 95% the process will terminate since the image has only one object remaining against a white background. However, if SP_t does not exceed 95%, the recursive process is repeated to threshold an object C_t. Recursion is repeated p times until no more objects can be segmented. As a result, we obtain:

$$f_{filt} = C_0 \cup C_1 \cup \ldots \cup f_p, \tag{3.10}$$

where $f_p = f_{thr}$ is the object that remains after segmenting the different objects. Figure 3.9 presents the diagram and Figures 3.10 illustrates this approach.

3.8 Application to Cheque Images

The purpose of the new proposed technique is to develop a general recursive thresholding technique for gray-scale images and apply it to document processing and un-

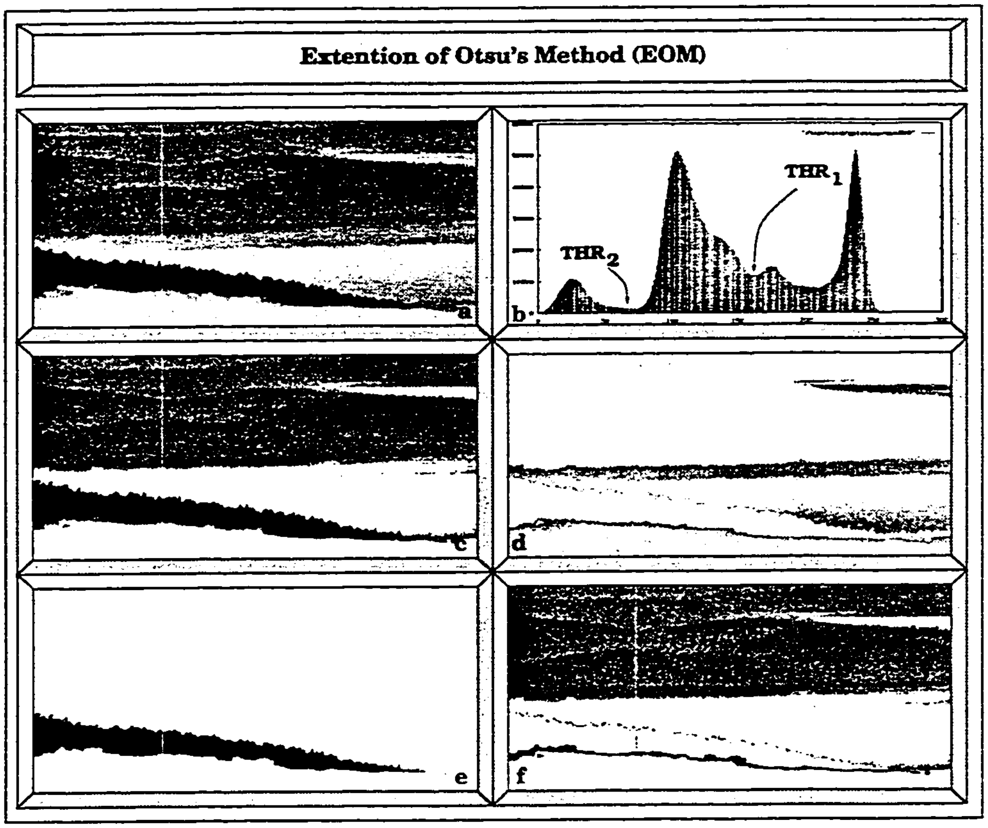

Figure 3.10: Image segmentation using the recursive method on an image with multiple objects. (a) the original image, (b) the histogram of the image in (a), (c) image after thresholding the brightest object whose pixel values are greater than the optimal threshold THR_1, (d) the brightest object, (e) the remaining darkest object whose pixel values are less than the optimal threshold THR_2, (f) the second brightest object whose pixel values are less than THR_1 and greater than THR_2.

derstanding. In fact, the new technique will facilitate the process of information extraction from document images.

As an application, 220 real life bank cheques were used to train the system and 505 different cheques were used to test its performance. The training set is needed to select an optimal value for S which is set at 95% in this experiment. For other applications, however, the value of S will differ depending on the intensity of the objects in the image. The optimal value of S should produce the best results even though performance evaluation of image processing techniques such as image segmentation is a non-trivial task by itself. In tackling this problem, a set of human visual criteria were defined where each criterion is given a weight. Several performance evaluation techniques for image segmentation [216] and thinning [126] follow the same approach. In this application, we considered several criteria such as the thickness, intensity, and connectivity of strokes.

In **TABLE I** we used the following convention that is clearly illustrated in Figures 3.12 - 3.16: **D:** *dark background or dark handwriting*, **D+:** *very dark background or very dark handwritten information*, **S:** *simple and homogeneous background with no figures*, **C:** *complex background with homogeneous figures considered as one object (refer to Section 4.5.1)*, **C+:** *complex background with non-homogeneous figures considered as multiple objects (refer to Section 4.5.1)*, **t:** *thin handwritten information*, **T:** *thick handwritten information*, **L:** *very light handwritten information*, **L+:** *light but not dark handwritten information*, As an example to illustrate this notation, the cheque image presented in Figure 4.2 is considered as having the attributes of D+, C+ for its background and {t, T} D+ for its foreground.

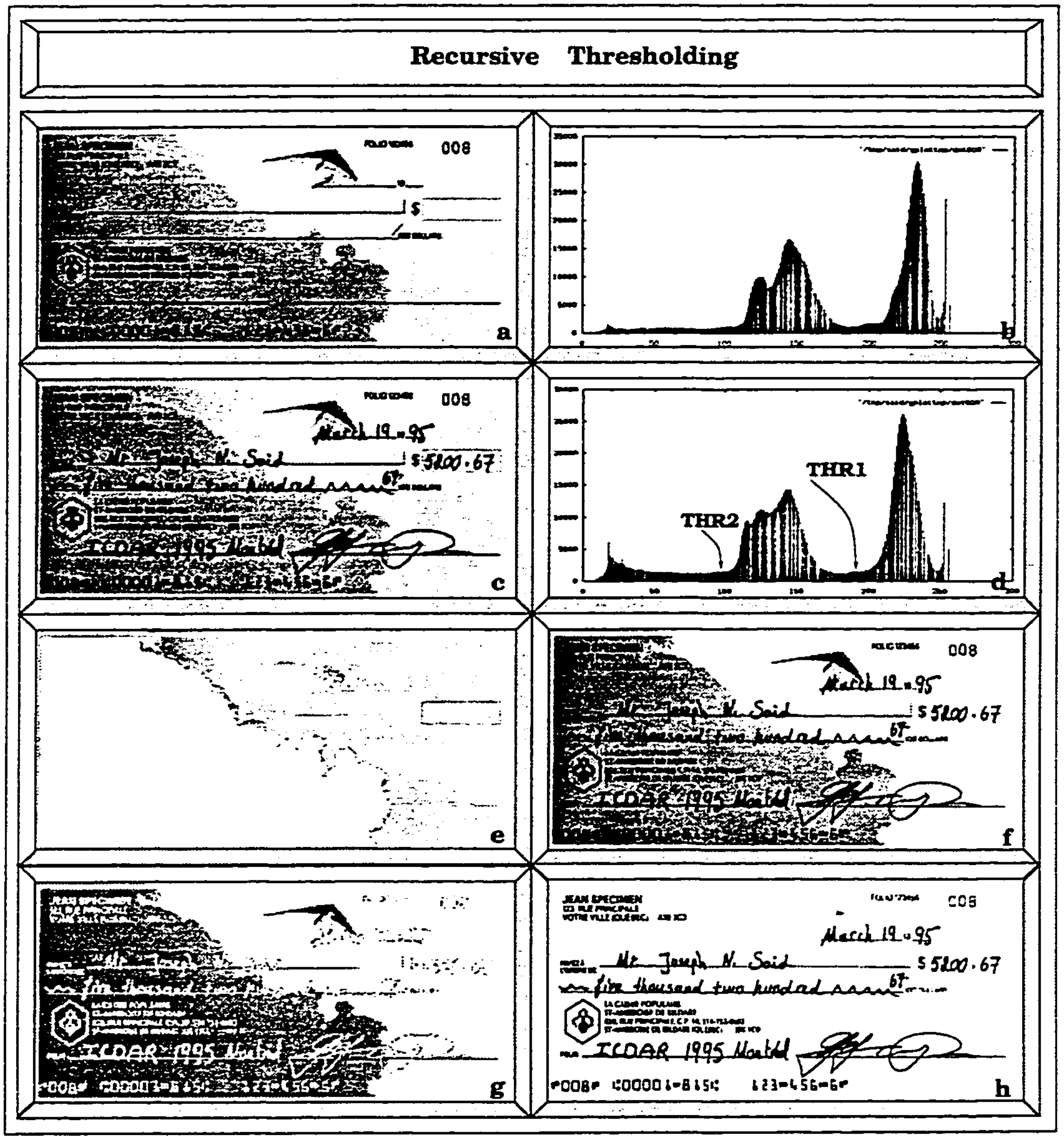

Figure 3.11: Image segmentation (recursive approach)

TABLE I						
# of Images	*size MB*	*Background/Foreground of a given cheque image f*	*Time in sec.* f_{filt},	f_{thr},	*Total*	*Performance Analysis %*
Training Set						
86	1,3	D S / t D+	14	2	16	99-100
61	1,4	D C / t D+	15	3	18	96-100
47	1,5	D C / {t, T} L+	16	5	21	85-100
26	1,5	D C+/ {t, T} D+	17	6	23	73-84
Testing Set (the training images were not used for testing)						
68	1,4	D S / {t, T} D+	15	2	17	99-100
74	1,5	D S / {t, T} L+	17	2	19	96-100
58	1,4	D S / {t, T} L	14	2	16	85-100
33	1,5	D+ S / {t, T} {D+, L+, L }	16	2	18	73-84
47	1,4	D C / {t, T} D+	15	5	20	98-100
56	1,4	D C / {t, T} L+	14	5	19	93-100
31	1,5	D C / {t, T} L	16	5	21	81-100
21	1,4	D+ C / {t, T} {D+, L+, L }	15	5	20	59-81
48	1,6	D C+ / {t, T} D+	18	5	23	98-100
35	1,6	D C+ / {t, T} L+	17	5	22	90-100
20	1,4	D C+ / {t, T} L	14	5	19	0
14	1,5	D+ C+ / {t, T} {D+, L+, L }	16	7	13	0

TABLE I indicates that the training and testing cheques have good and satisfactory results when the target object required is the darkest in the given image. However, when the target object in a given image is not the darkest object, recursive segmentation will not properly segment the target object. The last two entries in **TABLE I** illustrate this point.

The system is developed using a Sun Sparc 330 (Sparc 1), with 24M of memory and 25MHz. An HP scanner is employed to capture the image of cheques. The resolution of digitization used in the experiment can vary between 200-300 DPI.

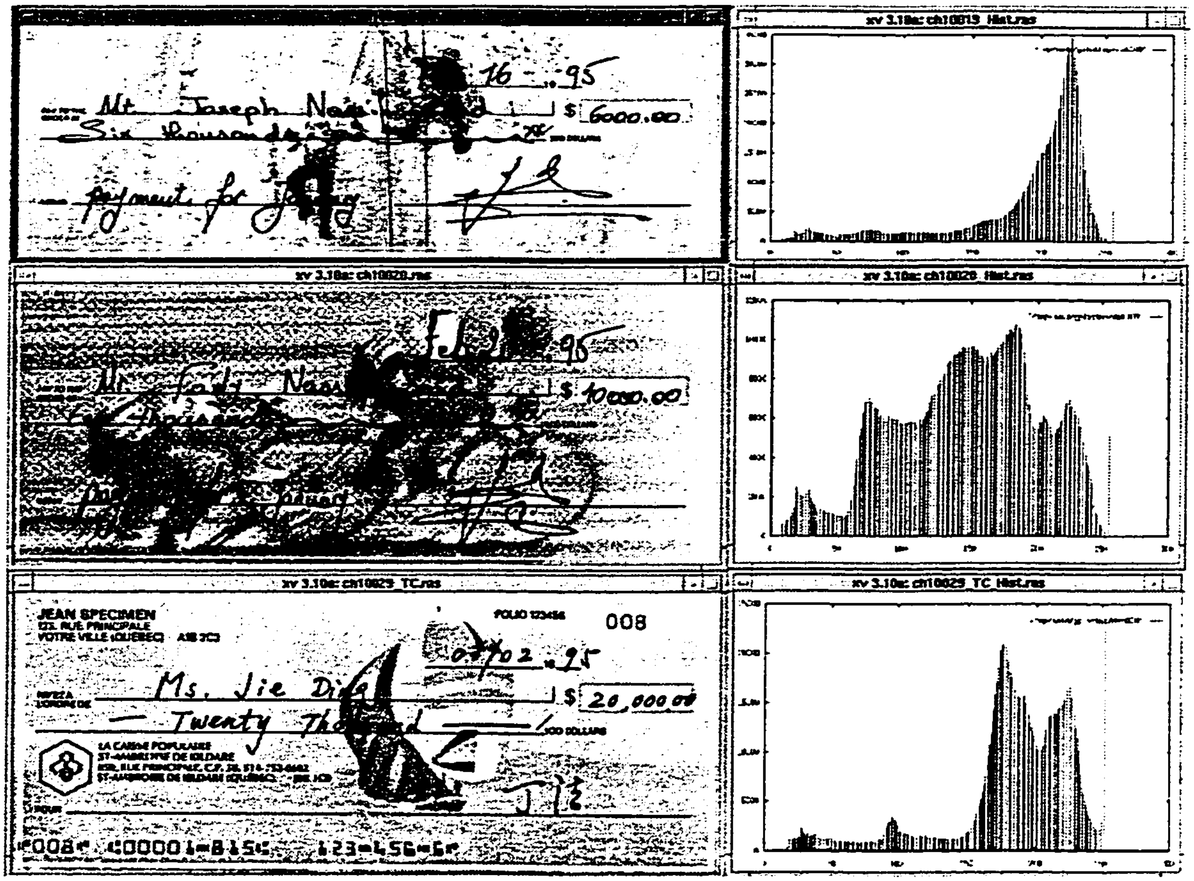

Figure 3.12: Cheque images considered as having a complex background (C+) with multiple objects that correspond to multiple peaks in their corresponding histograms. Some objects in the background images have very dark (D+) intensities that are very close to the intensity of the handwritten information which corresponds to the left most peak in the histogram. Handwritten information is considered to be thick (T) and very dark (D+).

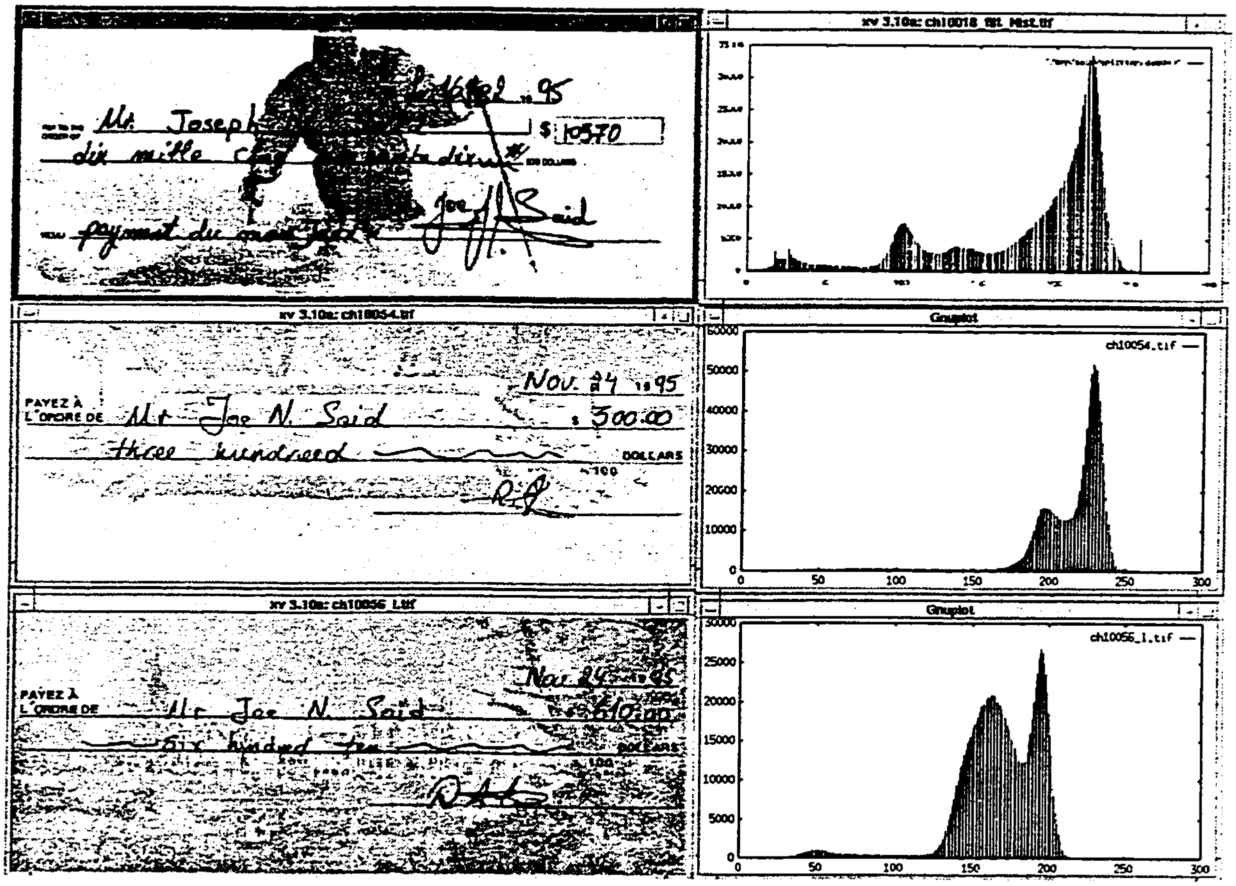

Figure 3.13: Cheque images considered as having a complex background (C+) with multiple objects that correspond to multiple peaks in their corresponding histograms. Some objects in the background images have very dark (D+) intensities (first cheque image from top). The other two cheque images have dark (D) intensities. Handwritten information is considered to be thick (T) and very dark (D+) in the top cheque image, however, it is considered to be thin (t) and dark (D) in the other two cheque images.

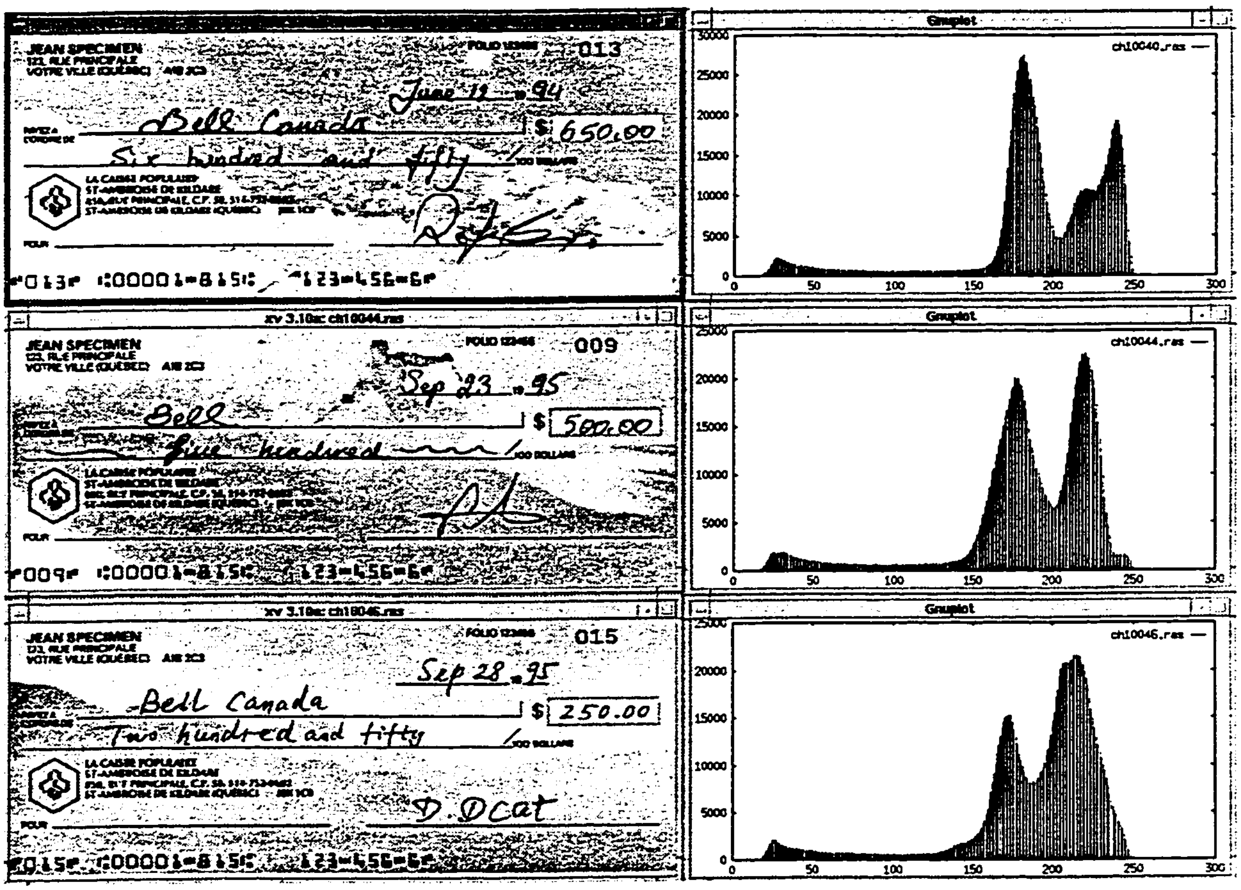

Figure 3.14: Cheque images considered as having a complex background (C+) with multiple objects that correspond to multiple peaks in their corresponding histograms. All objects in the background cheque images have dark (D) intensities. Handwritten information is considered to be thick (T) or thin (t) and very dark (D+) in all cheque images.

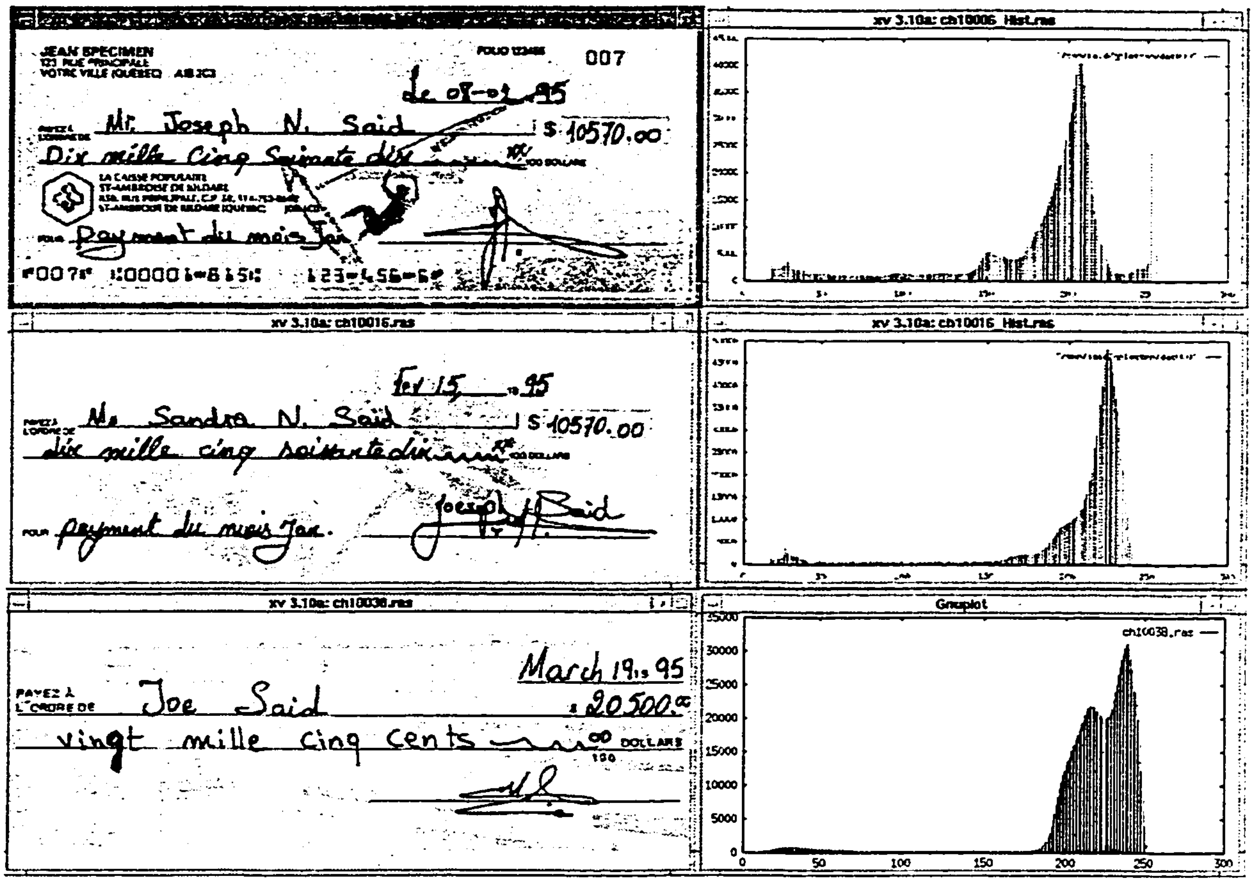

Figure 3.15: Cheque images considered as having a complex background (C) with single object that correspond to the same peak in their corresponding histograms. The first cheque image has a small peak that is considered as part of the highest nearest peak. All objects in the background cheque images have dark (D) intensities. Handwritten information is considered to be thick (T) and very dark (D+) in all cheque images.

Figure 3.16: Cheque images considered as having a simple homogeneous background (S) with single object that correspond to the same peak in their corresponding histograms. All objects in the background cheque images have dark (D) intensities. Handwritten information is considered to be thick (T) and very dark (D+) in the top two cheque images and (t) and light but dark (L+) in the last cheque image.

Chapter 4

A Basic Model for Visual Data Processing of Business Forms and Bank Cheques

A Top-down approach.

4.1 Introduction

In reality, it is very difficult to develop a general system that can process all kinds of binarized documents including business forms. As a first step, it is possible to develop a specific system that treats a specific type of documents. After carefully studying the major characteristics of different types of documents, the specific properties of business documents have been identified and analyzed. According to these properties, we have taken the approach of building a simple system for processing a class of gray-scale business documents, such as bank cheques, instead of building a complex system for processing all sorts of documents.

Cheques are commonly used in our daily life for the remittance of payments. Their use simplifies the remittance process on one hand, but requires additional processing at banks and post offices on the other hand. Typically, cheque processing at a bank consists of first reading information about the sender, the receiver and the amount of money, and then entering it into a database for storage or sending it to another bank for reconciliation. Although electronic information processing has become important in banks and business institutions nowadays, these processing steps are only partly automated. In this Chapter, we present a formal model for processing the visual data of bank cheques, and describe in detail the major aspects of this model.

The formal model aims at scanning the gray-scale images of bank cheques, enhance the quality of the scanned image, remove unnecessary information and extract the targeted data such as the date, courtesy amount, and legal amount. Of course, the challenge is to extract the desired information without introducing distortion into it. The extracted information will be presented later to various recognition systems such as [181] for recognition. The result is an automatic system that can process the images of cheque.

4.2 Methodology

The formal model is based on a new hybrid-based approach namely the baselines. In fact, to segment and extract handwritten data from bank cheques, knowledge rules and baselines will play important roles. The architectural design as well as the major components of the system will be discussed in full details. Moreover, the significant use of the morphological and topological processing on gray-scale images will be used as a major aspect to restore the information lost after the elimination of the background and the baselines from the gray-scale cheques.

4.3 The Formal Model

As mentioned earlier, we will consider gray-scale images because they provide us with more flexibility in extracting distinctive features from information written on bank cheques, as compared to the binary images. In fact, processing binary images requires less memory and is much simpler; however, binary images can complicate very much the process of extracting or segmenting handwritten data from them. Nevertheless. gray-scale bank cheques are not easy to manipulate. This is true since a bank cheque could have a background that has the same or similar luminance property as the handwritten data. Moreover, the background could have different luminance levels due to different graphical layouts.

To cope with the problem mentioned above we will divide our approach into various steps that lead us to the final results of extracting the handwritten data from these bank cheques. According to Figure 4.1, the process of extracting handwritten data from bank cheques is divided into the following steps: Image enhancement (average filtering), image segmentation (background elimination), object detection (detection and elimination of baselines), mathematical morphological processing (closing operation), image enhancement (binarization and median filtering) topological processing (restoration of lost information), segmentation of required information, and finally character recognition [181].

To increase the recognition rate, we should minimize any possible damage done to the data. In other words, we want to eliminate the unneeded information (background first and then the baselines) from the bank cheques while we attempt to preserve as much as possible the topological properties of the needed information (handwritten data) for recognition.

Definition 7 The geometrical structure of a given cheque image f is defined as a finite set

$$F = \{B_p, L_p, \alpha_f, \alpha_p\}$$

where

$$B_p = \{B_{p_1}, B_{p_2}, \ldots, B_{p'_n}\},$$

$$L_p = \{L_{p_1}, L_{p_2}, \ldots, L_{p_n}\},$$

$$\alpha_f = \{\alpha_{f_1}, \alpha_{f_2}, \ldots, \alpha_{f_m}\},$$

and

$$\alpha_p = \{\alpha_{p_1}, \alpha_{p_2}, \ldots, \alpha_{p_k}\},$$

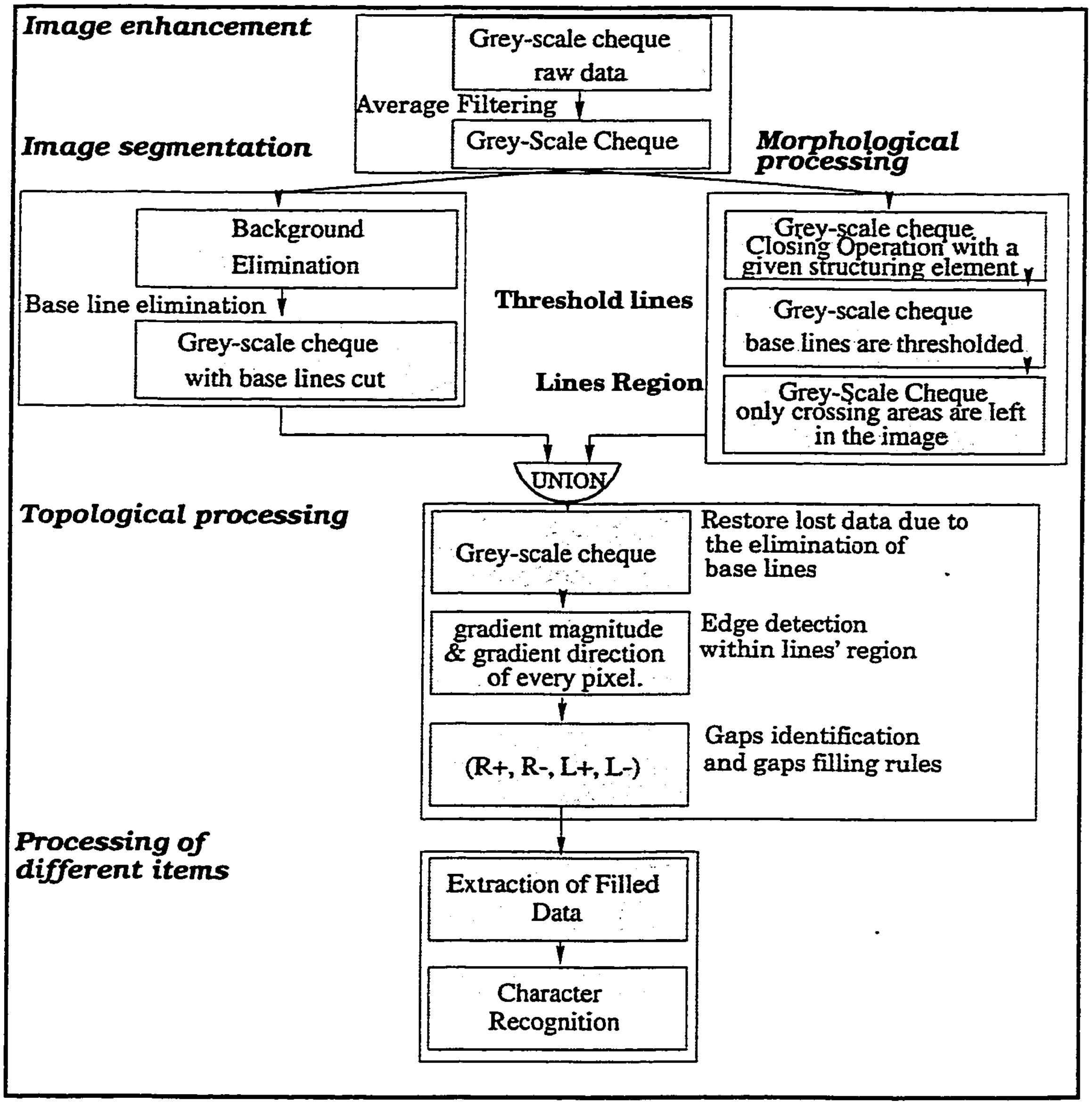

Figure 4.1: Major steps in system's design for bank cheque processing.

Figure 4.2: A sample output process.

where B_p is a set of preprinted background objects (figures and images), L_p stands for a finite set of pre-printed lines, α_f indicates a finite set of handwritten data. and α_p, denotes a set of pre-printed data.

Definition 8 The background B of a given cheque image f is defined as a subset of its geometrical structure F. Formally:

$$B = \{B_p, L_p\}$$

Definition 9 The foreground O of a given cheque image f is defined as a subset of its geometrical structure F. Formally:

$$O = \{\alpha_f, \alpha_p\}$$

Figure 4.3 illustrates a sample geometrical structure of a Canadian bank cheque.

Baselines $L_s = \{L_{s_1}, L_{s_2}, \dots, L_{s_q}\}$ are enclosed in the set of pre-printed lines $L_s \subseteq L_p$. For each baseline $L_{s_k} \in L_s$ we associate the upper left coordinates $P_u(i_u, j_u)$ and the lower right coordinates $P_b(i_b, j_b)$. In other words, L_{s_k} is defined as the set

$$L_{s_k}(P_u, P_b) = \{(i, j) \mid i_u \leq i \leq i_b, j_u \leq j \leq j_b\}.$$

A finite set, D, of cheques is a subset of

$$D_{B_p} \times D_{L_p} \times D_{\alpha_f} \times D_{\alpha_p},$$

where D_{B_p}, D_{L_p}, D_{α_f}, D_{α_p}, and D_{α_o} are the domains of $B_p, L_p, \alpha_f, \alpha_p, \alpha_o$ respectively. Therefore, a cheque image f, $f \in D$, has a finite set of pre-printed background objects B_p, a finite set of pre-printed lines L_s, a finite set of handwritten data α_f, and a finite set of pre-printed data α_p. Our purpose in this Chapter is to locate, identify, and separate these finite sets for further processing.

4.4 Image Enhancement

Given an image f, produce f_{filt} according to Equation 3.9. Such smoothing is required in order to produce a better histogram that will be used to generate more accurate thresholding values. Figure 4.4 illustrates an example.

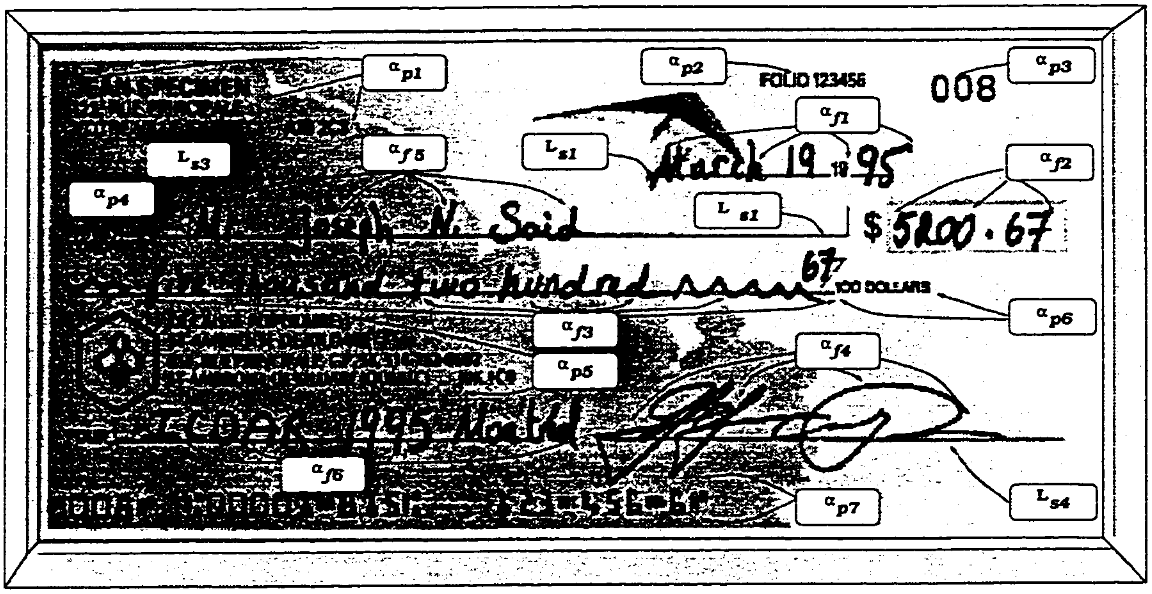

Figure 4.3: Structure of a bank cheque $F = \{B_p, L_p, \alpha_f, \alpha_p\}$.

4.5 Image Segmentation and Background Elimination

Background reduction and elimination of an image entails the division or the separation of an image into regions of similar attributes. For background reduction and background elimination we will present a natural extension to Otsu's method [161] since Otsu's method is considered by [183] as one of the best techniques for global thresholding. For baseline elimination, we will introduce a newly developed morphology-based approach that allows us to extract baselines from images of bank cheques. Extracting baselines from bank cheques, however, will introduce another problem, i.e., the gaps that result from the removal of the baselines. To cope with this difficulty, a topology-based approach is introduced which we will formally introduce in Section 4.6.

Many cheque images can be characterized as containing some objects of interest of reasonable uniform brightness placed against a background of different levels of brightness. For such images, luminance is a distinguishing feature that can be utilized to segment the object of interest from its background. If an object of interest is white against a black background, or vice versa, it is a trivial task to set a mid-gray threshold

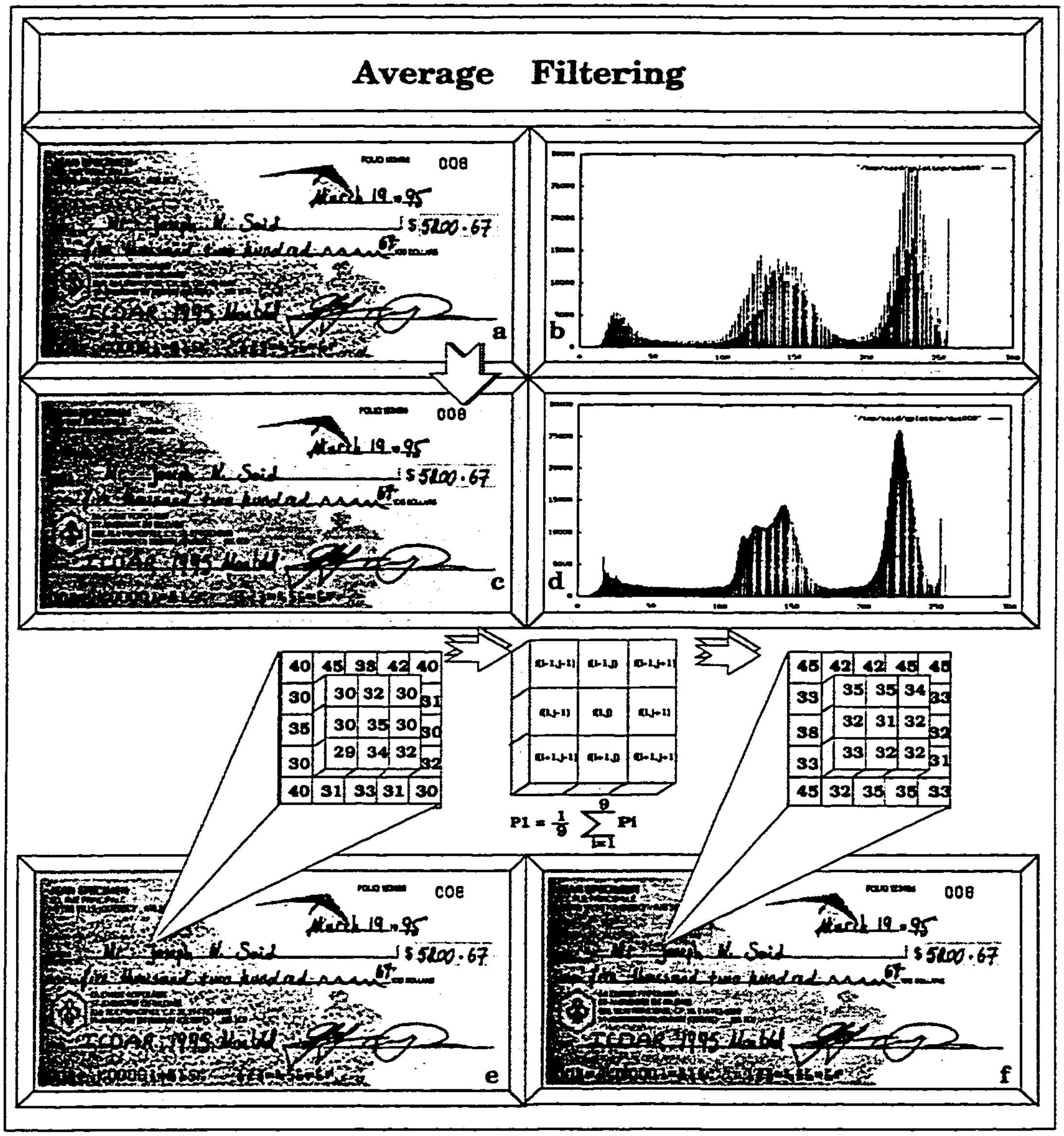

Figure 4.4: Image enhancement using a 3X3 window.

to segment the object from the background. Practical problems occur, however, when the observed image is subject to noise, and when both the object and the background assume some broad range of gray scales. As depicted in Figure 4.3, another common difficulty is that the background may not be uniform.

4.5.1 Image Segmentation Using Global Recursive Thresholding

In the histogram method for peak selection, the contributions of a small region may be masked by other larger ones. One solution is to apply thresholding recursively as Equation 3.10 suggests. The result of performing recursive thresholding on image f_{filt} according to Equation 3.10 is:

$$f_{filt} = C_0 \cup C_1 \cup \ldots \cup f_p,$$

where $f_p = f_{thr}$ is the image containing the handwritten information and the baselines that remains after segmenting the different objects that belong to the background of the cheque image.

4.5.2 Baseline Detection

Baselines correspond to pre-printed lines belonging to the geometrical structure of a financial document. They occur in bank cheques and other types of documents and serve as guidelines to help the writer or the typist to fill data in the proper locations.

After carefully analyzing the geometrical structure of the Canadian bank cheques, we found that they have a common general layout. By scanning any bank cheque from top to bottom, several knowledge rules can be derived, e.g.

- All baselines of interest extend into the area of the rightmost 1/4 of cheques.
- Although space between baselines as well as the position and style of pre-printed data vary in different bank cheques, all baselines are always located in the following order:

1. The first baseline is for the date.
2. The second baseline is for the courtesy amount.
3. The third baseline is for the legal amount.
4. The last baseline is for the signature.

By virtue of the first characteristic, the baselines can be easily extracted by the system. It is convenient to determine the different filled data in accordance with the second characteristic imposed on the different bank cheques.

The process of finding baselines in the image f_{thr} consists of three major stages: marking of segments, detecting of heads of baselines, and determining of baselines.

Marking segments is the operation of processing the image f_{thr} segment by segment, from top to bottom, and from right to left, looking for segments whose length exceeds a fixed threshold T. The marked segments are called candidate-segments denoted by the set S and defined as:

$$S = \{S_1, S_2, \ldots, S_i, \ldots, S_K\} \ (\forall i)(1 \leq i \leq K)(|S_i| \geq T)$$

.

Detecting the head of a single baseline is the process of connecting a series of candidate-segments. $S_l, S_{l+1}, \ldots, S_{l+n}$. The rightmost candidate-segment of the baseline is called the head of the baseline denoted as H_{S_k} ($H_{S_k} = S_{l+n}$). Thus, a cheque contains a set of heads defined as:

$$H_S = \{H_{S_1}, H_{S_2}, \ldots, H_{S_q}\}.$$

Any candidate-segment $S_{i'}$ has two parameters: start-position $P_s(i_s, j_s)$ and end-position $P_e(i_e, j_e)$, i.e., $S_{i'}(P_s, P_e)$. Once P_s and P_e have been detected, the corresponding segment $S_{i'}$ can be located. We define a candidate-segment $S_{i'}(P_s, P_e)$ as:

$$S_{i'}(P_s, P_e) = \{(i, j) \mid j_s \leq j \leq j_e\}.$$

All baselines of interest extend into the rightmost 1/4 area of the cheque. Suppose the width of a cheque is W, then the length from the left edge of the cheque to the end-position of the head of a baseline L_{s_k} must be greater than $\frac{3}{4}W$. It can be concluded that a candidate-segment is a head of a baseline, if and only if the following condition is satisfied:

$$H_{S_k} = \{S_l(P_s, P_e) \mid (S_l(P_s, P_e) \in S), (j_e \geq \frac{3}{4}W).$$

which is the necessary and sufficient condition for a candidate-segment to be the head of a baseline. From it the heads H_{S_k}, $k \in \{1, 2, \ldots, q\}$ can be determined.

Determining baselines is the process of finding the sequence $S_l, S_{l+1}, \ldots, S_{l+n}$, which constructs a baseline. Once the head S_{l+n} has been detected, a path algorithm can be used to search for the remaining segments of the baseline. In this algorithm, a search can start from the head S_{l+n}, from right to left. The segments $S_{l+(n-1)}, S_{l+(n-2)}, \ldots$ can be found in turn, until S_l and the baseline are determined. Furthermore, the coordinates of point P_s and ending point P_e can also be determined.

4.5.3 Elimination of Baselines

Given that the background has been completely eliminated from the gray-scale image and baselines are located, Figure 4.5 (c), elimination of baselines from the gray-scale image f_{thr} is performed as follows:

$$\forall L_{S_k} \in L_S \quad f_{lr}(i,j) = \begin{cases} 255 & \text{if } (i,j) \in L_{S_k} \\ f_{thr}(i,j) & otherwise \end{cases} \tag{4.1}$$

where $f_{lr}(i,j)$ represents the image after removal of all baselines and L_{S_k} represents a recorded baseline, i.e., $L_{S_k} \in L_S$. Unfortunately, as Figure 4.5 (b,e) illustrates, when the baselines are cut from the gray-scale image f_{thr} there is a loss in the handwritten information that intersects with the baselines before, as shown in Figures 4.5 (d) and Figure 4.2 (b). As a remedy to this problem, we will introduce a formal approach using morphological processing followed by topological processing to restore (fill the gaps) the lost information.

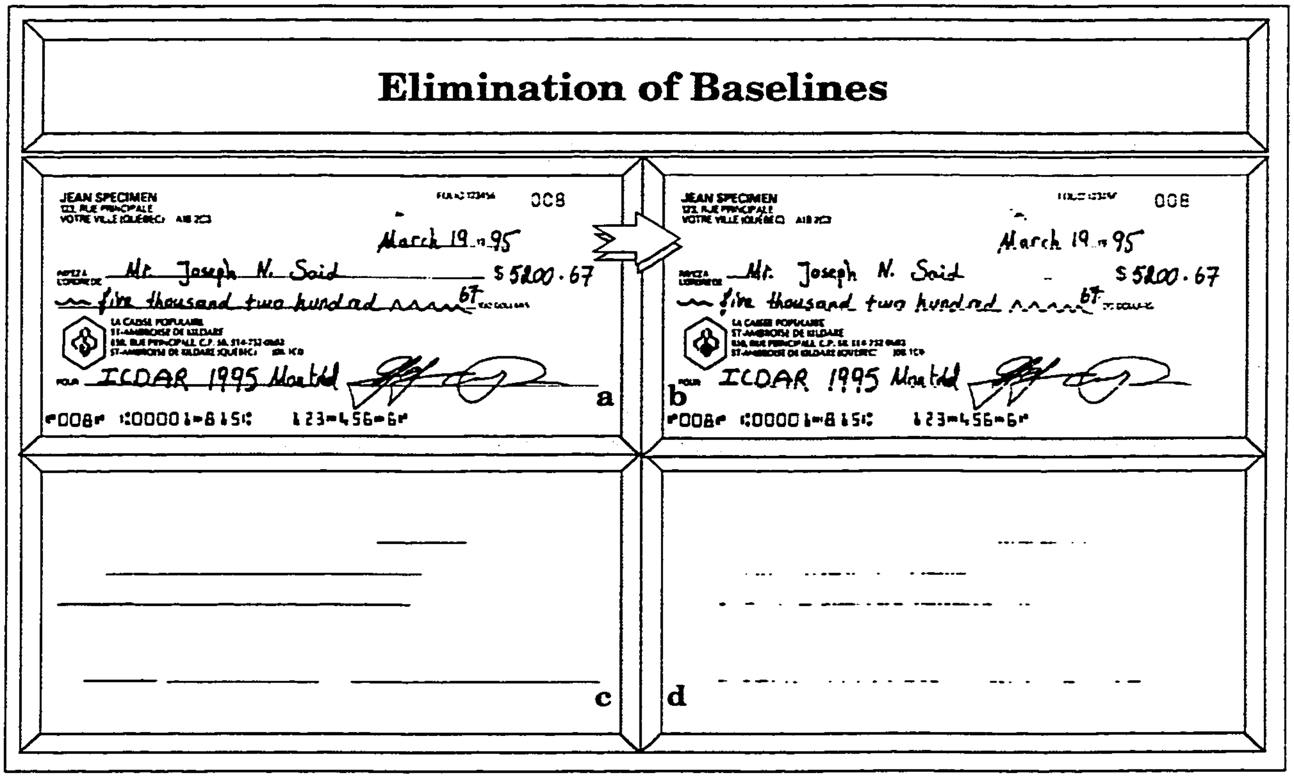

Figure 4.5: Object detection (elimination of baselines). (a) image with lines; (b) image with lines cut; (c) lines to be cut; (d) information that intersects with the lines that are cut.

4.6 Morphological Processing

Given f_{filt}, we will use morphological processing to restore the lost handwritten information that intersects with the eliminated baselines. To restore this lost information we will perform a morphological closing operation as in [14, 186] on f_{filt}, threshold the baselines from the morphological image, and extract the information within the line's region to be added to f_{lr}. In this section, we will present the morphological closing operation and its important property as in [14] before we deal with the other processes mentioned in this paragraph.

The gray-scale closing is defined as

$$f \bullet V = (f \oplus V) \ominus V \tag{4.2}$$

where

$$(f \ominus V)(i,j) = min\{f(i+k, j+l) \mid (k,l) \in V_{\Delta}\} \tag{4.3}$$

$$(f \oplus V)(i,j) = max\{f(i+k, j+l) \mid (k,l) \in V_{\Delta}\} \tag{4.4}$$

are the erosion and the dilation respectively of image f by the structuring element V, as illustrated in Figure 4.2 (c) and Figure 4.6.

Closing has an interesting property: it is an *idempotent* transform; once it has been applied, it is useless to apply it again. This is defined as:

$$f \bullet V \bullet V = f \bullet V. \tag{4.5}$$

According to [14], every nonlinear transformation results in some loss of information; but if a transformation is idempotent, the amount of information it would lose is self-controlled which is an important property to be considered in our case. The resulting image is named f_{clos}

4.7 Why Morphology?

As we pointed out in the previous sections, we aim to extract handwritten data from gray-scale bank cheques without damaging that data after the baselines are removed. Based on our observation, when the intensity of handwritten data is close to the intensity of the baselines, the situation becomes more difficult. Moreover, if the handwritten data and the baselines intersect, then the elimination of baselines and the preservation of the handwritten data becomes a non-trivial problem. *In fact, our purpose is to use the morphological closing operation to increase the difference in the intensity between a baseline and the handwritten data that intersects with that baseline. Technically, this means that we are propagating the intensity of the handwritten data as well as the intensity of the background over the intensity of the baselines. Figure 4.7 illustrates this point. As a result of this propagation, it is easy to threshold the baseline from the morphological image (cheque in our case) preserving the handwritten information that intersects with the thresholded baseline.* Figure 4.8 illustrates the results of performing a closing operation on a cheque image.

Without loss of generality, we should mention the most important role played by the size and shape of the structuring element in increasing the contrast between the baseline and the handwritten data. From an implementation point of view, a 3x3 structuring element (011, 011, 000) with 2 iterations is adopted in this context.

4.8 Restoring Lost Information

Knowing the positions and lengths of the baselines, we can now process the morphological image f_{clos}. This is achieved by thresholding baseline pixel values according to their mean M using Equation 4.6 as illustrated in Figure 4.8. and combine the results with the image f_{lr} as follows:

$$f_{clos}(i,j) = \begin{cases} f_{clos}(i,j) & \text{if } f_{clos}(i,j) \leq M \\ 255 & otherwise \end{cases} \tag{4.6}$$

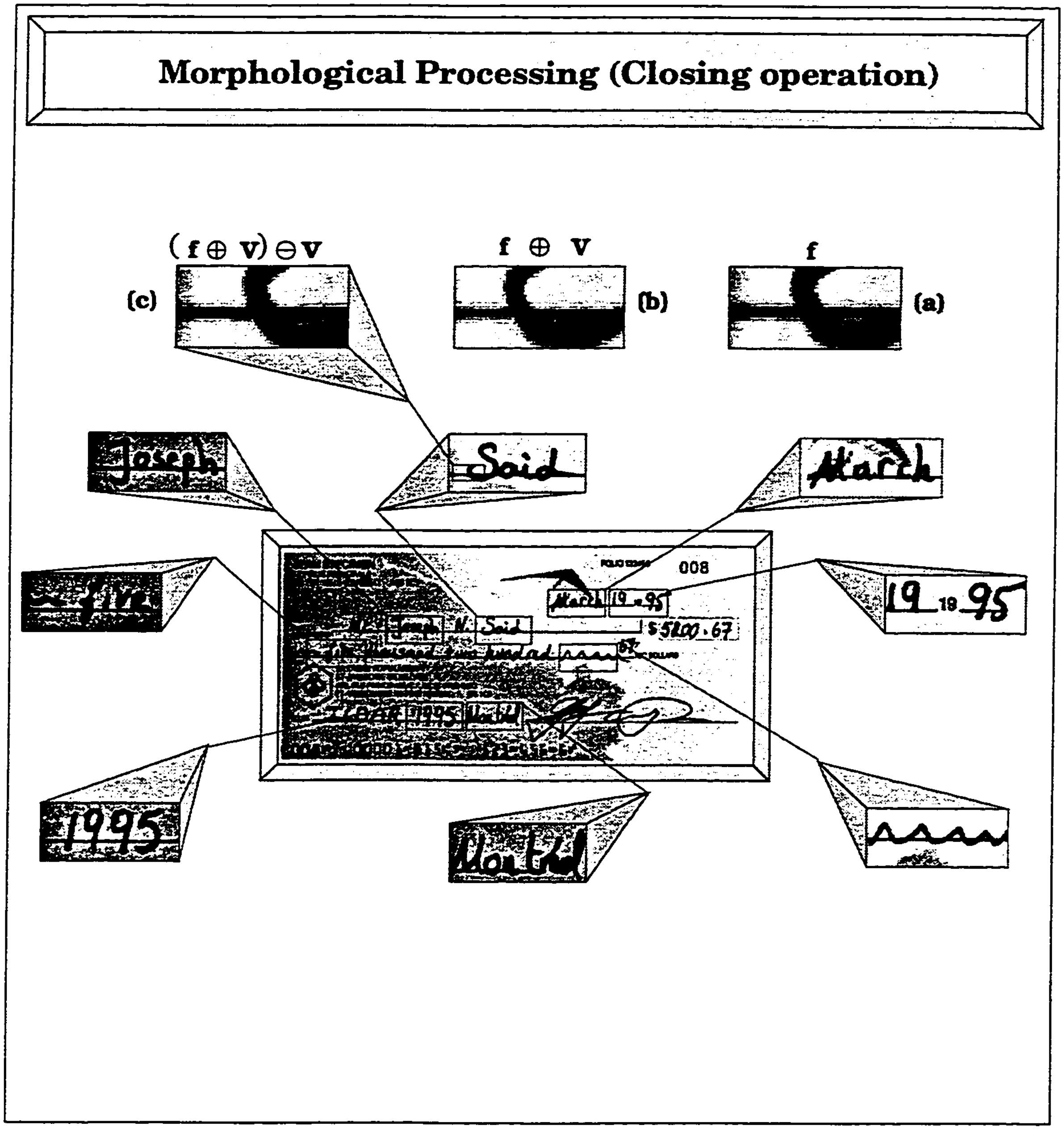

Figure 4.6: Morphological closing operation performed on image f_{filt} to produce f_{clos}. (a) illustrates a sample section of the original image and (b) and (c) illustrate the process.

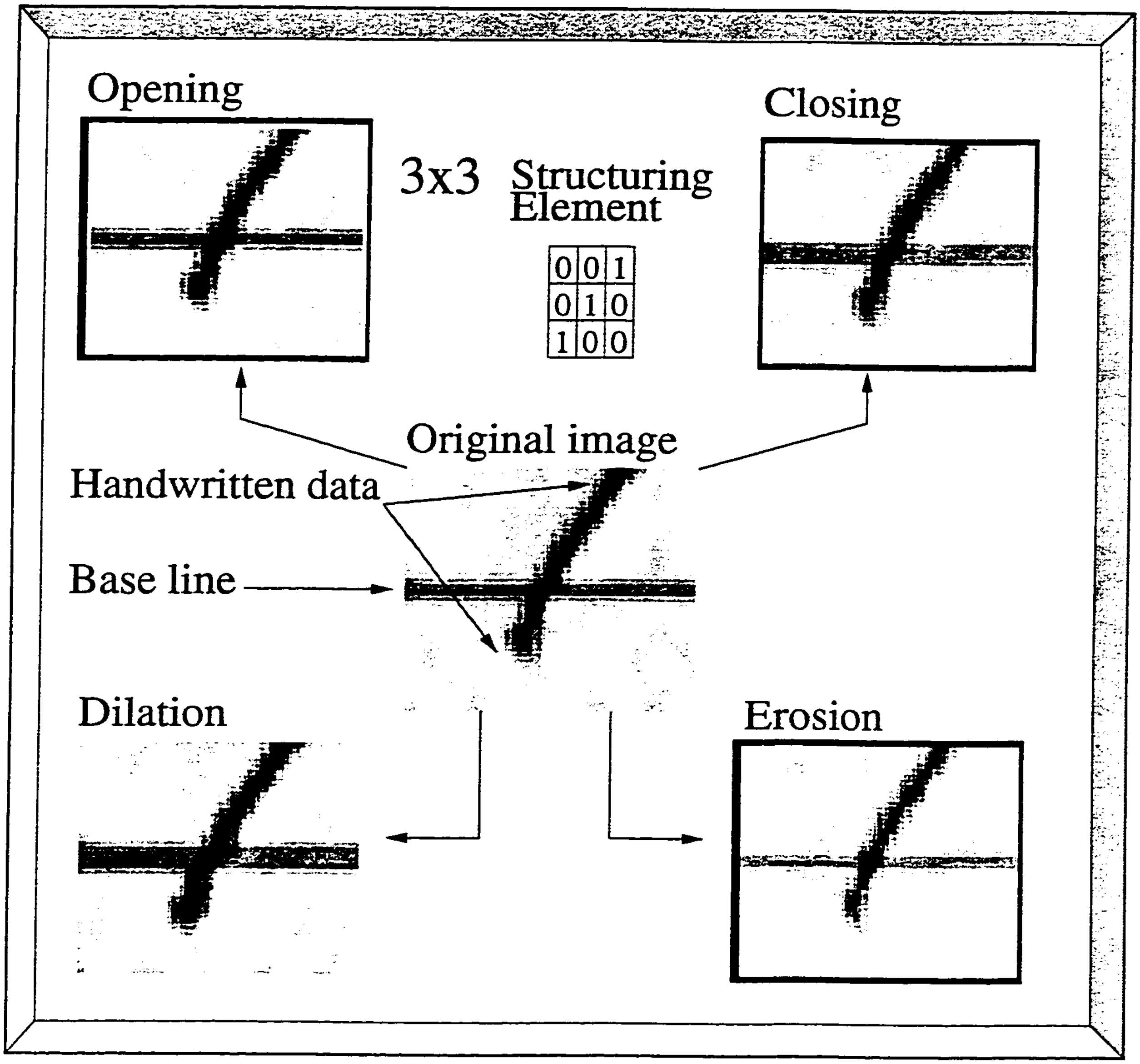

Figure 4.7: Dilation, erosion, opening, and closing of a gray-scale image using a 3×3 structuring element.

$$f_{union}(i,j) = \begin{cases} f_{clos}(i,j) & \text{if } (i,j) \in L_{s_k} \\ f_{lr}(i,j) & otherwise \end{cases} \quad (4.7)$$

Figures 4.10 (a) and 4.9 show the information gained after a morphological closing operation is performed. Figure 4.10 (b) shows the loss of information after the elimination of baselines according to Equation 6.2. Figure 4.10 (c) shows the result of combining Figure 4.10 (a) with Figure 4.10 (b) according to Equation 4.7.

4.9 Mathematical Morphology! Is it Enough?

After introducing mathematical morphology and showing that the morphological closing operation is needed in our implementation and how it helped in reducing the amount of lost handwritten information that intersects with the baselines on the cheques, it is important to ask the following question: Isn't it enough to use the closing operation to capture all the lost information and terminate the process by extracting the handwritten information? Unfortunately, the case is not as easy as one thinks. Further processing should be performed because there are cases where mathematical morphology does not restore all the lost information as illustrated in Figure 4.10. In the coming sections, we will introduce *topological processing* as a solution to fill the gaps which were not restored by mathematical morphology in order to restore as much information as possible after thresholding the baselines in the morphological image. But before we proceed we need to apply median filtering for noise elimination and image recovery on f_{union} as follows:

$$f_m(i,j) = Median\{f_{union}(i-k, j-l) \forall k, l \in W\},$$

where W is a suitably chosen window. In our work, a 3×3 window is employed.

4.10 Topological Processing

As mentioned in section 4.7, the purpose of morphological processing, namely the closing operation, is to reduce the intensity of baselines and increase the intensity of the handwritten data in order to minimize the loss of useful data resulting from a

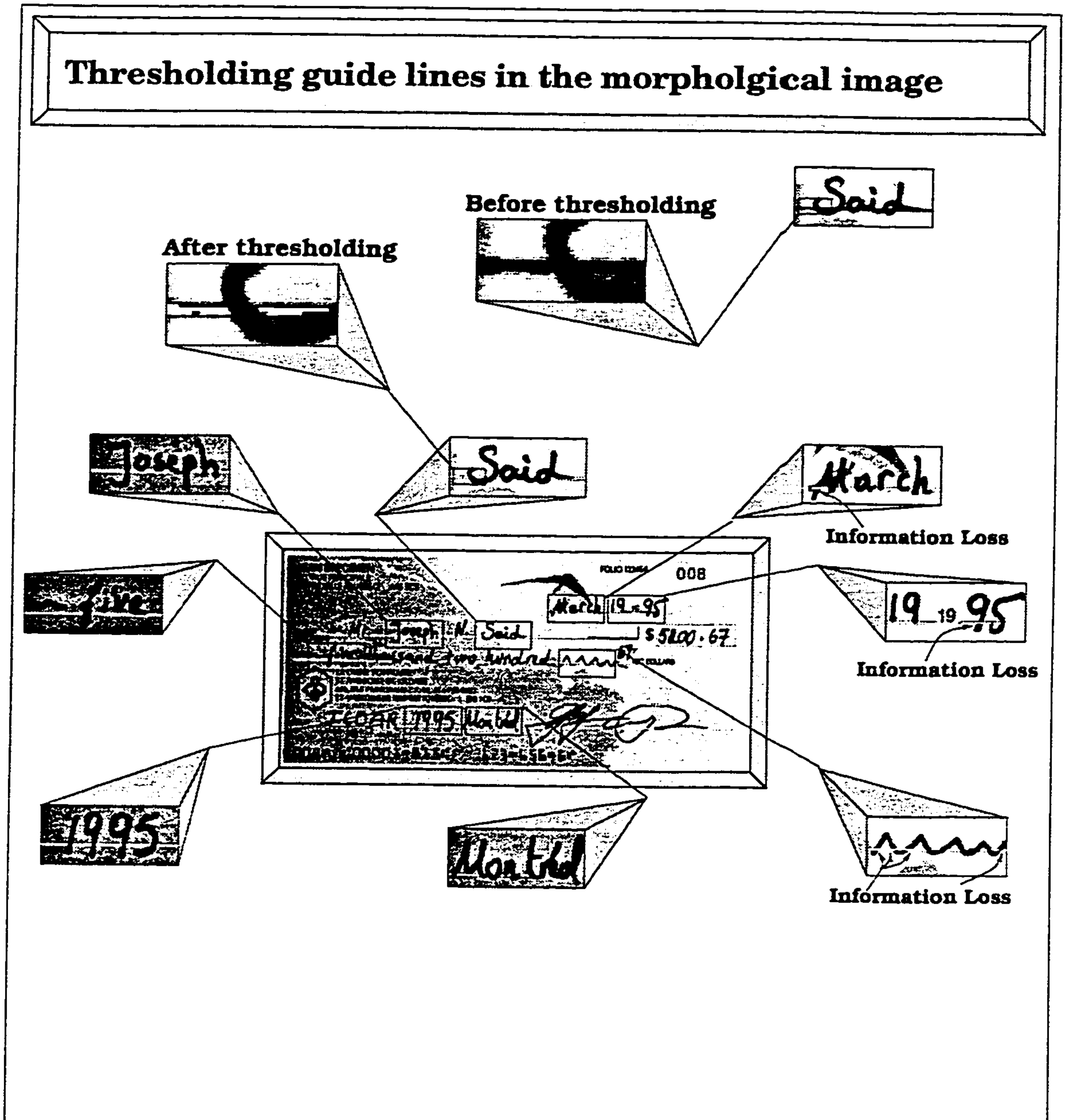

Figure 4.8: Thresholding lines from the morphological image according to their mean values.

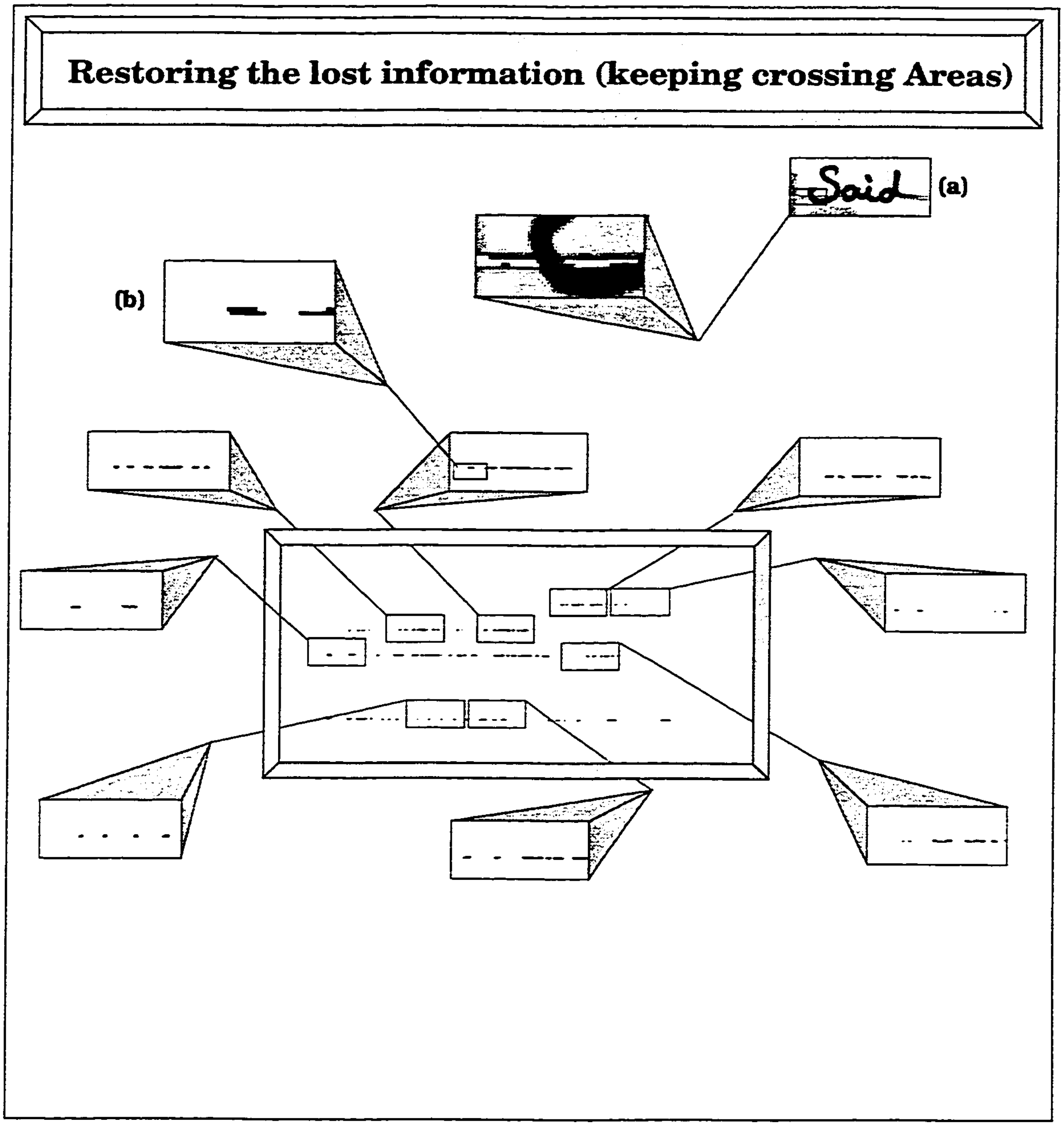

Figure 4.9: Restoring the information that belongs to the intersection between the handwritten data and the baselines.

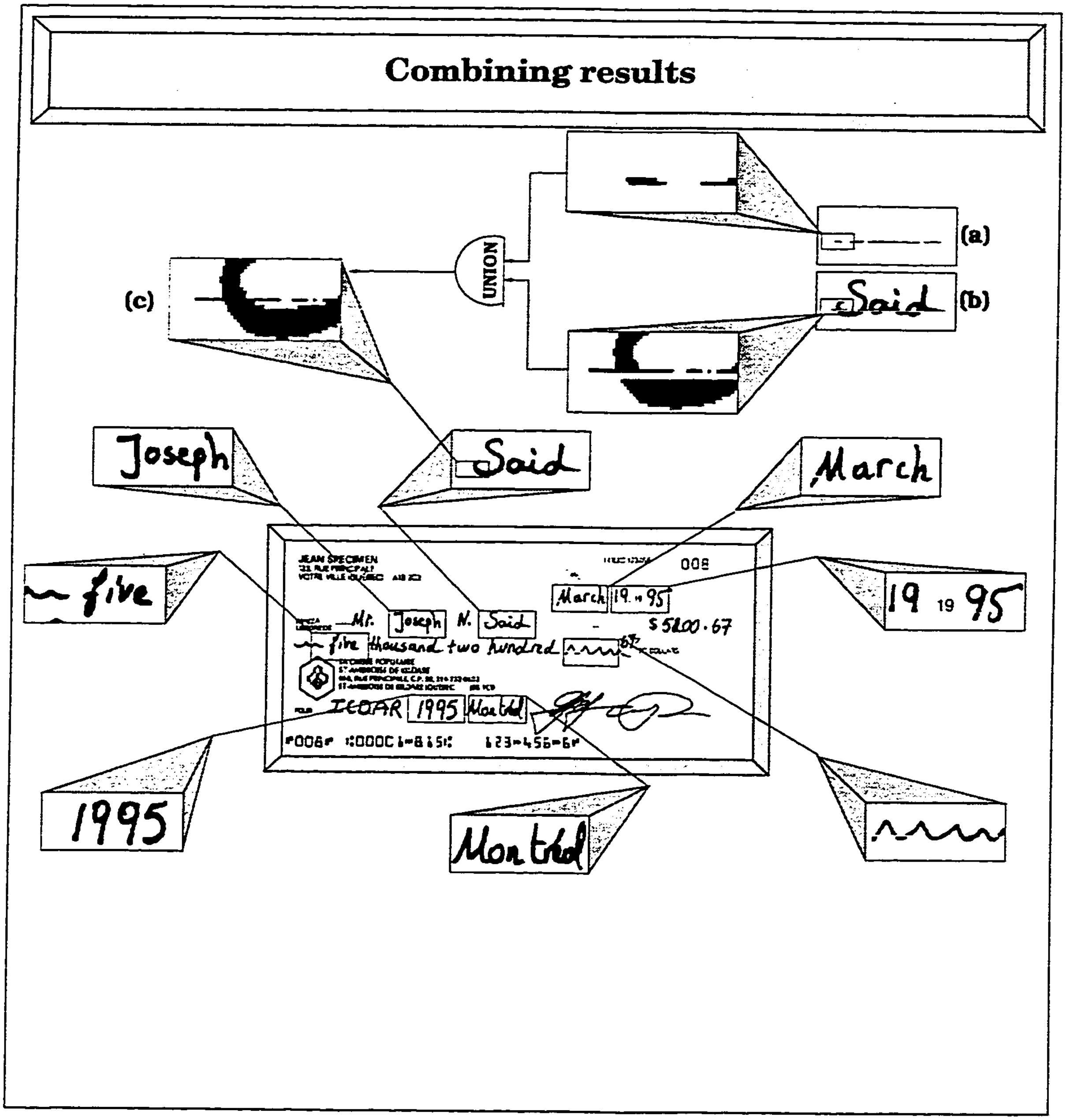

Figure 4.10: Restoring the lost information that intersects with the eliminated base-lines.

threshold that eliminates the baselines in the gray-scale morphological image. Unfortunately, this approach does not always completely restore the handwritten data that intersects with the baselines. In such cases, thresholding the baselines from f_{lr} will result in a loss in the topological properties of the handwritten information. As a solution to this problem, more intelligent processing should be applied to recover as much as possible the unfilled gaps, as illustrated in Figure 4.11. In pursuing this line, our algorithms identified the existence of the unfilled gaps between their corresponding extremities in the images and used an automated methodology to establish connections between these extremities to restore the lost information by filling the gaps. In fact, the closing operation contributes to enhancing the gaps in the line's region and makes the process of gaps identification and filling easier.

As illustrated by Figure 4.11 (a), three possible configurations of gaps have been considered according to the information lost with diagonal handwriting or perfect superposition. The four extremities that delimit gaps are labelled: L+, L-, R+, and R- as illustrated in Figure 4.17. Identification of gaps is achieved by first performing an edge detection algorithm on the image f_{union} with the purpose of finding the edges of the handwritten data which were lost due to thresholding the baselines in the morphological image f_{lr} and not restored in f_{clos} using the morphological closing operation. During the edge detection process, we will gather information about the magnitude and the digital gradient of each pixel; this will greatly help us to identify the existence of all unfilled gaps between the corresponding extremities. Figure 4.11 (b) demonstrates the extremities $L+, L-, R+$, and $R-$ that should be located using the digital gradient of each pixel and Figure 4.11 (a) presents the rules that are used to fill the gaps between corresponding extremities.

4.10.1 Edge Detection

In this work, we will consider the gradient magnitude $D(i,j)$ and the gradient direction $\theta(i,j)$ of a digitized image f_m at each pair of spatial coordinates (i,j), where $0 \leq i \leq h$ and $0 \leq j \leq w$, as:

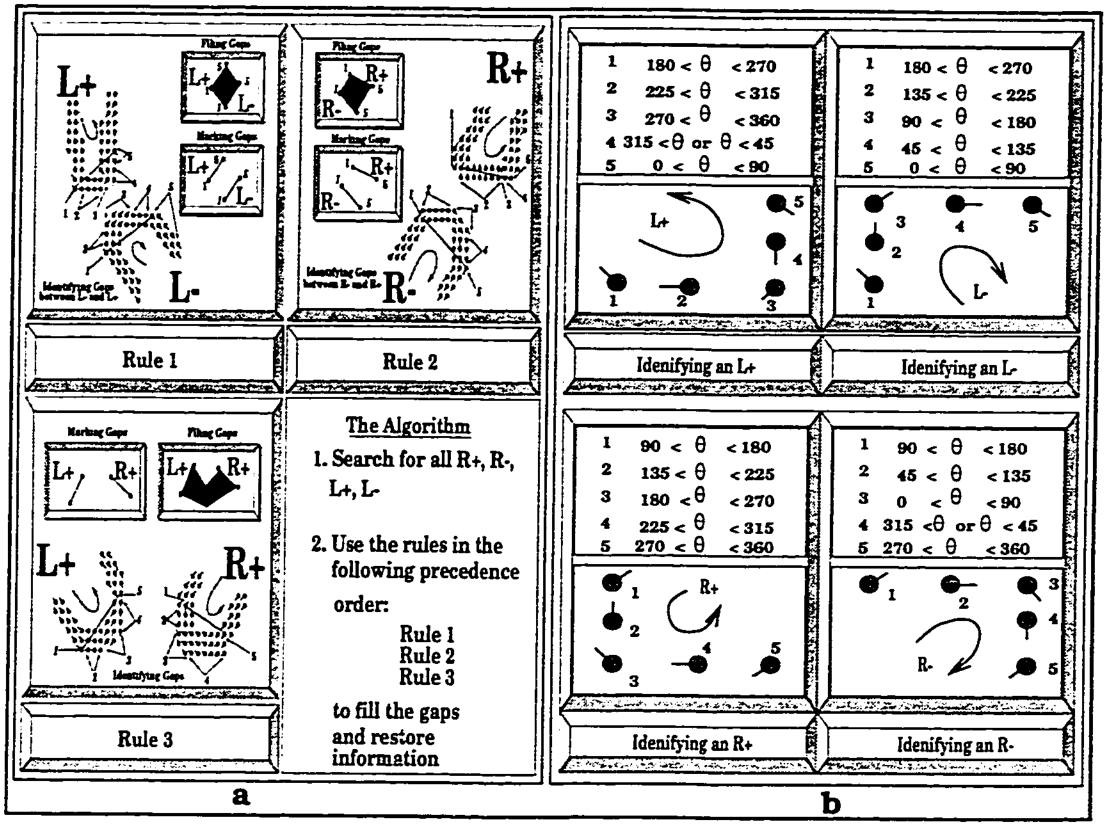

Figure 4.11: Rules for identification and filling of gaps. In calculating θ for each pixel value, the reference axis is considered to be relative for each pixel.

$$D(i,j) = \sqrt{(\triangle_x f_m(i,j))^2 + (\triangle_y f_m(i,j))^2} \tag{4.8}$$

$$\theta(i,j) = tan^{-1}(\triangle_y f_m(i,j))/(\triangle_x f_m(i,j)) \tag{4.9}$$

where

$$\begin{aligned} \triangle_x f_m(i,j) = f_m(i+1,j+1) + 2f_m(i+1,j) + f_m(i+1,j-1) \\ -f_m(i-1,j+1) - 2f_m(i-1,j) - f_m(i-1,j-1) \end{aligned} \tag{4.10}$$

and

$$\begin{aligned} \triangle_y f_m(i,j) = f_m(i-1,j+1) + 2f_m(i,j+1) + f_m(i+1,j+1) \\ -f_m(i-1,j-1) - 2f_m(i,j-1) - f_m(i+1,j-1). \end{aligned} \tag{4.11}$$

This technique is applied to the image f_m to produce the image f_{edge} that is used later for further processing. Figure 4.12 illustrates a sample process.

4.10.2 Identification and Filling of Gaps

As a second step after calculating the magnitude and the digital gradient of each pixel in the line's region of f_m, the system is able to go through f_{edge} searching for different extremities that delimit gaps; this algorithm uses different rules to identify the four possible extremities. Figures 4.11 (b) illustrates the rules in finding the four extremities when they exist and Figures 4.13 - 4.16 present the algorithms to identify each extremity. Consider, for example, the rule for identifying an L+ (Figure 4.11 (b) and 4.13) where the reference axis is considered to be relative to each pixel and a modified value of $\theta(i,j)$ in degrees $\forall\ 0 \leq i \leq h, 0 \leq j \leq w$ is used to facilitate processing. In this rule, there are five stages to pass through in order for the algorithm to declare that an L+ is found. In step 1 of the algorithm, after a pixel is located with a digital gradient θ ($180^0 < \theta < 270^0$), the algorithm starts searching for a neighbouring

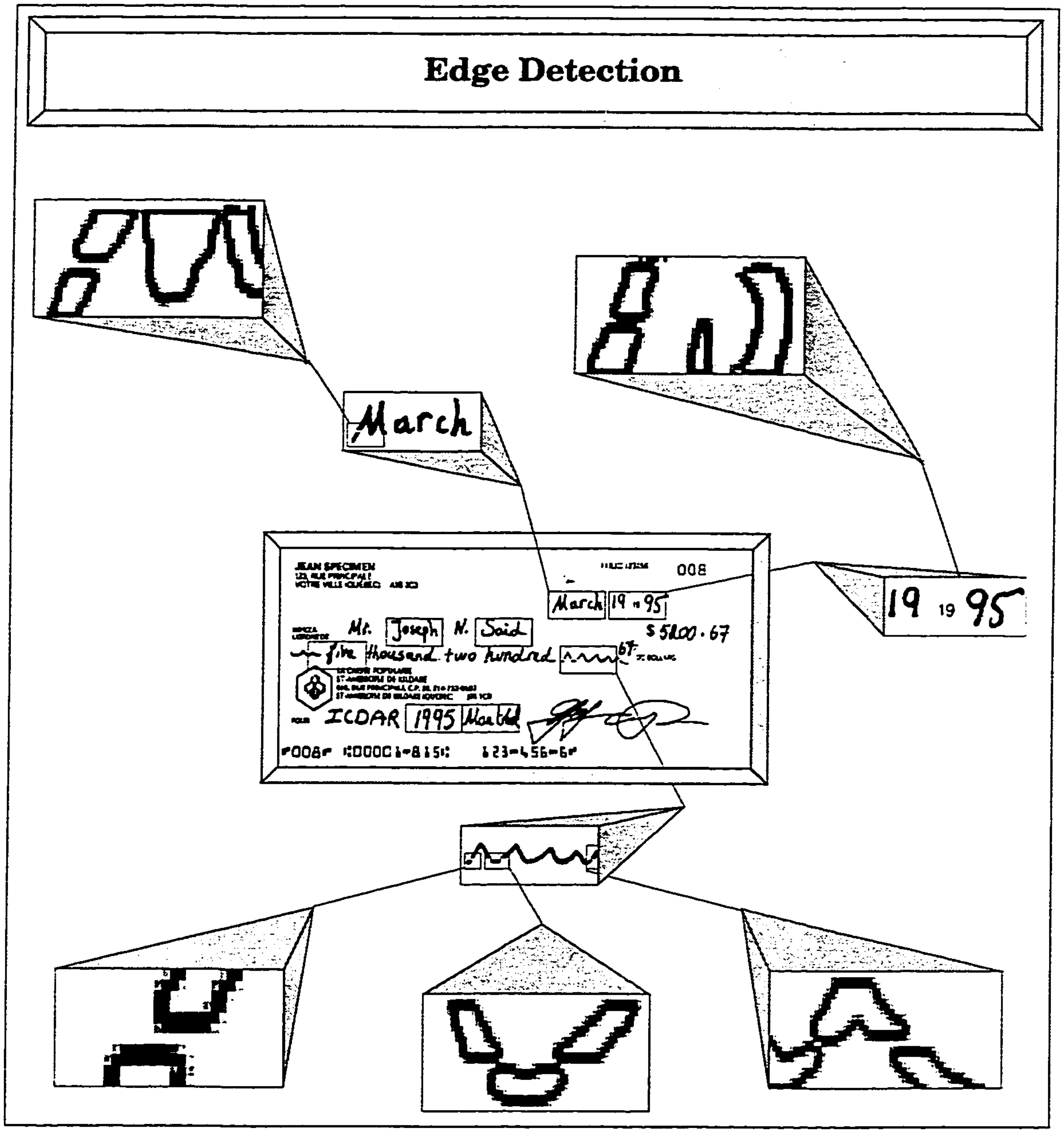

Figure 4.12: Performing edge detection to locate and identify the gaps.

pixel θ, such that θ ($135^0 < \theta < 225^0$). When such a θ ($135^0 < \theta < 225^0$) is read, the algorithm knows that step 2 is reached. Similarly, when steps 3, 4, and 5 are identified by the algorithm, the extremity L+ is found where a line is drawn between the pixel discovered at step 1 and the pixel discovered at step 5. (Figure 4.11 clearly illustrates this approach). The other extremities, L-, R+, and R-, are found following a similar concept.

Having identified all possible extremities, gaps are filled according to the following rules, as in Figure 4.17, in the order: (L+, L-), (R+, R-), and (L+, R+). In other words:

- If an L+ and an L- are only detected, then they must be connected.
- If an R+ and an R- are only detected, then they must be connected.
- If an L+ and an R+ are only detected, then they must be connected.
- If an L+, an L-, and an R+ are only detected connect the L+ to the L-.
- If an L+, L-, R+, and an R- are detected, then connect the L+ to the L- first and then connect the R+ to the R-.
- If an L+ and an R+ are only detected, then they must be connected.
- If an R+, an R-, and an L+ are only detected, then connect the R+ to the R-.

The image produced by this process is called f_{gaps}. Figure 4.11 (a) illustrates the last step in our algorithm that is associated with filling gaps. Given that we want to connect two extremities like L+ and L-, we will draw a filled rectangle between the two lines as shown in Figure 4.11. Figures 4.17, 4.18, and 4.2 (h) illustrate the final result.

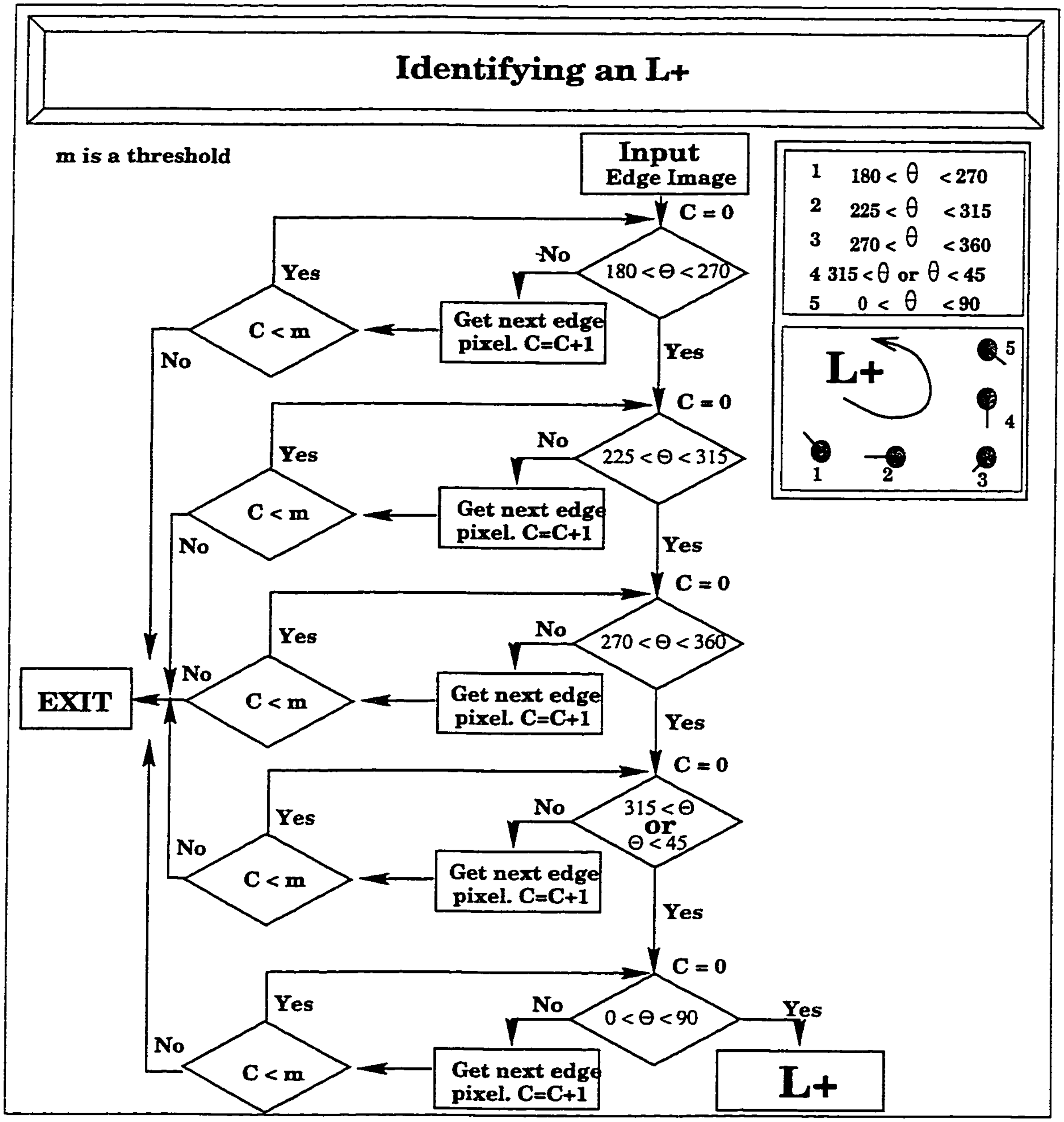

Figure 4.13: Identification of an L+ given the image f_{edge}.

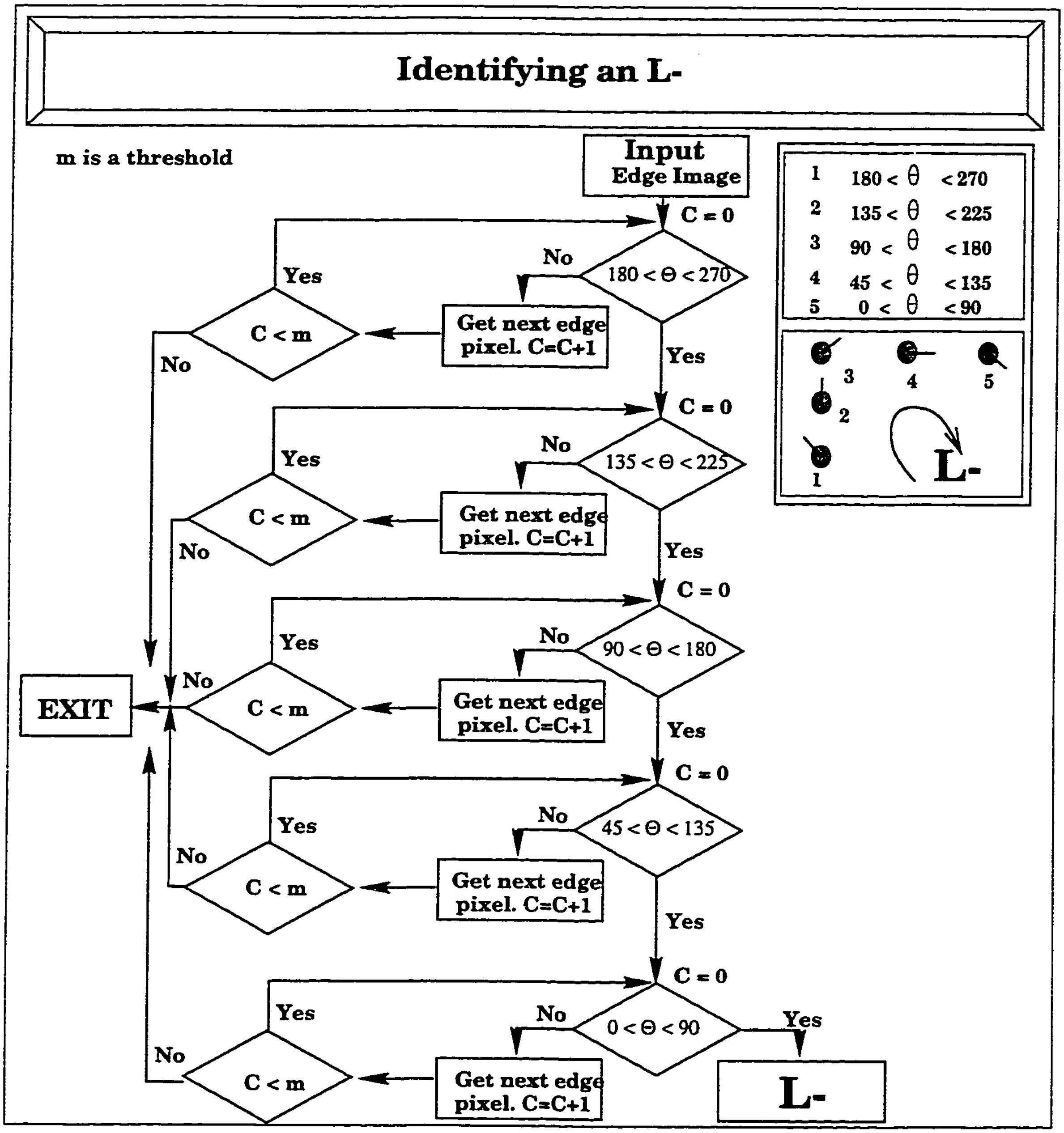

Figure 4.14: Identification of an L- given the image f_{edge}.

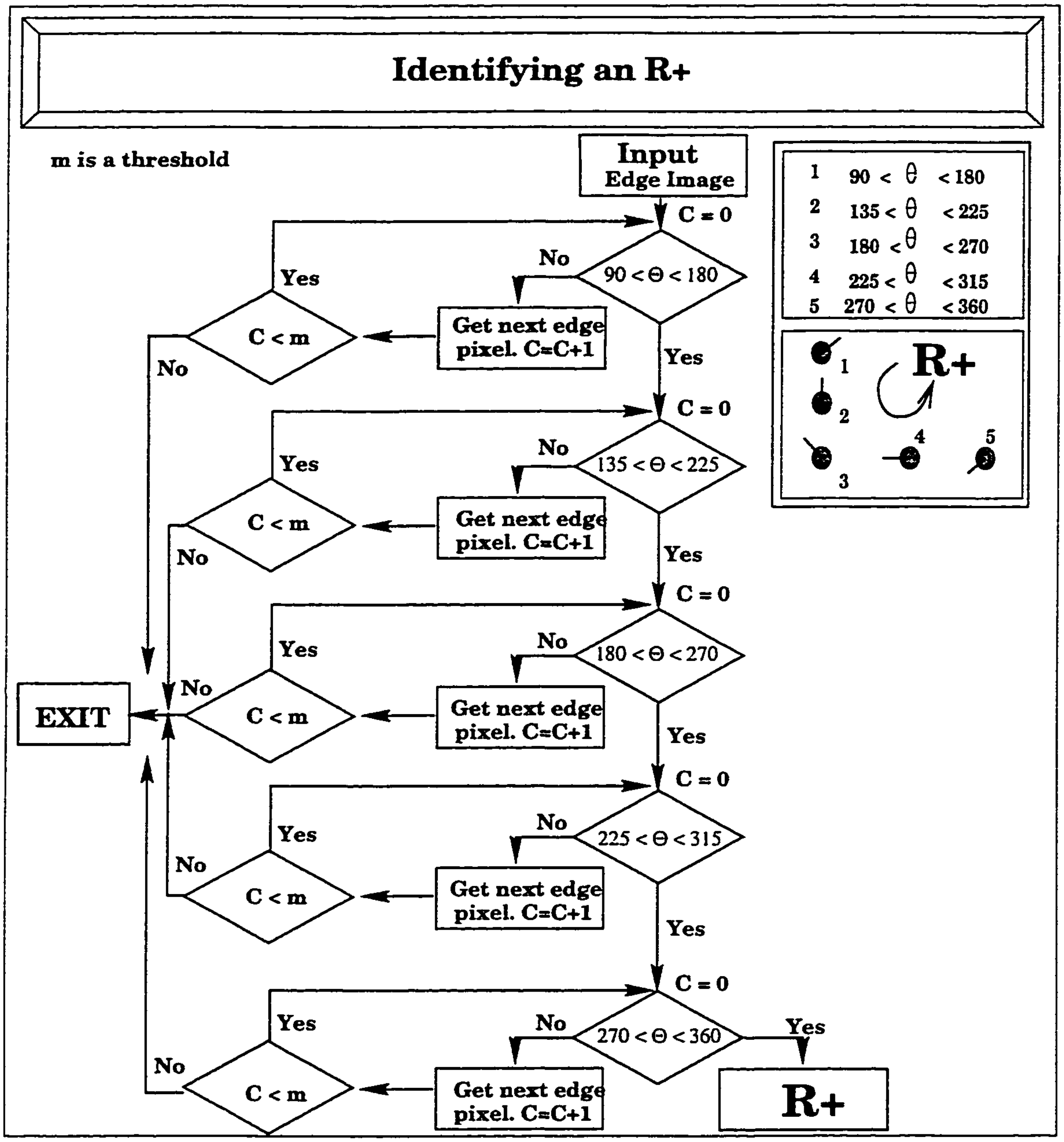

Figure 4.15: Identification of an R+ given the image f_{edge}.

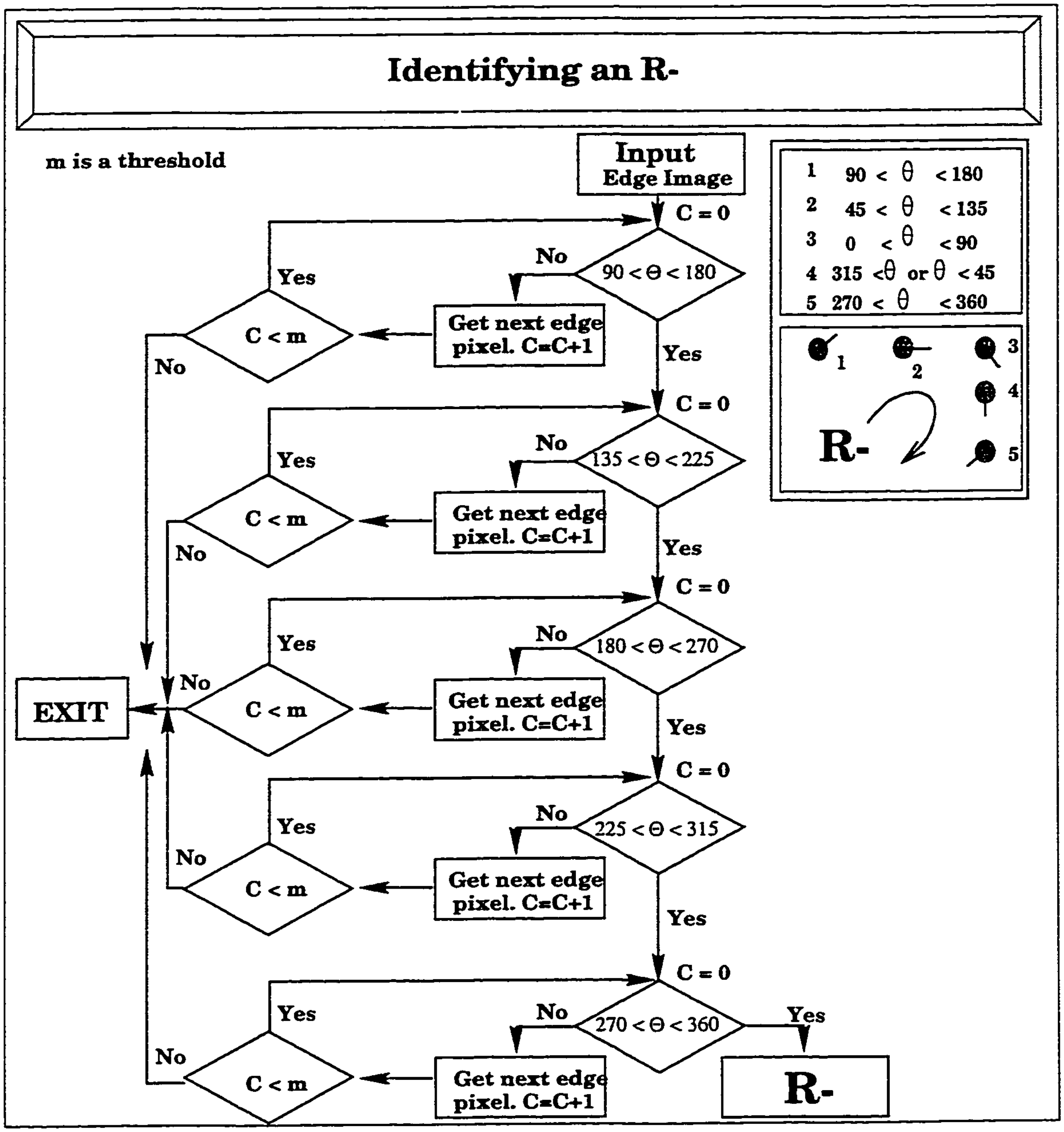

Figure 4.16: Identification of an R- given the image f_{edge}.

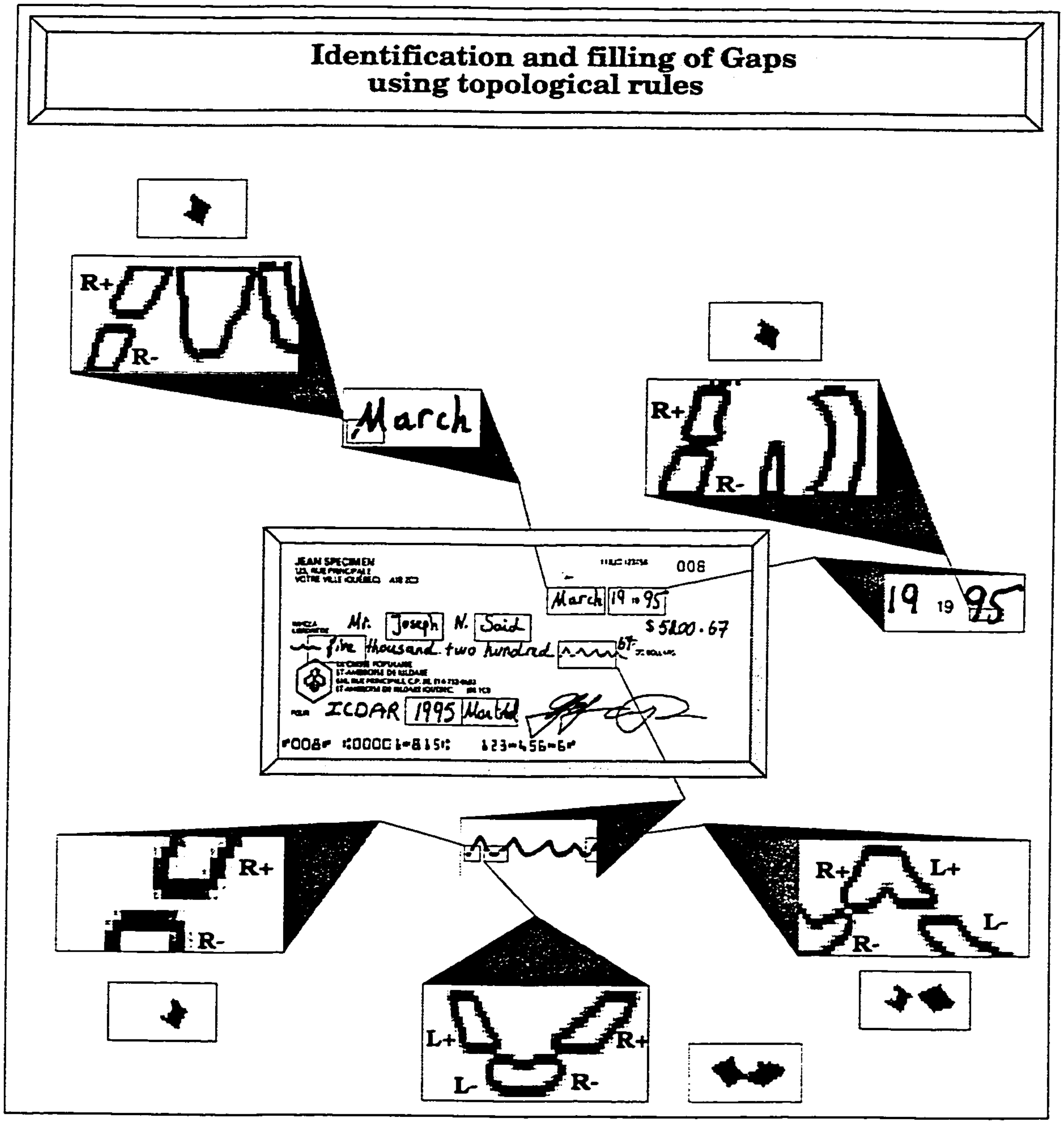

Figure 4.17: Identification and filling of gaps.

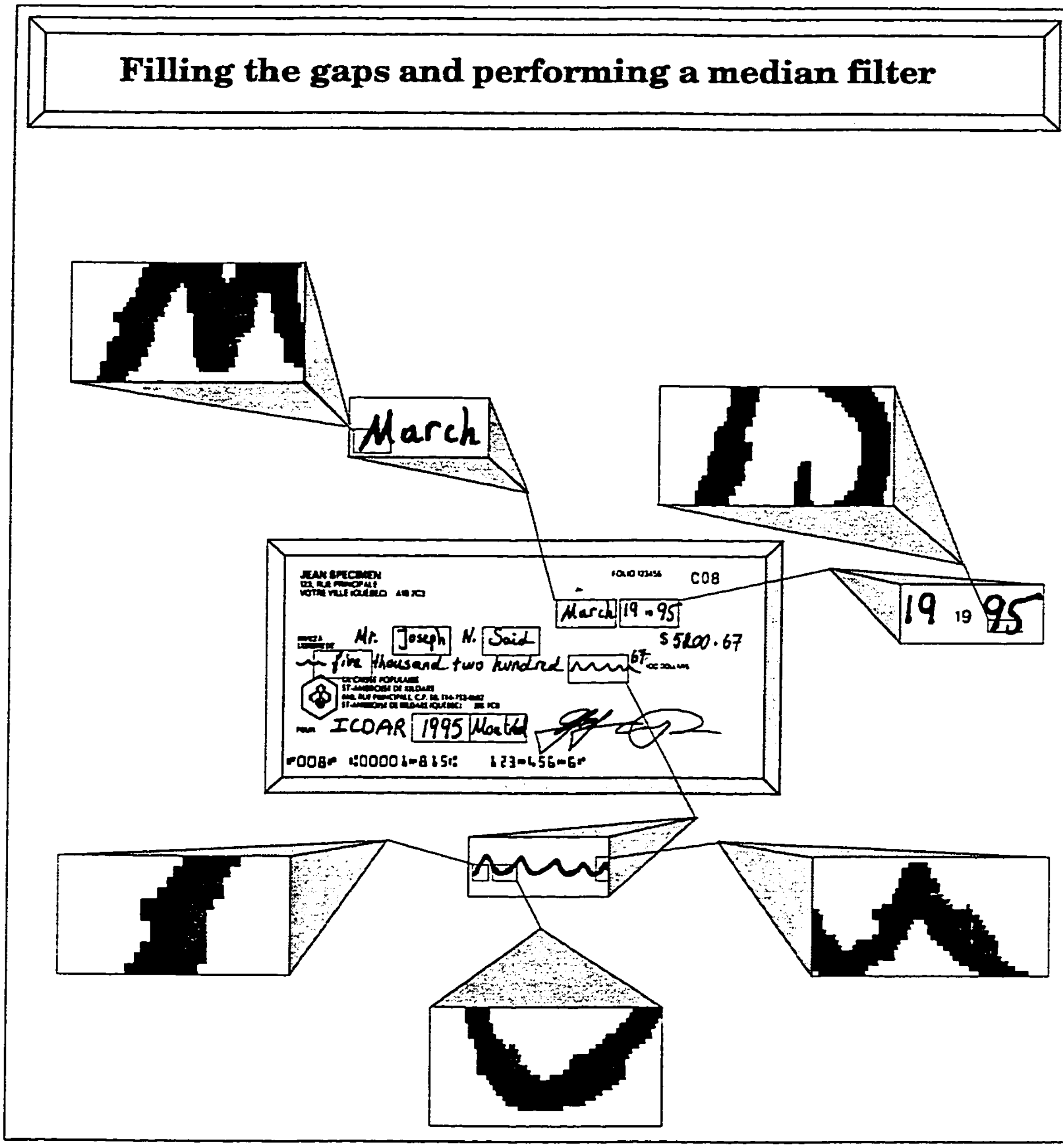

Figure 4.18: Final result after performing a median filtering and binarization (data is completely restored).

4.11 Extraction of Date, Courtesy Amount, and Legal Amount

Adopting and extending the notation as in [208], if $\alpha_f = \{\alpha_{f_1}, \alpha_{f_2}, \ldots, \alpha_{f_m}\}$ is a finite set of filled data and $L_s = \{L_{s_1}, L_{s_2}, L_{s_q}\}$ is the finite set of baselines, then we will define a finite set of relations denoted by $\Gamma = \{\Gamma_1, \Gamma_2, \ldots, \Gamma_k\}$ between the two sets α_f and L_s represented by a matrix M, such that:

$$M_{ij} = \Gamma_l \tag{4.12}$$

where $1 \leq i \leq m$ and $1 \leq j \leq q$, which satisfy the following condition: $\forall l(\Gamma_l = (\alpha_{f_i} \mathcal{R} L_{s_j}))$, $\mathcal{R} = \{R, L, A, B, I\}$, where R, L, A, B, and I represent Right, Left, Above, Below, and Intersect respectively.

It is clear that the filled data $\alpha_{f_i} \forall i \in \{1, \ldots, m\}$ could be easily located using the matrix M. We define the location description of α_{f_i} as a tuple whose components are: (a) the filled data α_{f_i} (b) any element of the set $\mathcal{R}$, and (c) a line L_{s_j}. Formally, we have:

$$(\alpha_{f_i}, r, L_{s_j}), \quad \alpha_{f_i} \in \alpha_f, \quad r \in \mathcal{R}, \quad L_{s_j} \in L_s. \tag{4.13}$$

An item description may have close relations with several graphs, so several location description tuples can be associated with each α_{f_i}.

Now given the matrix M and the location of all baselines $L_{s_i} \in L_s$, the system is able to extract α_{f_i} $\forall i \in \{1, \ldots, m\}$. The extracted set α_f could further be presented to a recognition system. Referring to Figure 4.3, the matrix M is represented as follows:

$$M = \begin{array}{c} \\ \alpha_{f_1} \\ \alpha_{f_2} \\ \alpha_{f_3} \\ \alpha_{f_4} \end{array} \begin{array}{c} \begin{array}{cccc} L_{s_1} & L_{s_2} & L_{s_3} & L_{s_4} \end{array} \\ \begin{pmatrix} I & A & A & A \\ B & R & A & A \\ B & B & I & A \\ B & B & B & I \end{pmatrix} \end{array}$$

In accordance with Equation 4.13, we can express the location description of each α_{f_i} as follows:

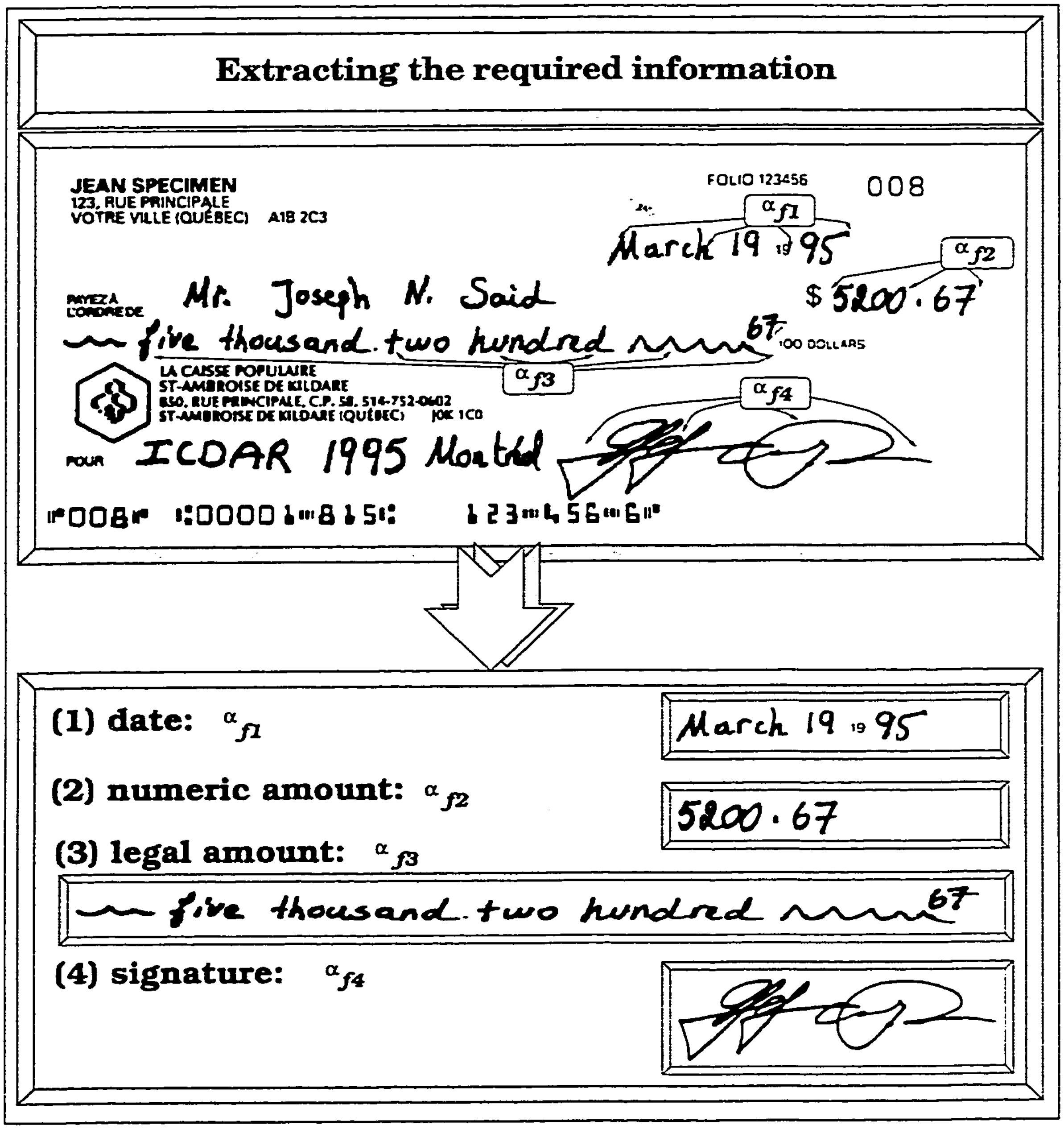

Figure 4.19: Extraction of the finite set of filled information $\alpha_f = \{\alpha_{f_1}, \alpha_{f_2}, \alpha_{f_3}, \alpha_{f_4}\}$.

1. For α_{f_1} we have $(\alpha_{f_1}, I, L_{s_1})$, $(\alpha_{f_1}, A, L_{s_2})$; $(\alpha_{f_1}, A, L_{s_3})$; $(\alpha_{f_1}, A, L_{s_4})$ which is interpreted as α_{f_1} intersects with L_{s_1} and is located above L_{s_2}, L_{s_3}, and L_{s_4}.

2. For α_{f_2} we have $(\alpha_{f_2}, B, L_{s_1})$; $(\alpha_{f_2}, R, L_{s_2})$; $(\alpha_{f_2}, A, L_{s_3})$; $(\alpha_{f_2}, A, L_{s_4})$ which is interpreted as α_{f_2} is located above L_{s_1}, on the right of L_{s_2}, and above L_{s_3}, and L_{s_4}.

3. For α_{f_3} we have $(\alpha_{f_3}, B, L_{s_1})$; $(\alpha_{f_3}, B, L_{s_2})$; $(\alpha_{f_3}, I, L_{s_3})$; $(\alpha_{f_3}, A, L_{s_4})$ which is interpreted as α_{f_3} is below L_{s_1} and L_{s_2}, intersects L_{s_3}, and is above L_{s_4}.

4. For α_{f_4} we have $(\alpha_{f_4}, B, L_{s_1})$; $(\alpha_{f_4}, B, L_{s_2})$; $(\alpha_{f_4}, B, L_{s_3})$; $(\alpha_{f_4}, I, L_{s_4})$ which is interpreted as α_{f_4} is located below L_{s_1}, L_{s_2}, L_{s_3} and intersects with L_{s_4}.

4.12 Experimental Results

Four hundred and twenty two bank cheques were used for training and 203 cheques were used for testing the performance of the system. The performance analysis results were produced following the same approach as that used in Chapter 3 Section 3.8. The experimental results in **TABLE II** indicate that the testing set of bank cheques have good and satisfactory results. The reason behind this is that when the handwritten information is darker than the background, background elimination becomes easier and the resulting image will contain less noise, see Figures 4.20 and 4.21. However, when the handwritten information is lighter than the background, background elimination causes a significant loss in the luminance and topological properties of the handwritten information and as a result the handwritten information will not be of any importance since recognition of the extracted handwritten data will be almost impossible.

TABLE II **Testing Set (non of the training images were used during testing)**		
# of images	*Background/Foreground (f) of a given cheque*	*Accuracy of the previous system*
40	DS/{t,T}D^+	99%-100%
23	DS/{t,T}L^+	96%-100%
17	DS/{t,T}L	88%-100%
31	D^+S/{t,T}{D^+,L^+,L}	76%-87%
25	DC/{t,T}D^+	99%-100%
26	DC/{t,T}L^+	94%-100%
14	DC/{t,T}L	82%-100%
8	D^+C/{t,T}{D^+,L^+,L}	61%-82%
8	DC^+/{t,T}D^+	99%-100%
3	DC^+/{t,T}L^+	91%-100%
3	DC^+/{t,T}L	0%
5	D^+C^+/{t,T}{D^+,L^+,L}	0%

The last two entries in **TABLE II** illustrate a case where cheques has a very complicated dark background and a lighter foreground; in such a case the system was not able to preserve the luminance and the topological properties of the foreground objects of interest resulting in an unsatisfactory processing. In fact, systems that deal with cheque processing are faced with this problem too. As a solution to this problem we will present in later Chapters a solution that will overcome this difficulty.

From Figure 4.22, one can notice that the system requires more time when the cheque images have a very dark (**D+**) and a very complex (**C+**) background. In fact, the last two entries in Figure 4.22 indicate that the system rejected the cheque images because their background was so complex that it can not be segmented further, else the handwritten information is partially lost which makes it impossible to be properly recognized. The first entry in Figure 4.22 shows that when the background is dark (**D**) and simple (**S**) versus a thin (**t**) or thick (**T**) and very dark (**D+**) handwriting, then the system becomes very robust and requires less processing time.

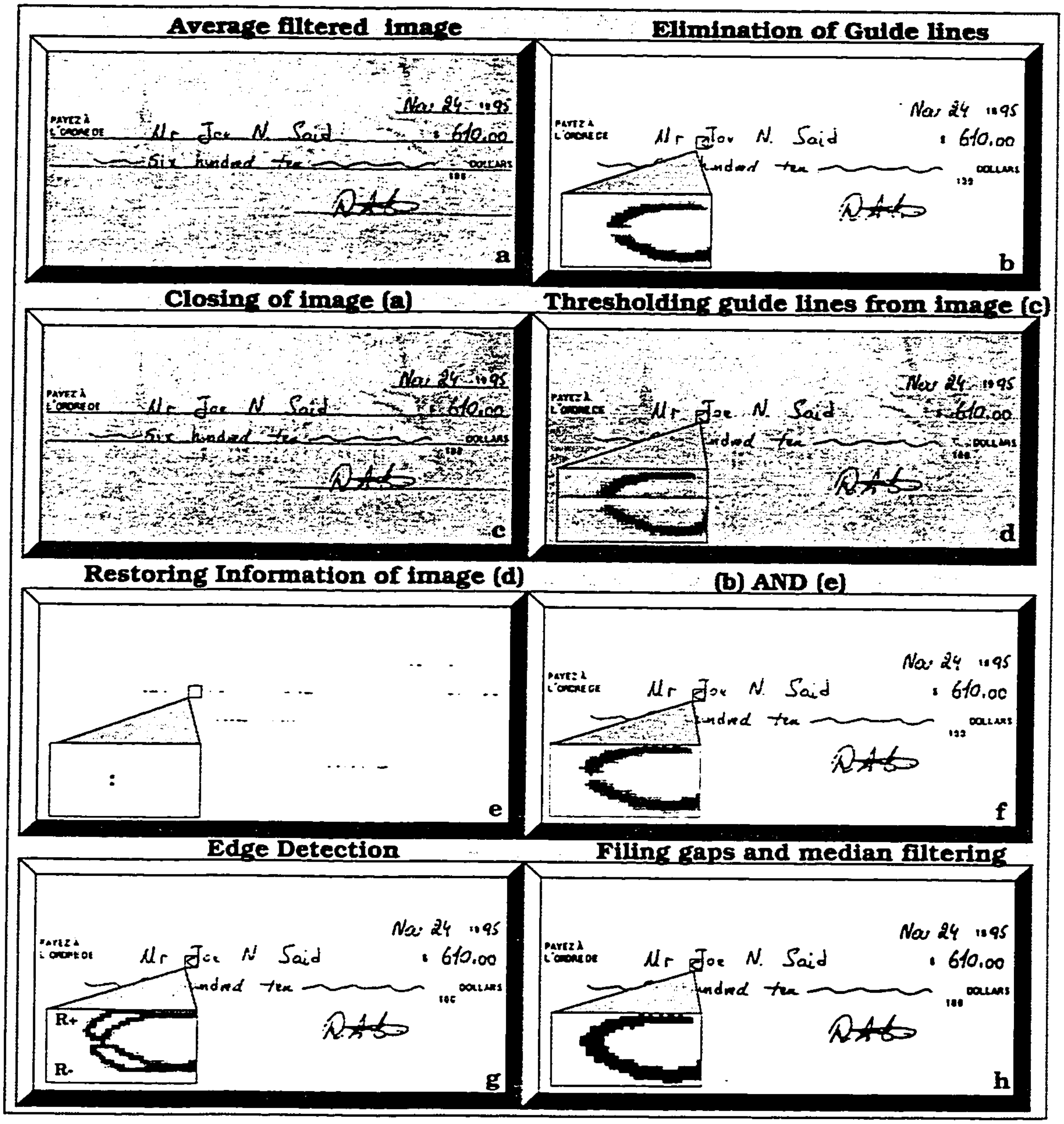

Figure 4.20: A sample cheque with (D) dark and (C) complex background and (t) thin and (D) handwritten information.

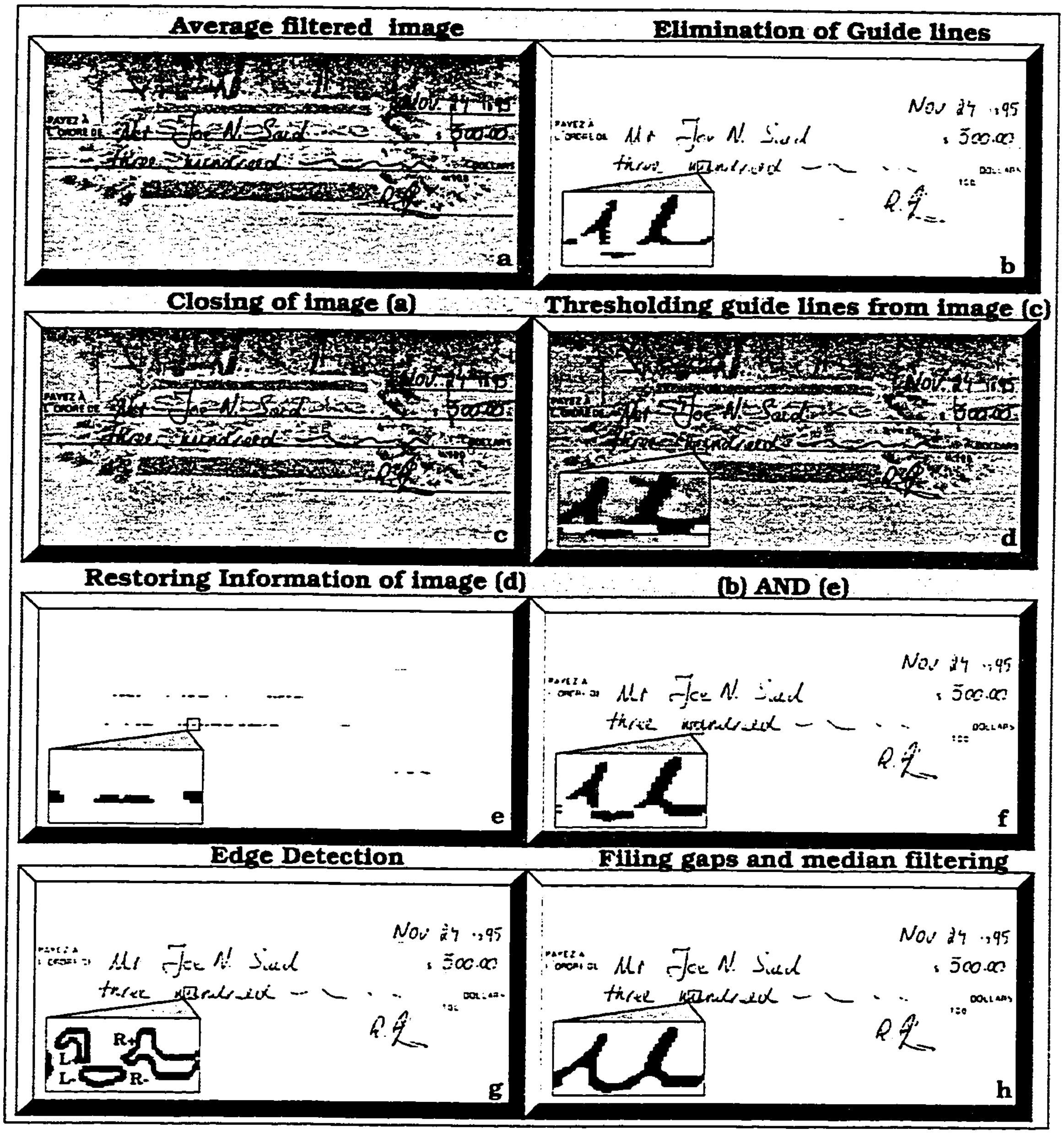

Figure 4.21: A sample cheque processing.

Average Processing Time

Background/Foreground	A	B	C	D	E	F	G	H	I	J
	Average Time (sec)									(min)
D S /{t,T} D+	17	2	1	23	1	33	20	2	30	2.09
D S /{t,T} L+	17	2	1	23	1	33	24	3	30	2.14
D S /{t,T} L	17	2	1	23	1	33	26	6	30	2.19
D S /{t,T} {D+, L+, L}	17	2	1	23	1	33	30	6	30	2.23
D C /{t,T} D+	17	5	1	23	1	33	22	3	30	2.15
D C /{t,T} L+	17	5	1	23	1	33	25	4	30	2.19
D C /{t,T} L	17	5	1	23	1	33	29	6	30	2.25
D C /{t,T} {D+, L+, L}	17	5	1	23	1	33	35	8	30	2.33
D C+ /{t,T} D+	17	5	1	23	1	33	29	6	30	2.25
D C+ /{t,T} L+	17	5	1	23	1	33	33	8	30	2.31
D C+ /{t,T} L	17	5	0	0	0	0	0	0	0	0.22
D C+ /{t,T} {D+, L+, L}	17	5	0	0	0	0	0	0	0	0.22

A: f_{filt} C: f_{lr} E: f_{AND} G: f_{edge} I: f_m

B: f_{thr} D: f_{clos} F: f_m H: f_{gaps} J: *Total Time*

Figure 4.22: Average processing time observed on different types of real life bank cheque images.

4.13 Discussion and Motivation to Extend the Model

Section 4.1 introduced a very efficient and novel approach for processing the gray-scale images of bank cheques towards the extraction of specific fields located inside these documents. In fact, when the handwritten information is darker than the background, background elimination becomes less complicated to threshold and the resulting image will contain less noise as well as the topological properties of the handwritten information is preserved. Unfortunately, having a dark handwriting and a light background is not always the case. When the handwritten information is lighter than or is about to be the same as that of the background, background elimination using a global thresholding technique on the whole gray-scale cheque image will cause a significant loss in the luminance and topological properties of the targeted handwritten information. Figures 4.23, 4.24, 4.25, and 4.26 illustrate some of these cases. To overcome this difficulty, a new approach will be presented in Section 5.1 to increase the productivity of the system introduced in this Chapter. The new approach will seek specific regions and dynamically locate them inside the gray-scale documents to perform a local thresholding technique to better preserve the finest granularity of the extracted information. This will result in a more robust system that is able to visualize, understand, and extract handwritten information from different types of document images that have various intensities of handwriting against a simple or a complex background.

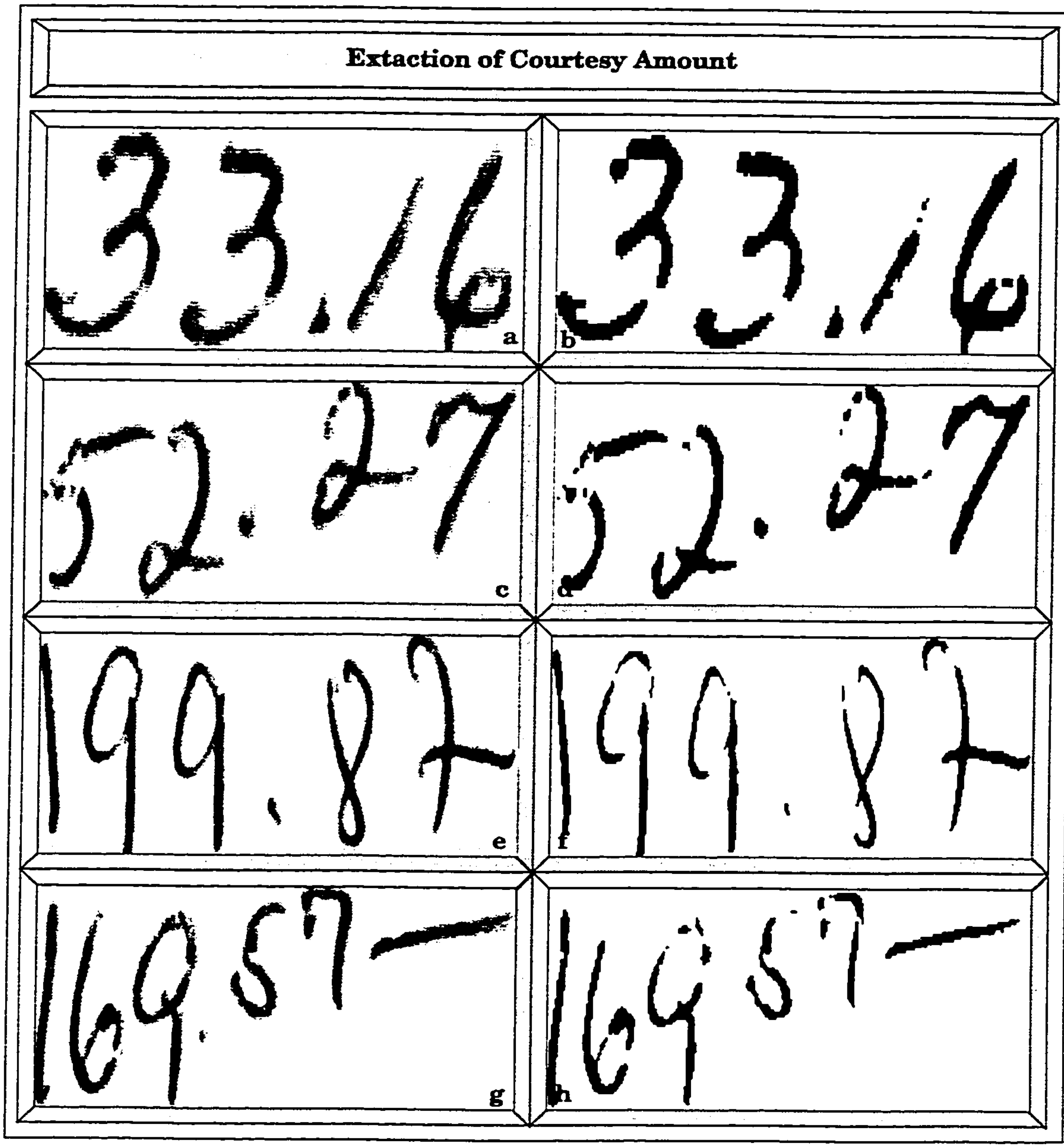

Figure 4.23: Sample output images to illustrate that global thresholding in some cases could cause distortion to the luminance and to the topological properties of the extracted items.

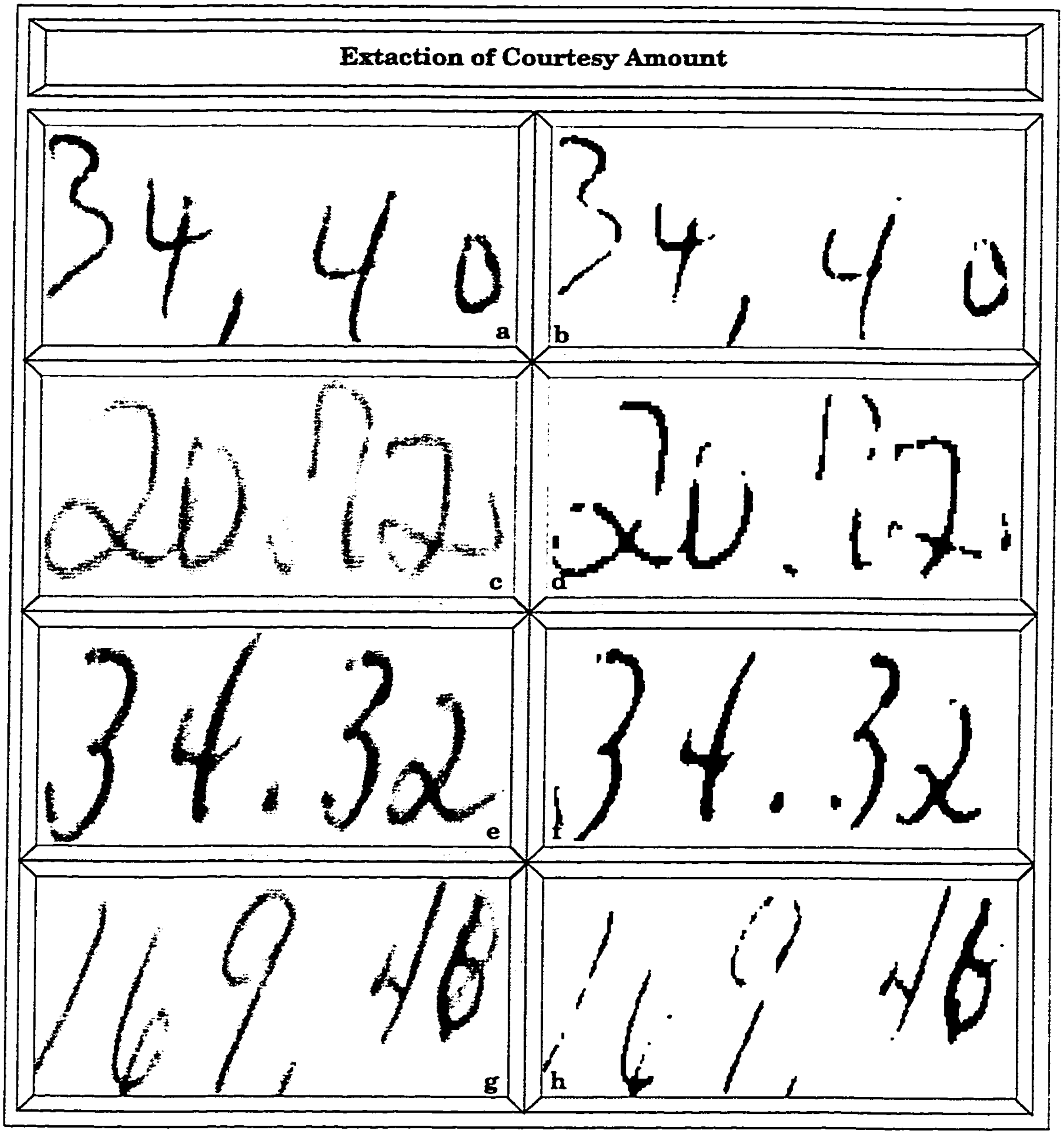

Figure 4.24: (continues) Sample output images to illustrate that global thresholding in some cases could cause distortion to the luminance and to the topological properties of the extracted items.

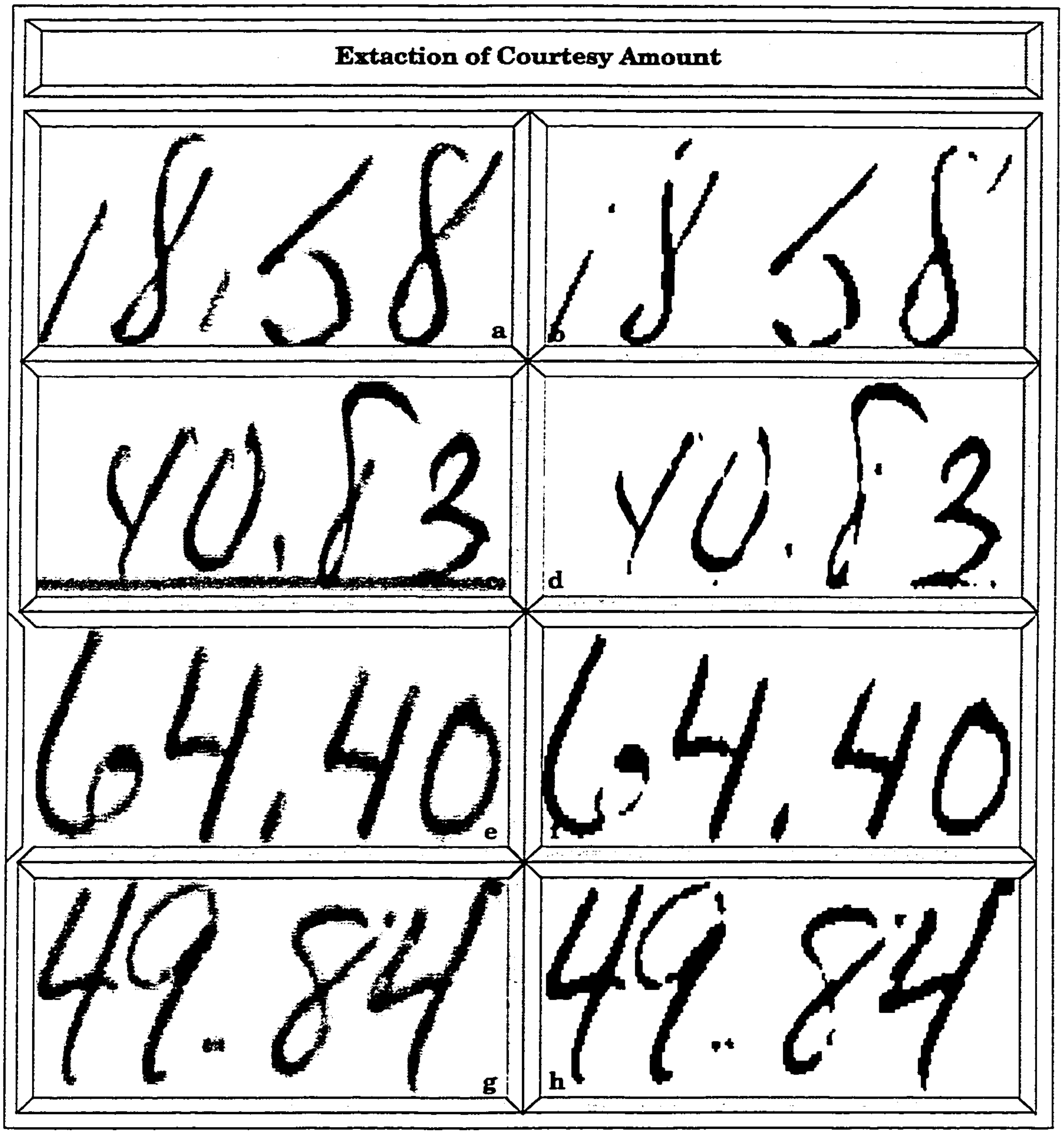

Figure 4.25: (continues) Sample output images to illustrate that global thresholding in some cases could cause distortion to the luminance and to the topological properties of the extracted items.

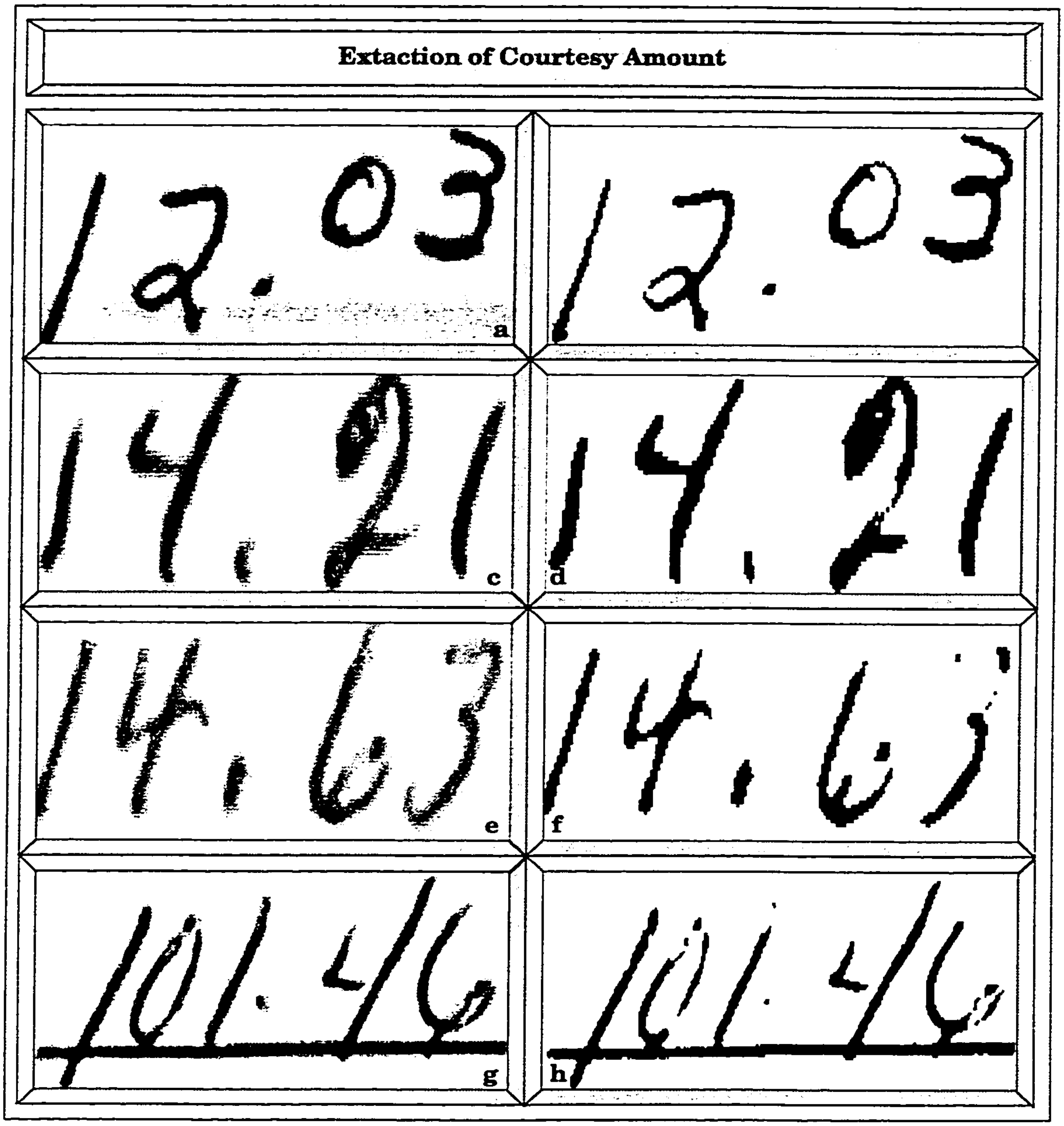

Figure 4.26: (continues) Sample output images to illustrate that global thresholding in some cases could cause distortion to the luminance and to the topological properties of the extracted items.

Chapter 5

An Extended Formal Model for Visual Data Processing of Business Forms and Bank Cheques

Local processing to enhance the quality of the extracted items.

5.1 Introduction

Chapter 4 introduced a very efficient and novel approach for processing the gray-scale images of bank cheques towards the extraction of specific fields located inside these documents. In fact, according to Chapter 4 when the handwritten information is darker than the background, background elimination becomes less complicated to threshold and the resulting image will contain less noise. Moreover, the topological properties of the handwritten information is preserved. Unfortunately, having a dark handwriting and a light background is not always the case. When the handwritten information is lighter than or is about the same as that of the background, background elimination using a global thresholding technique on the whole gray-scale cheque image will cause a significant loss in the luminance and topological properties of the handwritten information. To overcome this difficulty, a new approach will be presented in this Chapter to increase the productivity of the system introduced in Chapter 4. The new approach will seek specific regions and dynamically locate them inside the gray-scale documents and perform a local thresholding technique, instead of a global one introduced in Chapter 4, to better preserve the finest granularity of the extracted information. This will result in a more robust system that is able to visualize, understand, and extract handwritten information from different types of document images that have various intensities of handwriting against a simple or a complex background.

5.2 Methodology

As stated earlier the intention in this new approach is to enable the system to preserve and visualize the finest granularity of data written or printed on the cheque images. In pursuing this line, we will use a local segmentation technique as opposed to the global segmentation one presented in Chapter 3. This will enhance the quality of the extracted information and further increase the visibility and understandability of the system. This is a very important contribution towards improving the recognition results of the extracted information.

Referring to Figure 5.1, the process of extracting handwritten data from a given bank cheque image f is extended in the following manner: (1) eliminate the background after performing an image enhancement (average filtering) on the original image f; (2) locate the baselines and extract the required items from the original image e.g. date, courtesy amount and legal amount; (3) and finally, according to Figure 5.2, process locally the extracted items using the approach presented in Chapter 4.

In the following sections the detailed process of the new approach is presented with an application on the extraction of date, courtesy and legal amounts written on bank cheques.

5.3 Image Enhancement

Given a gray-scale cheque image f, an enhanced image f_{filt} is produced according to Equation 3.9. Figure 4.4 illustrates an example.

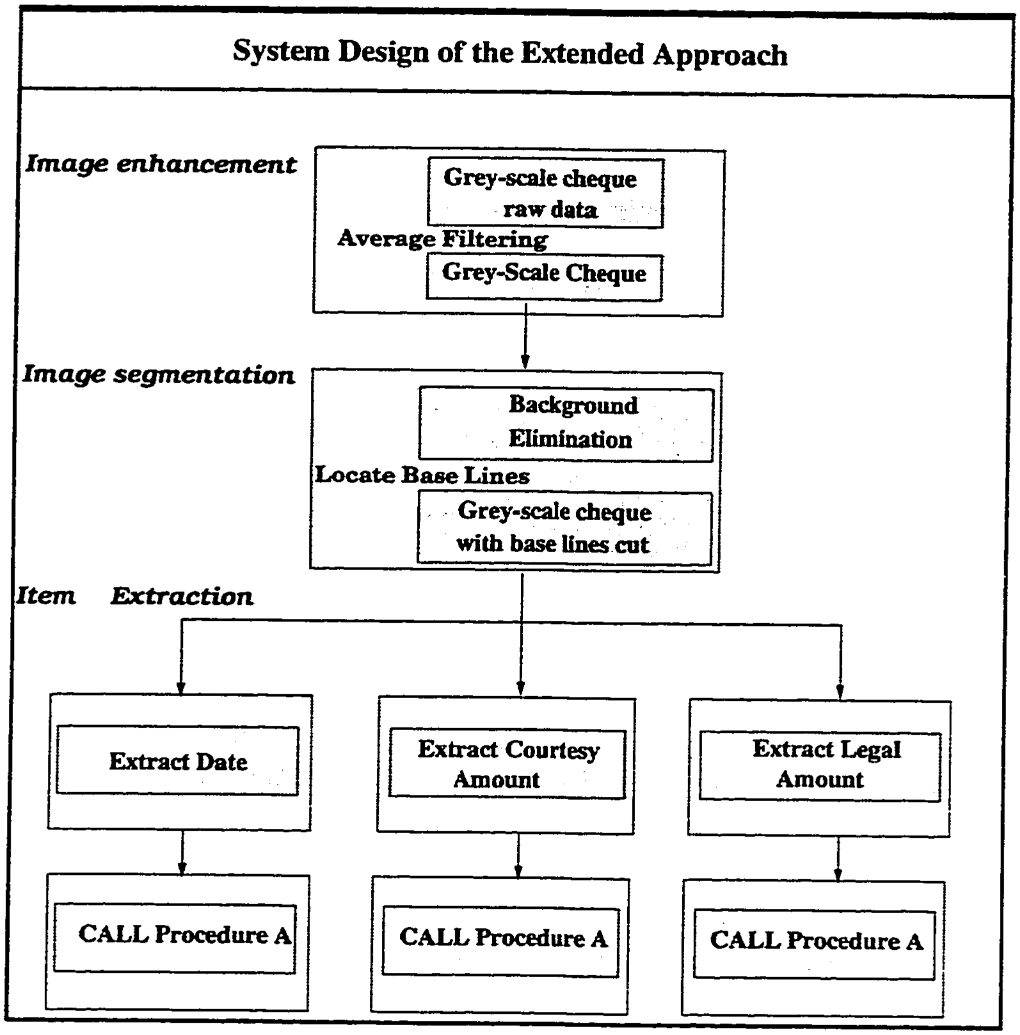

Figure 5.1: System design of the extended approach.

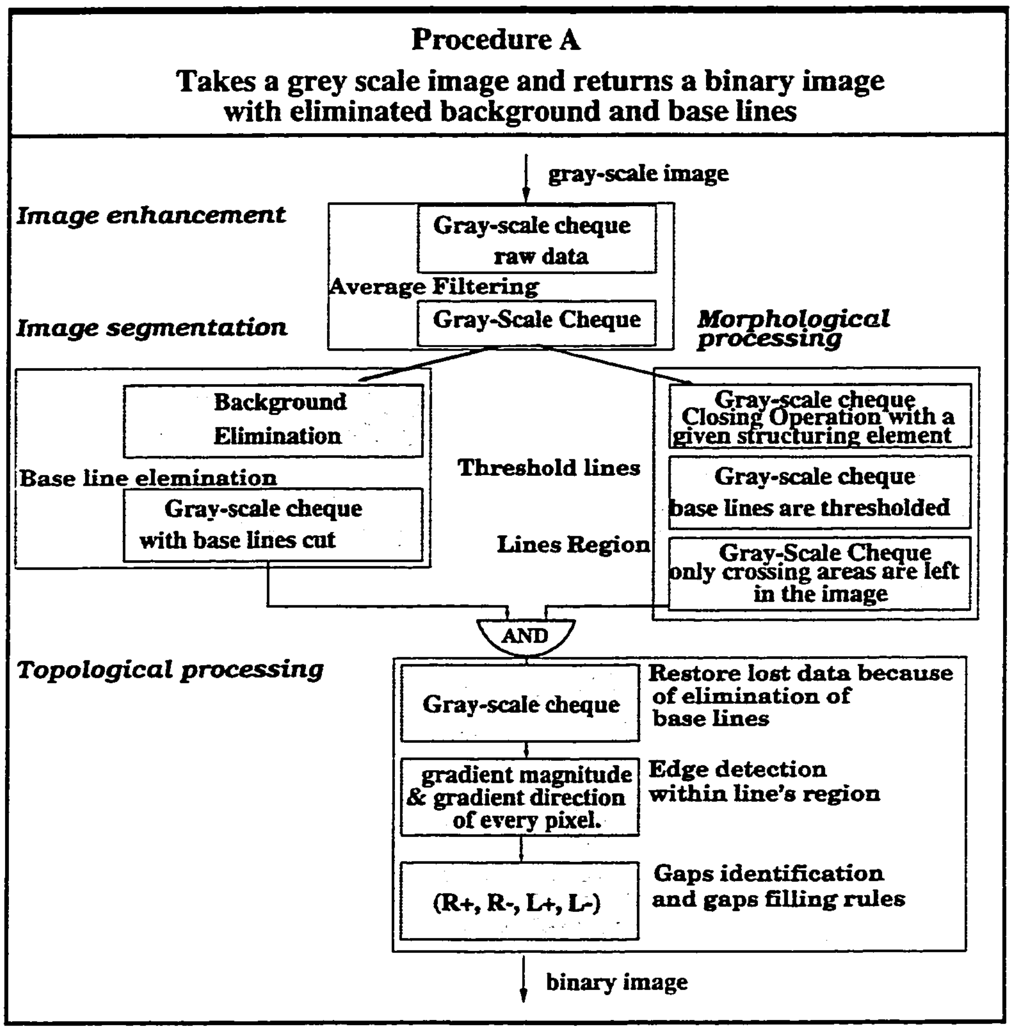

Figure 5.2: Procedure A.

5.4 Global Recursive Thresholding

The thresholding technique of Section 4.5 is applied to f_{filt} to produce the image f_{thr}. i.e.,

$$f_{filt} = C_0 \cup C_1 \cup \ldots \cup C_p \tag{5.1}$$

where $f_{thr} = C_p$.

5.5 Baseline Determination and Early Extraction

Determine $L_s = \{L_{s_1}, L_{s_2}, L_{s_3}\}$ the finite set of baselines, where L_{s_1}, L_{s_2}, and L_{s_3} represent the date baseline, the courtesy amount baseline, and the legal amount baseline. Determine the matrix M, as in Section 4.11 that describes the location description tuples to extract the required information. Referring to Figure 5.3 (a), for example, the matrix M is represented as follows:

$$M = \begin{array}{c c} & \begin{array}{ccc} L_{s_1} & L_{s_2} & L_{s_3} \end{array} \\ \begin{array}{c} \alpha_{f_1} \\ \alpha_{f_2} \\ \alpha_{f_3} \end{array} & \begin{pmatrix} A & A & A \\ B & A & A \\ B & B & I \end{pmatrix} \end{array}.$$

Now we can express the location description of each α_{f_i} as follows:

1. For α_{f_1} we have $(\alpha_{f_1}, A, L_{s_1})$, $(\alpha_{f_1}, A, L_{s_2})$, and $(\alpha_{f_1}, A, L_{s_3})$. This is interpreted as α_{f_1}, the date in image f_{filt}, is located above L_{s_1}, above L_{s_2}, and above L_{s_3}.

2. For α_{f_2} we have $(\alpha_{f_2}, B, L_{s_1})$, $(\alpha_{f_2}, A, L_{s_2})$, and $(\alpha_{f_2}, A, L_{s_3})$. This is interpreted as α_{f_2}, the courtesy amount in image f_{filt}, which is located below L_{s_1}, above L_{s_2}, and above L_{s_3}.

3. For α_{f_3} we have $(\alpha_{f_3}, B, L_{s_1})$; $(\alpha_{f_3}, B, L_{s_2})$; and $(\alpha_{f_3}, I, L_{s_3})$. This is interpreted as α_{f_3}, the legal amount in image f_{filt}, which is below L_{s_1}, below L_{s_2}, and intersects L_{s_3}.

Now, given the matrix M, extract $\alpha_f = \{f_D, f_C, f_L\}$ the finite set of filled data. f_D, f_C, and f_L represent the gray-scale image of the extracted date, the gray-scale image of the extracted courtesy amount, and the gray-scale image of the extracted legal amount respectively as illustrated in Figure 5.3 (c).

5.6 Enhancement of Extracted Images (Date, Courtesy Amount, and Legal Amount)

Enhance f_k, $\forall k \in \{D, C, L\}$ according to Equation 3.9 to produce the images $f_{k_{filt}}$. Formally we have:

$$f_{D_{filt}}(i,j) = \frac{1}{K} \sum_{(n,m)\in S} f_D(n,m), \ \forall (i,j) \in W_D \times H_D \tag{5.2}$$

$$f_{C_{filt}}(i,j) = \frac{1}{K} \sum_{(n,m)\in S} f_C(n,m), \ \forall (i,j) \in W_C \times H_C \tag{5.3}$$

$$f_{L_{filt}}(i,j) = \frac{1}{K} \sum_{(n,m)\in S} f_L(n,m), \ \forall (i,j) \in W_L \times H_L \tag{5.4}$$

where K is the total number of points in S which is assumed to be a 3×3 neighbourhood and W_k and H_k $\forall k \in \{D, C, L\}$ are the width and height of the images f_D, f_C and f_L respectively.

5.7 Local Recursive Thresholding

Locally use recursive thresholding as presented in Section 4.5 for background elimination of the gray-scale images $f_{k_{filt}}$ $\forall k \in \{D, C, L\}$ as follows:

$$f_{D_{filt}} = D_1 \cup \ldots \cup D_p \tag{5.5}$$

$$f_{C_{filt}} = C_1 \cup \ldots \cup C_p \tag{5.6}$$

$$f_{L_{filt}} = L_1 \cup \ldots \cup L_p \tag{5.7}$$

where $f_{k_{thr}} = k_p$ $\forall k \in \{D, C, L\}$ as illustrated in Figure 5.4.

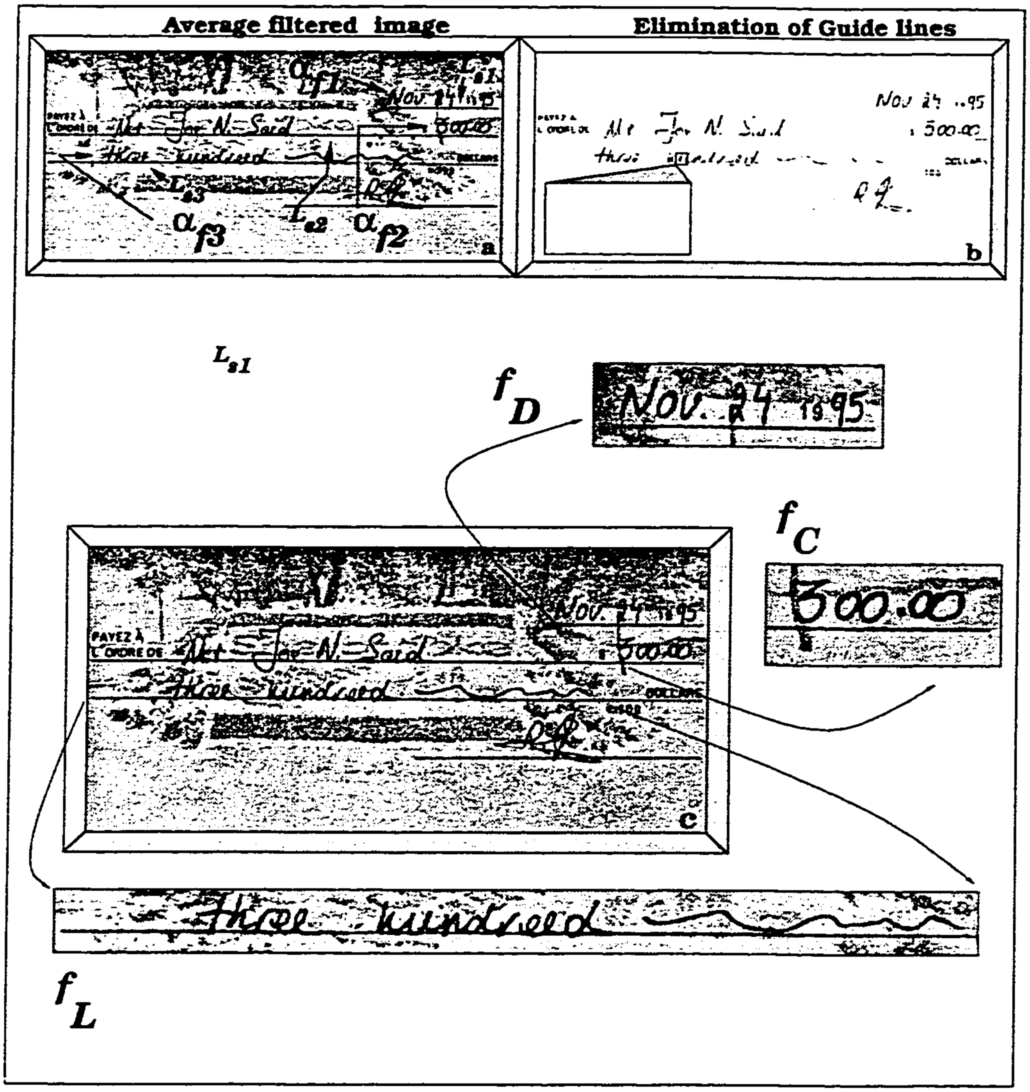

Figure 5.3: The new system and early extraction.

5.8 Local Baseline Elimination

Given the images $f_{D_{thr}}$, $f_{C_{thr}}$, and $f_{L_{thr}}$ as in Figure 5.4 (e), (j), and (o), the purpose is to search these images for the baselines L_{s_D}, L_{s_C}, and L_{s_L} respectively and eliminate them to produce the images $f_{D_{lr}}$, $f_{C_{lr}}$, and $f_{L_{lr}}$ for further processing.

In $f_{D_{thr}}$, determine the location description of α_{f_D}, the filled in date, as the tuple $(\alpha_{f_D}, A, L_{s_D})$ and eliminate the lines as in Equation 6.2 to produce the gray-scale image $f_{D_{lr}}$. Formally, this is defined as:

$$f_{D_{lr}}(i,j) = \begin{cases} 255 & \text{if } (i,j) \in L_{S_D} \\ f_{D_{thr}}(i,j) & otherwise \end{cases} \tag{5.8}$$

In $f_{C_{thr}}$, determine the location description of α_{f_C}, the filled in courtesy amount, as the tuple $(\alpha_{f_D}, A, L_{s_C})$ and eliminate the lines as in Equation 4.5.3 to produce the gray-scale image $f_{C_{lr}}$. Formally, this is defined as:

$$f_{C_{lr}}(i,j) = \begin{cases} 255 & \text{if } (i,j) \in L_{S_C} \\ f_{C_{thr}}(i,j) & otherwise \end{cases} \tag{5.9}$$

Similarly, in $f_{L_{thr}}$, determine the location description of α_{f_L}, the filled in legal amount, as the tuple $(\alpha_{f_L}, A, L_{s_L})$ and eliminate the lines as in Equation 4.5.3 to produce the gray-scale image $f_{L_{lr}}$. Formally, this is defined as:

$$f_{L_{lr}}(i,j) = \begin{cases} 255 & \text{if } (i,j) \in L_{S_L} \\ f_{L_{thr}}(i,j) & otherwise \end{cases} \tag{5.10}$$

5.9 Local Morphological Processing

Perform the morphological closing operation on $f_{k_{filt}}$ $\forall k \in \{D, C, L\}$ according to Equation 4.2 to produce the images $f_{k_{clos}}$. Formally we have:

$$f_{D_{clos}} \bullet V = (f_{D_{filt}} \oplus V) \ominus V \tag{5.11}$$

$$f_{C_{clos}} \bullet V = (f_{C_{filt}} \oplus V) \ominus V \tag{5.12}$$

$$f_{L_{clos}} \bullet V = (f_{L_{filt}} \oplus V) \ominus V \tag{5.13}$$

where V is the structuring element used in Equation 4.2.

5.10 Local Restoration of Lost Information

This operation restores the lost information from the morphological images $f_{k_{clos}}$ $\forall k \in \{D, C, L\}$ according to Equations 4.6 and 4.7 to produce the image $f_{k_{union}}$. Formally we have:

$$f_{D_{clos}}(i,j) = \begin{cases} f_{D_{clos}}(i,j) & \text{if } f_{D_{clos}}(i,j) \leq M_D \\ 255 & otherwise \end{cases} \tag{5.14}$$

$$f_{C_{clos}}(i,j) = \begin{cases} f_{C_{clos}}(i,j) & \text{if } f_{C_{clos}}(i,j) \leq M_C \\ 255 & otherwise \end{cases} \tag{5.15}$$

$$f_{L_{clos}}(i,j) = \begin{cases} f_{L_{clos}}(i,j) & \text{if } f_{L_{clos}}(i,j) \leq M_L \\ 255 & otherwise \end{cases} \tag{5.16}$$

$$f_{D_{union}}(i,j) = \begin{cases} f_{D_{clos}}(i,j) & \text{if } f_{D_{res}}(i,j) \in L_{s_D} \\ f_{D_{lr}}(i,j) & otherwise \end{cases} \tag{5.17}$$

$$f_{C_{union}}(i,j) = \begin{cases} f_{C_{clos}}(i,j) & \text{if } f_{C_{res}}(i,j) \in L_{s_C} \\ f_{C_{lr}}(i,j) & otherwise \end{cases} \tag{5.18}$$

$$f_{L_{union}}(i,j) = \begin{cases} f_{L_{clos}}(i,j) & \text{if } f_{L_{res}}(i,j) \in L_{s_L} \\ f_{L_{lr}}(i,j) & otherwise \end{cases} \tag{5.19}$$

where M_D, M_C, and M_L are the mean pixel values of the lines L_{s_D}, L_{s_C}, and L_{s_L} in the images $f_{D_{clos}}, f_{C_{clos}}$ and $f_{L_{clos}}$ respectively.

5.11 Local Edge Detection

Within the lines region, this operation determines the edges of the images f_{k_m}, $\forall k \in \{D, C, L\}$, where f_{D_m}, f_{C_m}, and f_{L_m} are the images after performing a median filtering operation on the images $f_{D_{union}}$, $f_{C_{union}}$, and $f_{L_{union}}$. Formally we have:

$$D_{f_D}(i,j) = \sqrt{(\Delta_x f_{D_m}(i,j))^2 + (\Delta_y f_{D_m}(i,j))^2} \tag{5.20}$$

$$D_{f_C}(i,j) = \sqrt{(\Delta_x f_{C_m}(i,j))^2 + (\Delta_y f_{C_m}(i,j))^2} \tag{5.21}$$

$$D_{f_L}(i,j) = \sqrt{(\triangle_x f_{L_m}(i,j))^2 + (\triangle_y f_{L_m}(i,j))^2} \tag{5.22}$$

$$\theta_{f_D}(i,j) = tan^{-1}(\triangle_y f_{D_m}(i,j))/(\triangle_x f_{D_m}(i,j)) \tag{5.23}$$

$$\theta_{f_C}(i,j) = tan^{-1}(\triangle_y f_{C_m}(i,j))/(\triangle_x f_{C_m}(i,j)) \tag{5.24}$$

$$\theta_{f_L}(i,j) = tan^{-1}(\triangle_y f_{L_m}(i,j))/(\triangle_x f_{L_m}(i,j)) \tag{5.25}$$

where

$$\begin{aligned}
\triangle_x f_{D_m}(i,j) &= f_{D_m}(i+1,j+1) + 2f_{D_m}(i+1,j) + f_{D_m}(i+1,j-1) \\
&\quad - f_{D_m}(i-1,j+1) - 2f_{D_m}(i-1,j) - f_{D_m}(i-1,j-1) \\
\triangle_x f_{C_m}(i,j) &= f_{C_m}(i+1,j+1) + 2f_{C_m}(i+1,j) + f_{C_m}(i+1,j-1) \\
&\quad - f_{C_m}(i-1,j+1) - 2f_{C_m}(i-1,j) - f_{C_m}(i-1,j-1) \\
\triangle_x f_{L_m}(i,j) &= f_{L_m}(i+1,j+1) + 2f_{L_m}(i+1,j) + f_{L_m}(i+1,j-1) \\
&\quad - f_{L_m}(i-1,j+1) - 2f_{L_m}(i-1,j) - f_{L_m}(i-1,j-1)
\end{aligned} \tag{5.26}$$

and

$$\begin{aligned}
\triangle_y f_{D_m}(i,j) &= f_{D_m}(i-1,j+1) + 2f_{D_m}(i,j+1) + f_{D_m}(i+1,j+1) \\
&\quad - f_{D_m}(i-1,j-1) - 2f_{D_m}(i,j-1) - f_{D_m}(i+1,j-1) \\
\triangle_y f_{C_m}(i,j) &= f_{C_m}(i-1,j+1) + 2f_{C_m}(i,j+1) + f_{C_m}(i+1,j+1) \\
&\quad - f_{C_m}(i-1,j-1) - 2f_{C_m}(i,j-1) - f_{C_m}(i+1,j-1) \\
\triangle_y f_{L_m}(i,j) &= f_{L_m}(i-1,j+1) + 2f_{L_m}(i,j+1) + f_{L_m}(i+1,j+1) \\
&\quad - f_{L_m}(i-1,j-1) - 2f_{L_m}(i,j-1) - f_{L_m}(i+1,j-1)
\end{aligned} \tag{5.27}$$

The results after detecting the edges on the images f_{D_m}, f_{C_m}, and f_{L_m} generate images $f_{D_{edge}}$, $f_{C_{edge}}$, and $f_{L_{edge}}$.

5.12 Local Gap Identification and Filling

This operation uses the rules of figure 4.11 to identify and fill in the gaps in the images $f_{D_{edge}}$, $f_{C_{edge}}$, and $f_{L_{edge}}$ to restore all the lost information that is not completely restored by the morphological closing operation. This process produces the images $f_{D_{gaps}}$, $f_{C_{gaps}}$, and $f_{L_{gaps}}$.

5.13 Local Image Enhancement

As a final step, binarized images are produced after performing a median filtering on the following images $f_{D_{gaps}}$, $f_{C_{gaps}}$, and $f_{L_{gaps}}$.

$$\forall(i \times j) \in W_D \times H_D \quad f_{D_{gaps}}(i,j) = Median\{f_{D_{gaps}}(i-k, j-l) \forall k, l \in W\} \tag{5.28}$$

$$\forall(i \times j) \in W_C \times H_C \quad f_{C_{gaps}}(i,j) = Median\{f_{C_{gaps}}(i-k, j-l) \forall k, l \in W\} \tag{5.29}$$

$$\forall(i \times j) \in W_L \times H_L \quad f_{L_{gaps}}(i,j) = Median\{f_{L_{gaps}}(i-k, j-l) \forall k, l \in W\} \tag{5.30}$$

where W_D and H_D are the width and the height of the image $f_{D_{gaps}}$; W_C and H_C are the width and the height of the image $f_{C_{gaps}}$; W_L and H_L are the width and the height of the image $f_{L_{gaps}}$; and W is a suitably chosen window. In this work, a 3×3 window is employed.

5.14 Experimental Results

Four hundred and twenty two bank cheques were used for training and 203 cheques were used for testing the performance of both approaches (the approach that uses the global thresholding technique to eliminate the background and the approach that uses a local thresholding after segmenting the document into various regions of interest). **TABLE III** illustrates the efficiency of the system and **TABLE IV** illustrates the comparative study conducted on the performance analysis of both systems. The performance analysis results were produced following the same approach as that used in Chapter 3 Section 3.8.

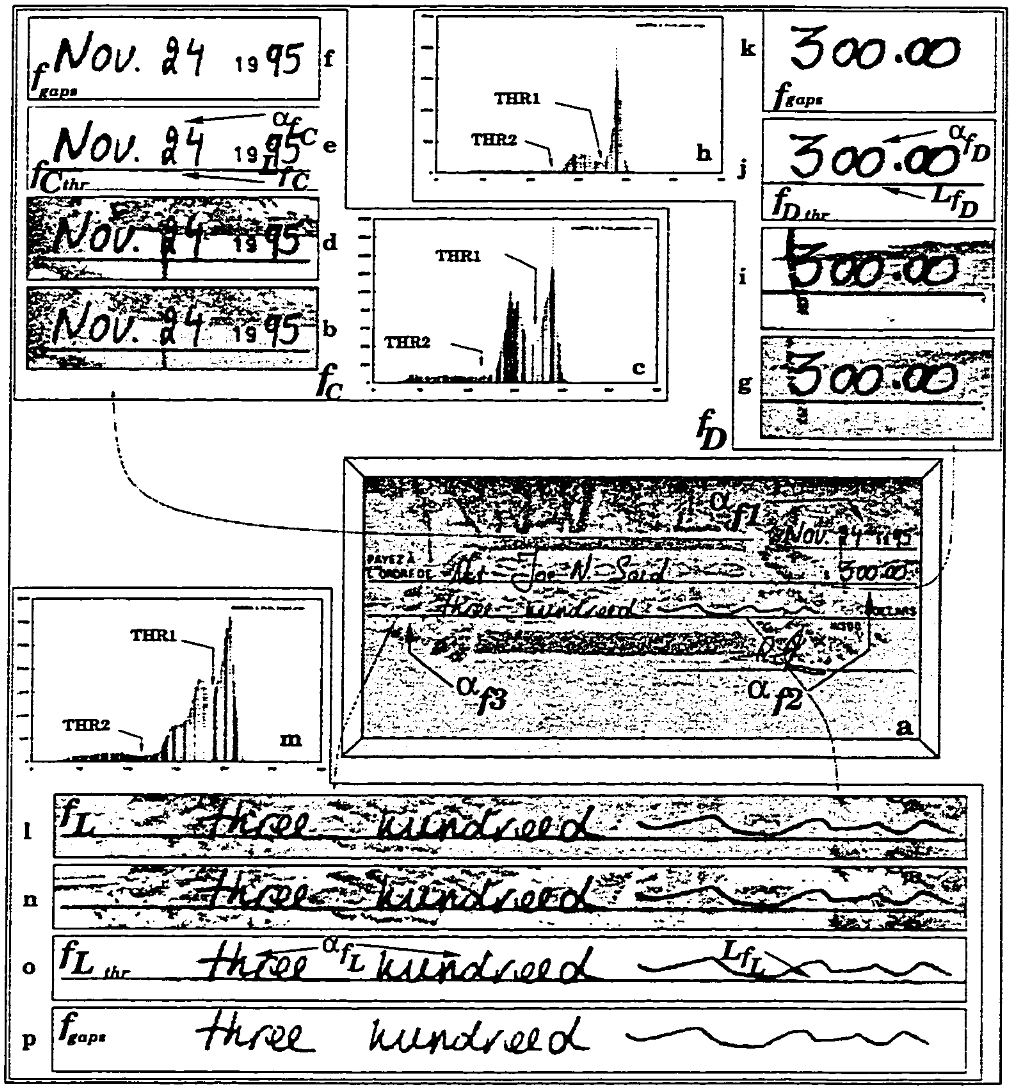

Figure 5.4: Local processing of extracted information.

As an interpretation of **TABLE IV**, line one in **TABLE IV** indicates the accuracy of both systems on 40 BELL cheques whose background is dark and simple and whose foreground is dark thin or dark thick. Figures 5.5, 5.6, 5.7, and 5.8 visually illustrate that the extended approach presented in this Chapter better preserve the luminance and the topological properties of the extracted items.

TABLE III Average Time (sec.)										
Local Processing Time $k \in \{D, C, L\}$										
A:f_{filt} **B:**f_{thr} **C:**f_k **D:**$f_{k_{thr}}$ **E:**$f_{k_{lr}}$ **F:**$f_{k_{clos}}$ **G:**$f_{k_{union}}$ **H:**$f_{k_{edge}}$ **I:**$f_{k_{gaps}}$										
Background/Foreground (f)	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	Total
D S / {t, T} D+	17	2	1	1	1	5	1	7	2	37
D S / {t, T} L+	17	2	1	1	1	5	1	7	3	38
D S / {t, T} L	17	2	1	1	1	5	1	7	6	41
D+ S / {t, T} {D+, L+, L }	17	2	1	1	1	5	1	7	6	41
D C / {t, T} D+	17	5	1	1	1	5	1	7	3	41
D C / {t, T} L+	17	5	1	1	1	5	1	7	4	42
D C / {t, T} L	17	5	1	1	1	5	1	7	6	44
D+ C / {t, T} {D+, L+, L }	17	5	1	1	1	5	1	7	8	46
D C+ / {t, T} D+	17	5	1	1	1	5	1	7	6	44
D C+ / {t, T} L+	17	5	1	1	1	5	1	7	8	46
D C+ / {t, T} L	17	5	1	1	1	5	1	7	8	46
D+ C+ / {t, T} {D+, L+, L }	17	5	1	1	1	5	1	7	8	46

TABLE IV **Testing Set (none of the training images were used during testing)**			
# of images	*Background/Foreground (f) of a given cheque*	*Accuracy of the previous system*	*Accuracy of the new approach*
40	$DS/\{t,T\}D^+$	99%-100%	99%-100%
23	$DS/\{t,T\}L^+$	96%-100%	99%-100%
17	$DS/\{t,T\}L$	88%-100%	95%-100%
31	$D^+S/\{t,T\}\{D^+,L^+,L\}$	76%-87%	90%-94%
25	$DC/\{t,T\}D^+$	99%-100%	99%-100%
26	$DC/\{t,T\}L^+$	94%-100%	99%-100%
14	$DC/\{t,T\}L$	82%-100%	89%-100%
8	$D^+C/\{t,T\}\{D^+,L^+,L\}$	61%-82%	75%-88%
8	$DC^+/\{t,T\}D^+$	99%-100%	99%-100%
3	$DC^+/\{t,T\}L^+$	91%-100%	95%-100%
3	$DC^+/\{t,T\}L$	0%	86%-94%
5	$D^+C^+/\{t,T\}\{D^+,L^+,L\}$	0%	76%-85%

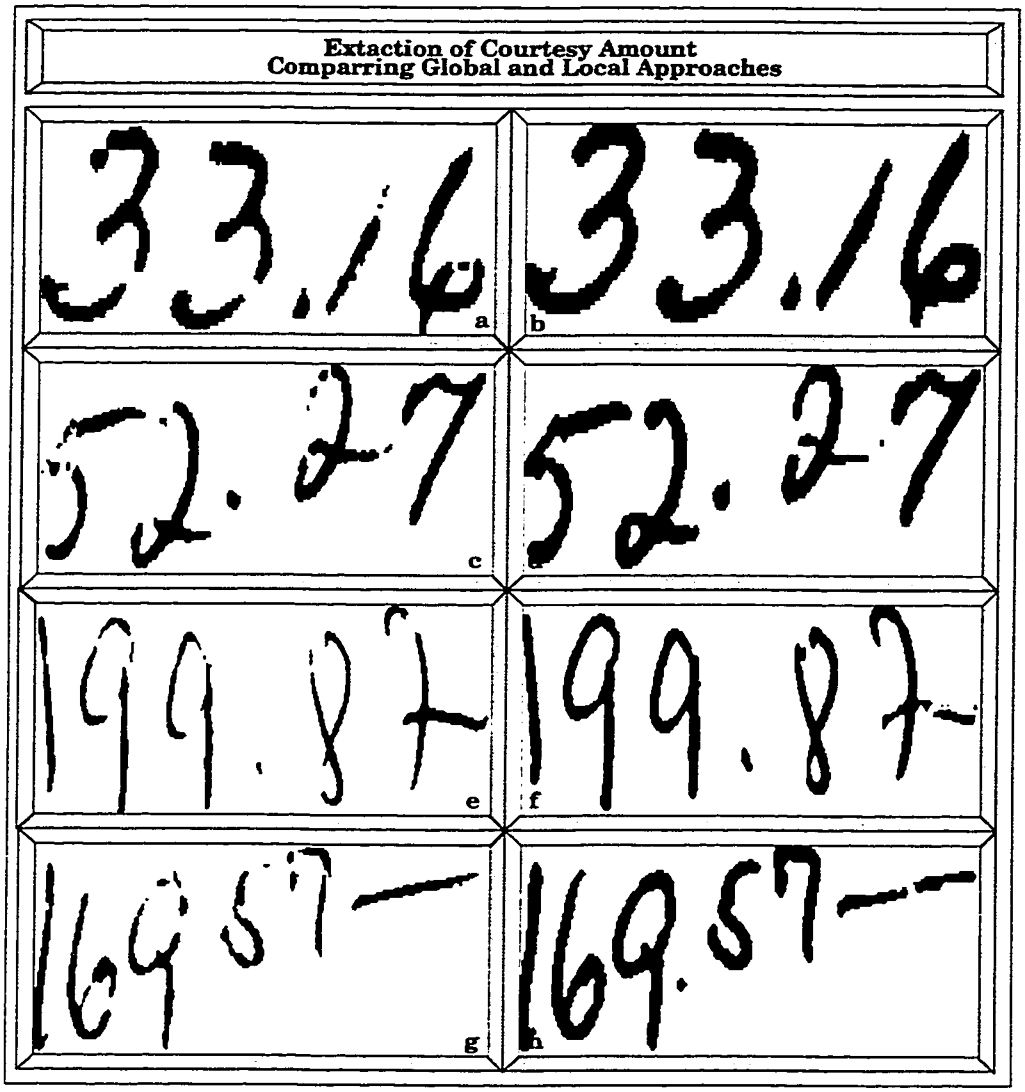

Figure 5.5: Sample output images to illustrate that local thresholding as opposed to global thresholding was able to decrease the distortion of the luminance and to the topological properties of the extracted items.

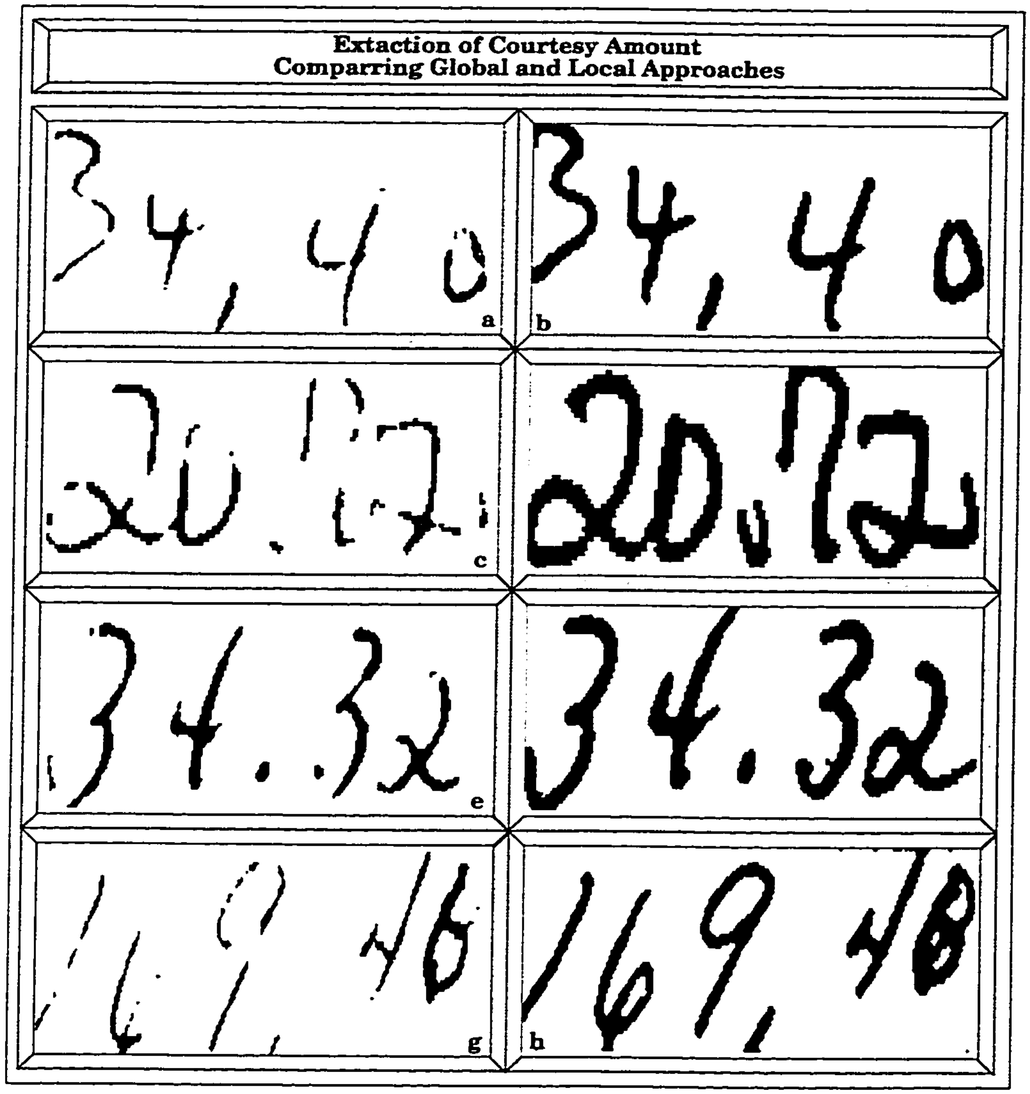

Figure 5.6: (continues) Sample output images to illustrate that local thresholding as opposed to global thresholding was able to decrease the distortion of the luminance and to the topological properties of the extracted items.

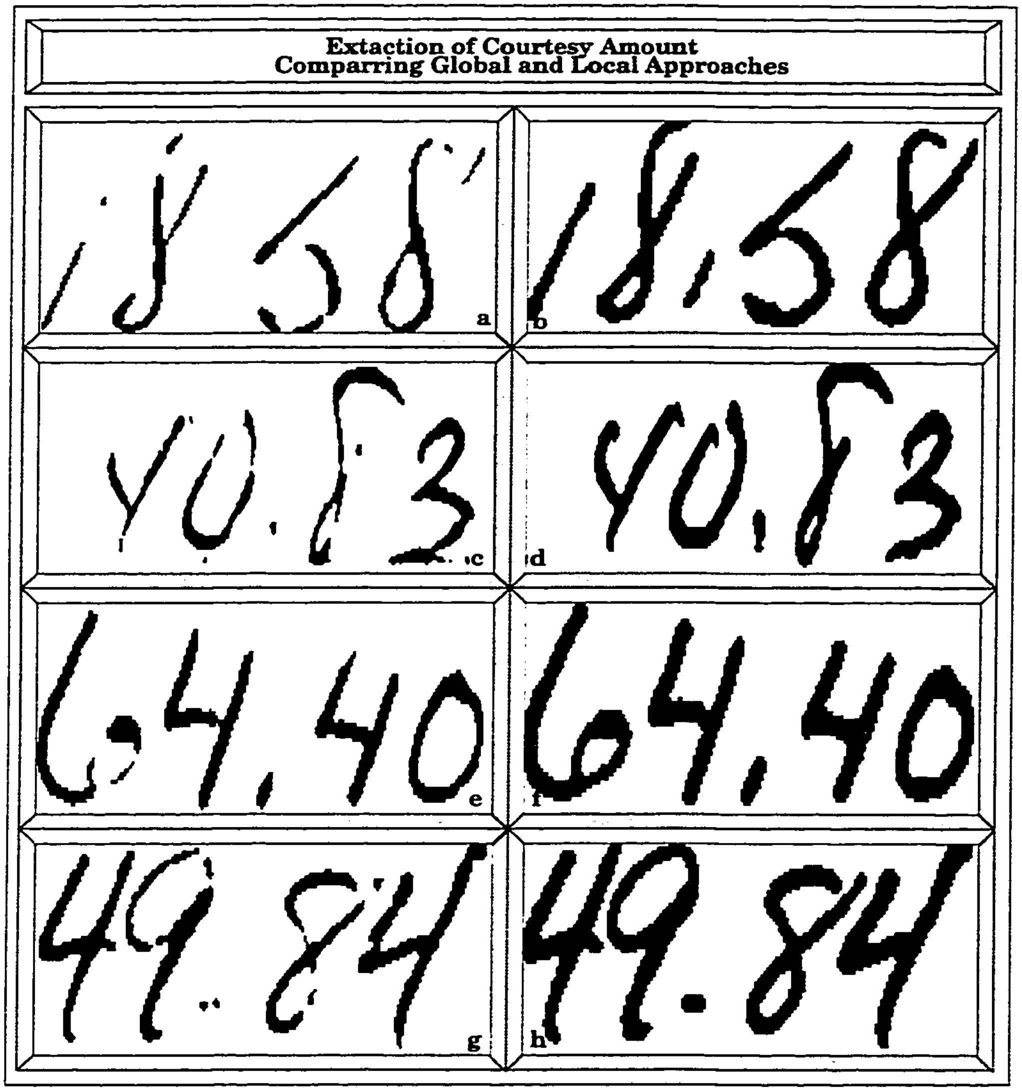

Figure 5.7: (continues) Sample output images to illustrate that local thresholding as opposed to global thresholding was able to decrease the distortion of the luminance and to the topological properties of the extracted items.

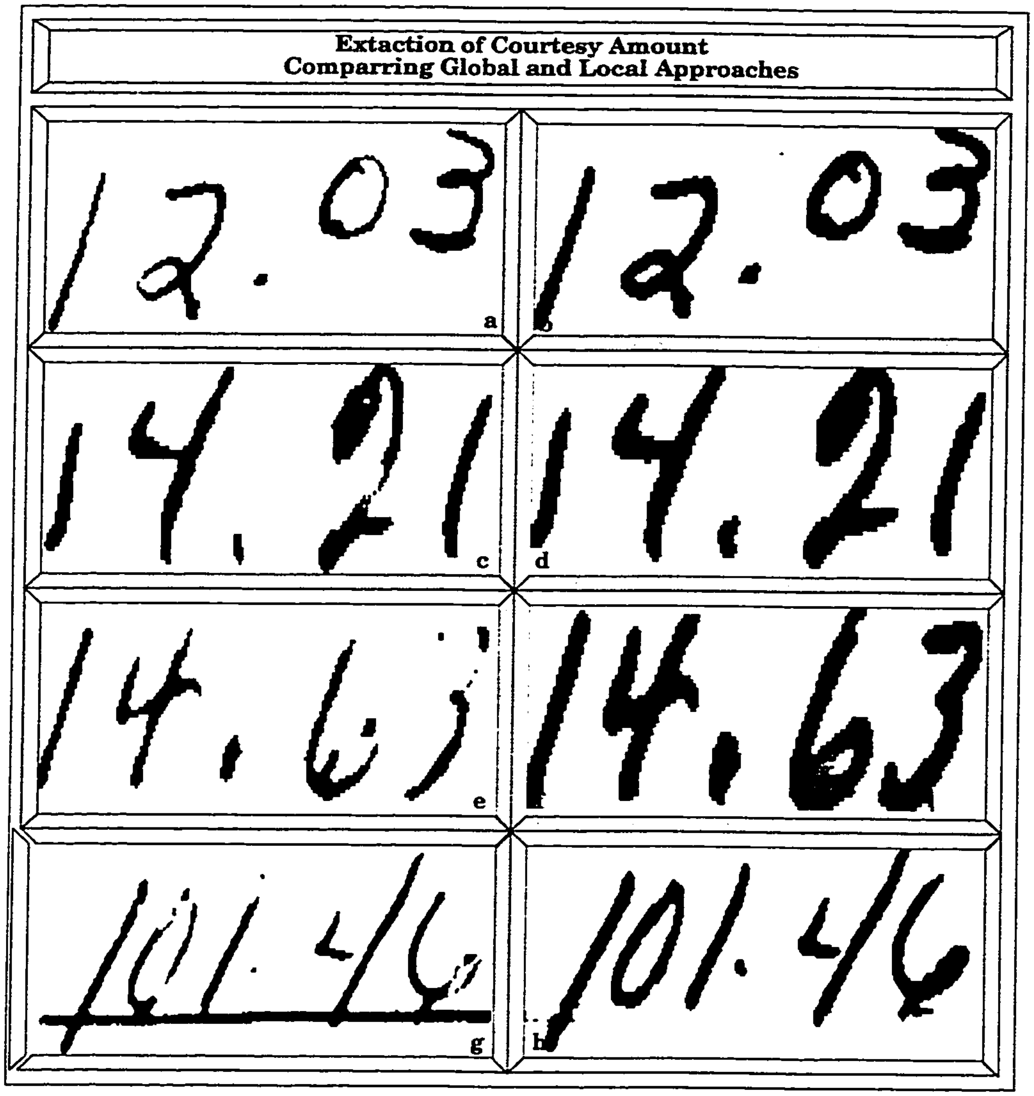

Figure 5.8: (continues) Sample output images to illustrate that local thresholding as opposed to global thresholding was able to decrease the distortion of the luminance and to the topological properties of the extracted items.

5.15 Discussion and Motivation for an Efficient Model

Section 5.1 introduced an extension of the model presented in Section 4.1 which is efficient and reliable in extracting information from cheque images only when the foreground targeted objects are dark and the background objects are lighter. However, for cases when the foreground objects are not dark, global thresholding caused a loss in the handwritten information that should be extracted from the cheque images. As a remedy to this problem, Section 5.1 introduced an extended model that processed the images and extracted the targeted information in a local approach. As a result of this extended method, the problem of having light targeted objects is solved and the system becomes better trained in extracting the targeted information. This resulted in a more robust system that is able to visualize, understand, and extract handwritten information from different types of document images that have various intensities of handwriting against a simple or a complex background.

Looking at efficiency issues one can realize that the systems in Section 4.1 and Section 5.1 have the same execution time, however, the system in Section 5.1 better preserved the topological properties of the extracted information. In a step to extend the system presented in Section 5.1 towards a more efficient model, a new dynamic morphological processing operation is presented. The purpose of the new dynamic morphological operation is to eliminate the need for further topological processing since topological processing is a time consuming operation. In the following Chapter, the new approach will be presented as well as the experimental results. The experimental results achieved by using the dynamic morphological approach advanced the efficiency of the system presented in Chapter 4 by 76.76% and the system presented in Chapter 5 by 35.98% on the average.

Chapter 6

Dynamic Morphological Processing for Better Efficient Extraction

A new dynamic morphological closing operation to increase the efficiency of the system.

6.1 Introduction

In most document analysis and recognition systems, straight lines are considered as one of the basic elements that should be located and eliminated to simplify the process of document analysis and recognition. The superposition or the intersection of different objects of interest found in the same area makes the process of detecting and extracting these line segments a non-trivial task, especially, if the method should preserve the valuable objects of interest that intersect with these lines. In this Chapter, a new and effective approach that detects the existence of line segments and eliminates them with the challenge of preserving the valuable information that intersects these line segments will be introduced. The new approach aims at extending the use of the very well known morphological closing operation which used a fixed structuring element as introduced in Chapters 4 and 5 towards a morphological closing operation that uses a dynamic structuring element. The new approach will restore the handwritten information that intersects the baselines that should be eliminated from the image and reduces the need for further topological processing which is a very time consuming operation.

6.2 Methodology

As mentioned earlier the purpose of the new morphological closing operation is to reduce the need for further topological processing. In fact, the new dynamic morphological operation could be applied in two approaches: a global approach to extend the model presented in Chapter 4 and a local approach to extend the model presented in Chapter 5. Following this line, we will present both approaches. The model presented in Chapter 4 could be extended as Figure 6.1 illustrates. The model presented in Chapter 5 could be extended as Figure 6.2 illustrates. Both approaches will enhance the efficiency of the system.

As stated in Section 6.1, one can observe that the morphological closing operation could, in some cases, be seen as a detector or as a detector and a preserver of the information that intersects the baselines that should be segmented from a document image. In fact, in the cases where $V_\triangle$ has the same orientation as that of the handwriting at the location it intersects the baselines, the morphological closing operation preserves the information at that location as illustrated in Figure 6.4 (c). In such cases, the morphological closing operation is considered as a detector and as a preserver. However, in cases where the orientation of the handwriting and $V_\triangle$ are not the same, the morphological closing operation acts as a detector only and further topological processing is required to completely restore what was missed by the closing operation. Figure 6.4 (a) and (b) illustrate this point.

One good observation to bring out here is that all possible orientations where the handwriting intersects a baseline are finite. Moreover, for each orientation a morphological closing operation using the appropriate structuring element, that has the same orientation as the handwriting when it intersects the baselines, could be made to preserve the particular information at the intersection. Now, combining all these preserved particular information results will completely restore all the information

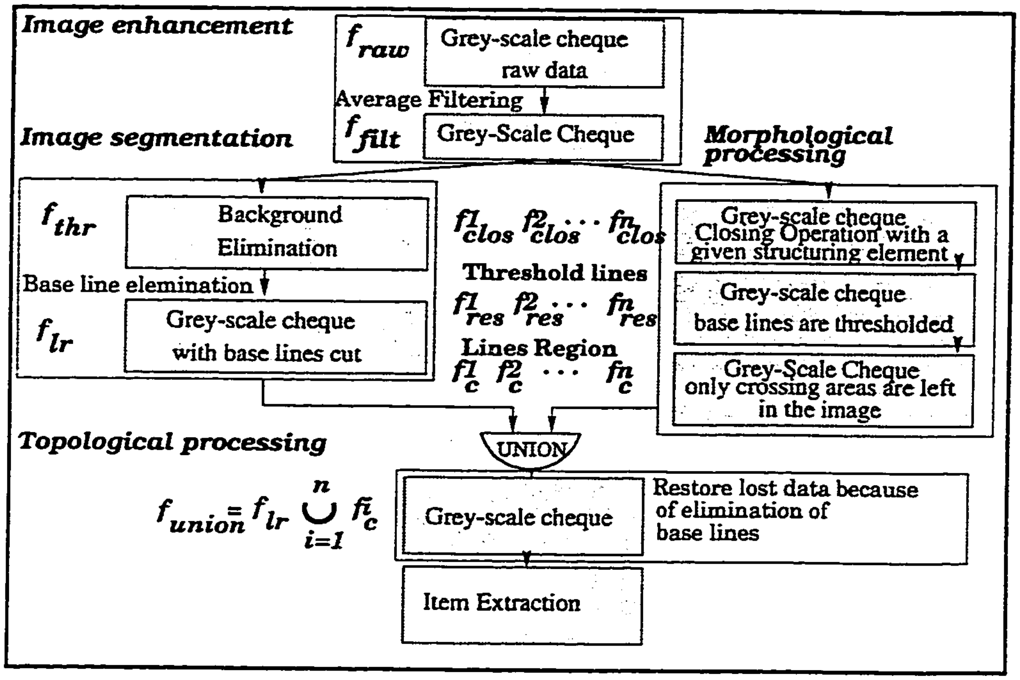

Figure 6.1: System design of the extended global approach.

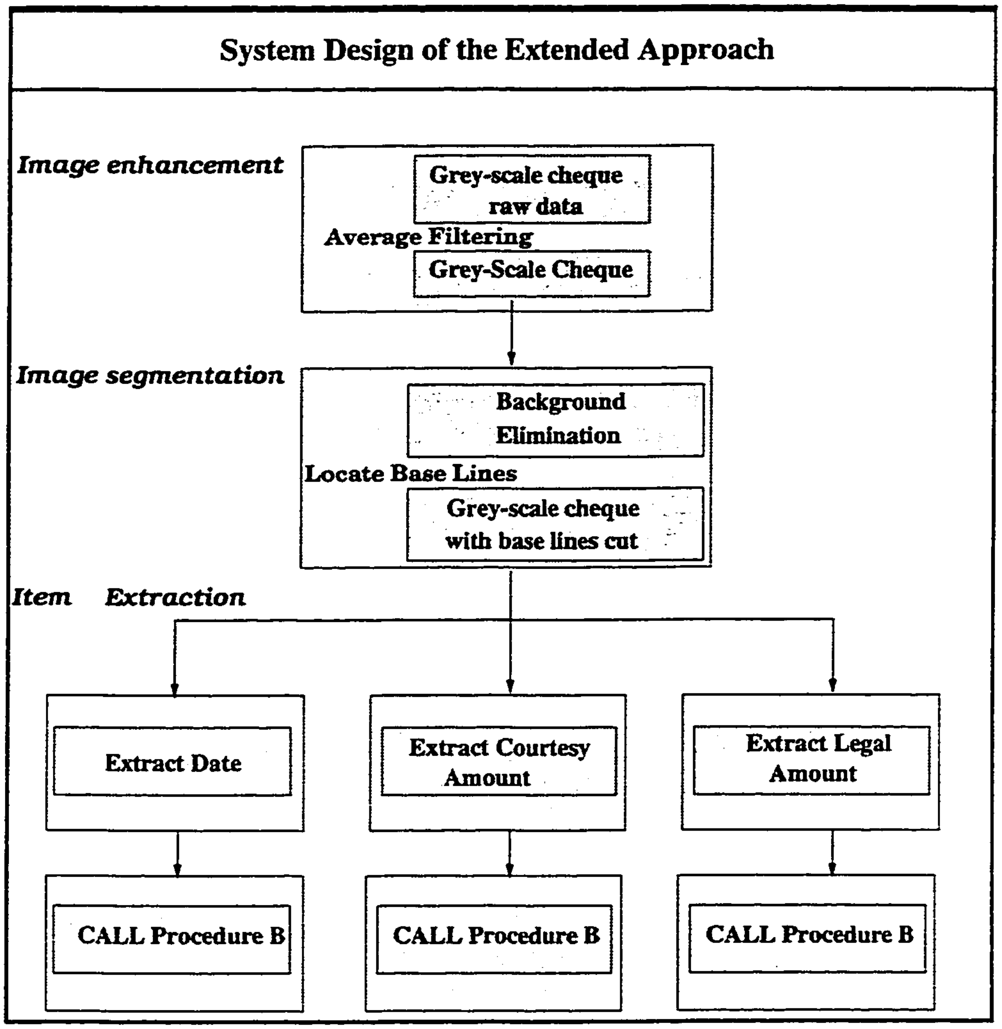

Figure 6.2: System design of the extended local approach.

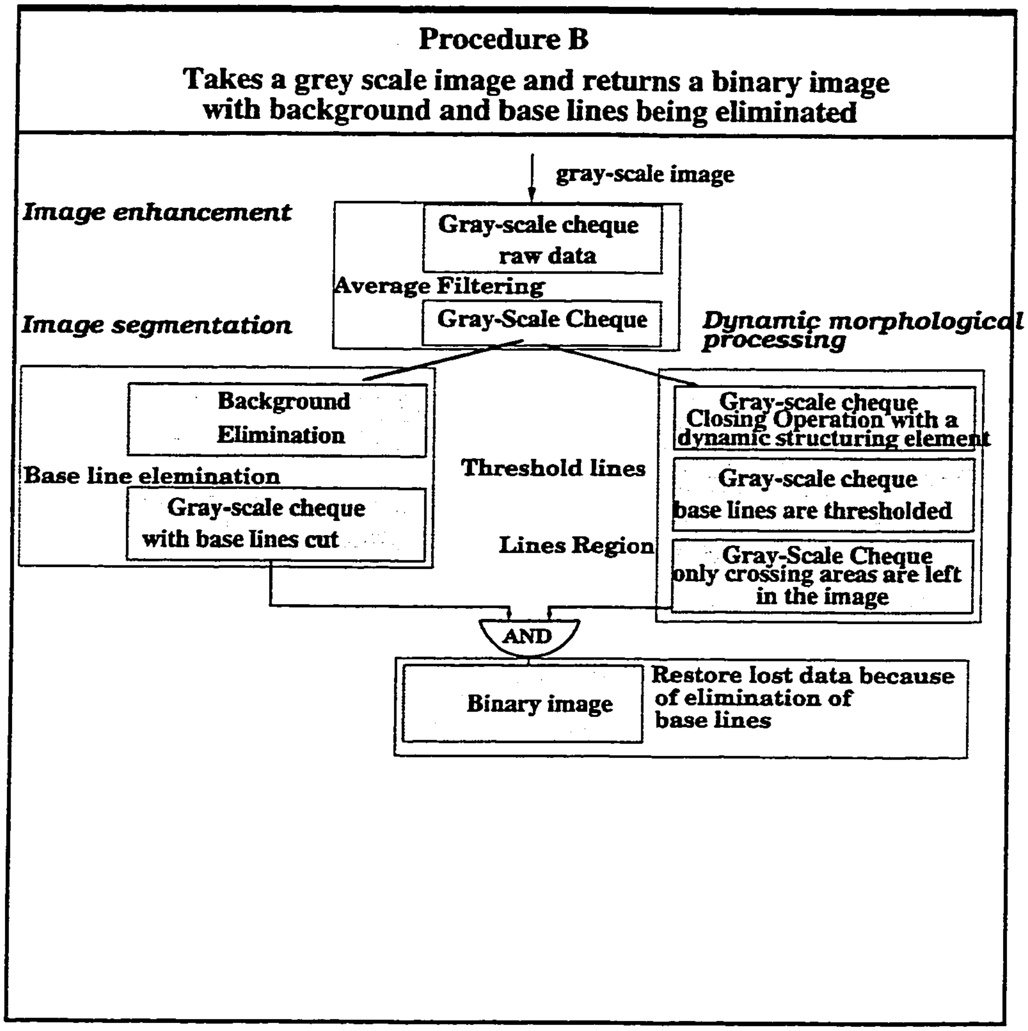

Figure 6.3: Procedure B.

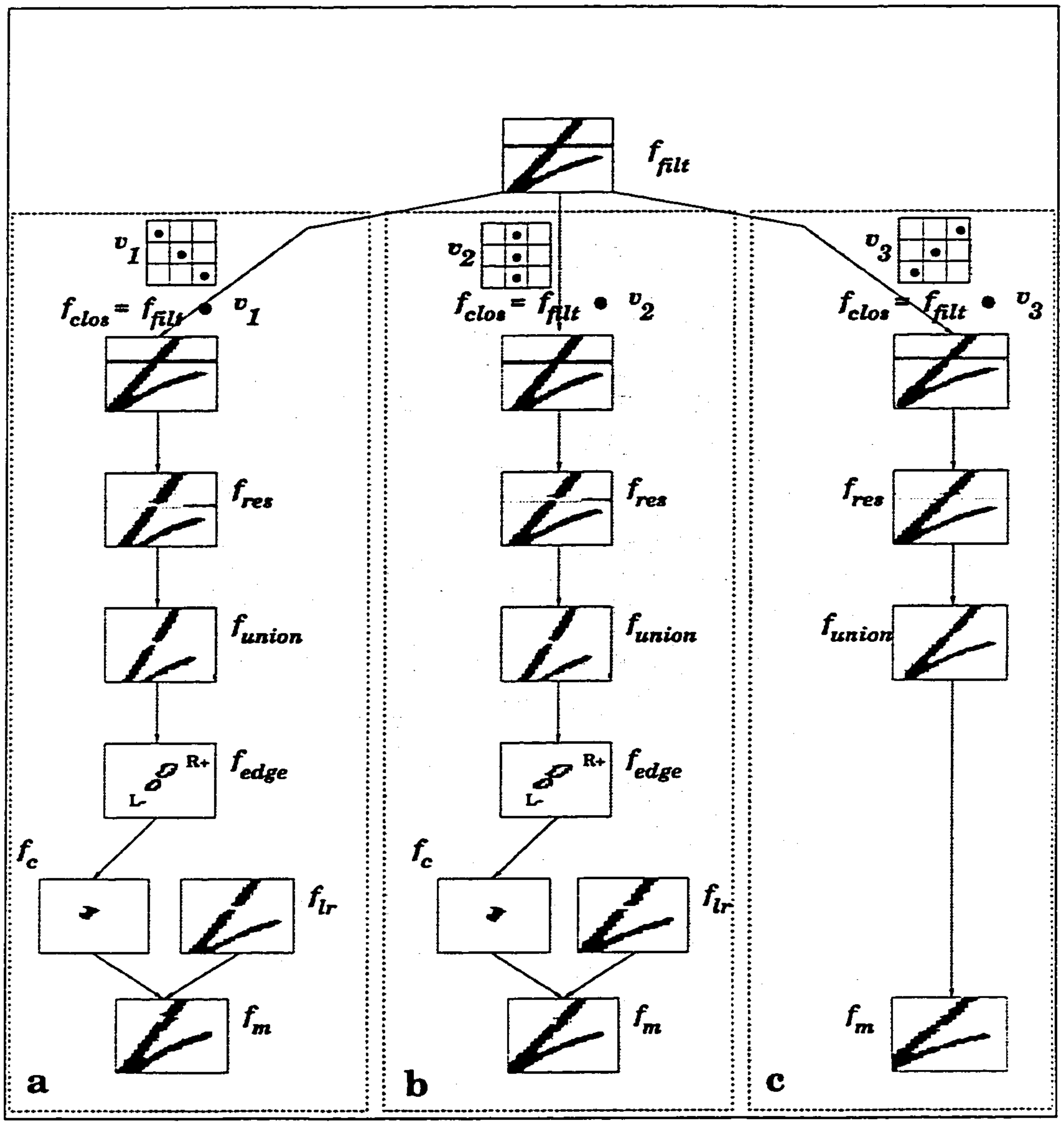

Figure 6.4: Processing using different structuring elements.

that belongs to the intersection of the baselines and the handwriting. Figure 6.5 illustrates this point.

6.3 Global Approach

Referring to Figure 6.1, the process of extracting handwritten data from a given bank cheque image f is extended in the following sense: (1) eliminate the background after performing an image enhancement (average filtering) on the original image f; (2) locate and eliminate the baselines; (3) use the dynamic morphological closing operation that completely restores the information and reduces the need for further topological processing.

In what follows, the system that globally uses a dynamic morphological operation will be formally presented.

6.3.1 Image Enhancement

Given a gray-scale cheque image f, an enhanced image f_{filt} is produced according to Equation 3.9.

6.3.2 Recursive Thresholding

Given f_{filt}, use the approach presented in Section 4.5 to segment f_{filt} as follows:

$$f_{filt} = C_0 \cup C_1 \cup \ldots \cup C_p, \tag{6.1}$$

where $f_{thr} = C_p$, as illustrated in Figures 6.7 and 6.9.

6.3.3 Baseline Elimination

Given f_{thr}, the purpose is to produce an image f_{lr} without the baselines.

$$f_{lr}(i,j) = \begin{cases} 255 & \text{if } f_{thr}(i,j) \in L \\ f_{thr}(i,j) & otherwise \end{cases} \tag{6.2}$$

where $f_{lr}(i,j)$ represents the image after the removal of line L that represents a recorded baseline.

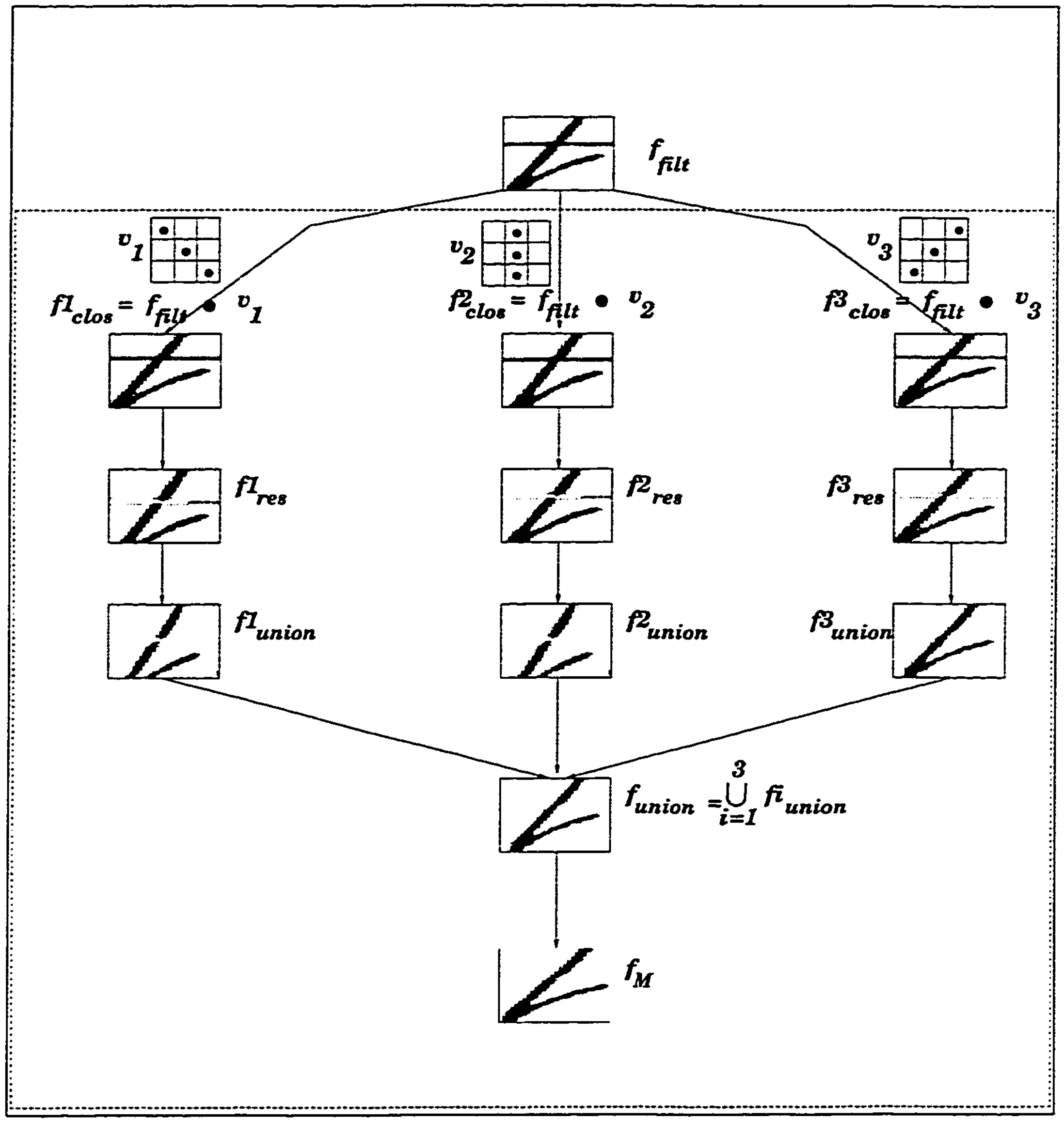

Figure 6.5: A sample process to illustrate the new approach.

6.3.4 Morphological Processing Using a Dynamic Structuring Element

Knowing the location of each baseline, one can perform a morphological closing operation on and around these baselines to reduce the processing time. Now, given f_{filt}, produce the images $f_{k'_{clos}}$, where $k' \in \{1, \ldots, K'\}$, as follows:

$$f_{k'_{clos}} \bullet V_{k'} = (f_{filt} \oplus V_{k'}) \ominus V_{k'} \tag{6.3}$$

where

$$(f_{filt} \ominus V_{k'})(i,j) = min\{f_{filt}(i+k, j+l) \mid (k,l) \in V_{\triangle_{k'}}\} \tag{6.4}$$

$$(f_{filt} \oplus V_{k'})(i,j) = max\{f_{filt}(i+k, j+l) \mid (k,l) \in V_{\triangle_{k'}}\} \tag{6.5}$$

and $V_{k'}$ is a structuring element and $V_{\triangle_{k'}}$ is the set of coordinate points within $V_{k'}$. Figure 6.5 illustrates this.

6.3.5 Restoring Lost Information

Knowing each baseline's position and length, we can now process the morphological images $f_{k'_{clos}}$, where $k' \in \{1, \ldots, K'\}$, as in Figure 6.5, by thresholding baseline pixel values in each $f_{k'_{clos}}$ according to the mean $M_{k'}$ of each recorded line L in $f_{k'_{clos}}$ as follows:

$$f_{k'_{res}}(i,j) = \begin{cases} f_{k'_{clos}}(i,j) & \text{if } f_{k'_{clos}}(i,j) \leq M_{k'} \\ 255 & otherwise \end{cases} \tag{6.6}$$

$$f_{k'_{union}}(i,j) = \begin{cases} f_{k'_{res}}(i,j) & \text{if } f_{k'_{res}}(i,j) \in L \\ f_{lr}(i,j) & otherwise \end{cases} \tag{6.7}$$

$$f_{union} = \bigcup_{k' \in \{1,\ldots,K'\}} f_{k'_{union}} \tag{6.8}$$

6.4 Local Approach

Referring to Figure 6.2, the process of extracting handwritten data from a given bank cheques image f is extended in the following sense: (1) eliminate the background after performing an image enhancement (average filtering) on the original image f: (2) locate the baselines and extract the required items such as date, courtesy amount and legal amount from the original image; (3) and finally, according to Figure 6.2, process locally the extracted items using the dynamic morphological closing operation that completely restores the information at the intersection between the lines to be eliminated and the handwritten information.

In what follows, the system that locally uses a dynamic morphological operation will be formally presented.

6.4.1 Image Enhancement

Given a gray-scale cheque image f, an enhanced image f_{filt} is produced as in Section 6.3.1.

6.4.2 Global Recursive Thresholding

Given f_{filt}, an image f_{thr} is produced as in Section 6.3.2, as illustrated in Figures 6.7 and 6.9.

6.4.3 Determination and Early Extraction of Baselines

As Section 5.5 illustrates, one can determine $L_s = \{L_{s_1}, L_{s_2}, L_{s_3}\}$ the finite set of baselines, representing the date baseline, the courtesy amount baseline, and the legal amount baseline. Moreover, the matrix M that describes the location description tuples can be determined to extract the set $\alpha_f = \{f_D, f_C, f_L\}$ which represents the finite set of filled data. As illustrated in Figure 5.3 (c), f_D, f_C, and f_L are the extracted images that will be enhanced as in Section 5.6 to produce the images $f_{k_{filt}}$ $\forall k \in \{D, C, L\}$.

6.4.4 Local Recursive Thresholding

Locally recursive thresholding as presented in Section 4.5 can be used for background elimination of the gray-scale images $f_{k_{filt}}$ $\forall k \in \{D, C, L\}$ to produce the images $f_{k_{thr}} = k_p$ $\forall k \in \{D, C, L\}$.

6.4.5 Local Baseline Elimination

This operation determines the baselines L_{s_D}, L_{s_C}, and L_{s_L} in the images $f_{D_{thr}}$, $f_{C_{thr}}$, and $f_{L_{thr}}$ respectively and eliminates them to produce the images $f_{D_{lr}}$, $f_{C_{lr}}$, and $f_{L_{lr}}$ as illustrated in Section 5.8.

6.4.6 Dynamic Morphological Processing

Knowing the location of the baselines L_{s_D}, L_{s_C}, and L_{s_L} in the images $f_{D_{thr}}$, $f_{C_{thr}}$, and $f_{L_{thr}}$ respectively, a dynamic morphological closing operation is performed on and around the baselines L_{s_D}, L_{s_C}, and L_{s_L} in the images $f_{D_{filt}}$, $f_{C_{filt}}$, and $f_{L_{filt}}$ to reduce the processing time.

Formally, $\forall k \in \{D, C, L\}$ consider $f_{k_{filt}}$, and produce the images $f_{k{k'}_{clos}}$ where $k' \in \{1, \ldots, K'\}$, where $K' = 3$ is the number of different structuring elements assumed, as follows:

$$f_{D{k'}_{clos}} \bullet V_{k'} = (f_{D_{filt}} \oplus V_{k'}) \ominus V_{k'} \tag{6.9}$$

$$f_{C{k'}_{clos}} \bullet V_{k'} = (f_{C_{filt}} \oplus V_{k'}) \ominus V_{k'} \tag{6.10}$$

$$f_{L{k'}_{clos}} \bullet V_{k'} = (f_{L_{filt}} \oplus V_{k'}) \ominus V_{k'} \tag{6.11}$$

where $V_{k'}$ is a structuring element and $V_{\triangle_{k'}}$ is the set of coordinate points within $V_{k'}$, as illustrated in Figures 6.7 and 6.9.

6.4.7 Restoring Lost Information

Knowing the position and length of the baselines L_{s_D}, L_{s_C}, and L_{s_L} in the images $f_{D{k'}_{clos}}$, $f_{C{k'}_{clos}}$, and $f_{L{k'}_{clos}}$ respectively where $k' \in \{1, \ldots, K'\}$, we can process these morphological images by thresholding baseline pixel values, in $f_{D{k'}_{clos}}$, $f_{C{k'}_{clos}}$, and $f_{L{k'}_{clos}}$ according to the mean $M_{Dk'}$, $M_{Ck'}$, and $M_{Lk'}$ of the lines L_{s_D}, L_{s_C}, and L_{s_L} as follows:

$$f_{D{k'}_{res}}(i,j) = \begin{cases} f_{D{k'}_{clos}}(i,j) & \text{if } f_{D{k'}_{clos}}(i,j) \leq M_{Dk'} \\ 255 & otherwise \end{cases}$$

$$f_{D{k'}_{union}}(i,j) = \begin{cases} f_{D{k'}_{res}}(i,j) & \text{if } f_{D{k'}_{res}}(i,j) \in L_{s_D} \\ f_{Dl_r}(i,j) & otherwise \end{cases} \tag{6.12}$$

$$f_{D_{union}} = \bigcup_{k' \in \{1,\ldots,K'\}} f_{D{k'}_{union}} \tag{6.13}$$

$$f_{C{k'}_{res}}(i,j) = \begin{cases} f_{C{k'}_{clos}}(i,j) & \text{if } f_{C{k'}_{clos}}(i,j) \leq M_{Ck'} \\ 255 & otherwise \end{cases}$$

$$f_{C{k'}_{union}}(i,j) = \begin{cases} f_{C{k'}_{res}}(i,j) & \text{if } f_{C{k'}_{res}}(i,j) \in L_{s_C} \\ f_{Cl_r}(i,j) & otherwise \end{cases} \tag{6.14}$$

$$f_{C_{union}} = \bigcup_{k' \in \{1,\ldots,K'\}} f_{C{k'}_{union}} \tag{6.15}$$

$$f_{L{k'}_{res}}(i,j) = \begin{cases} f_{L{k'}_{clos}}(i,j) & \text{if } f_{L{k'}_{clos}}(i,j) \leq M_{Lk'} \\ 255 & otherwise \end{cases}$$

$$f_{L{k'}_{union}}(i,j) = \begin{cases} f_{L{k'}_{res}}(i,j) & \text{if } f_{L{k'}_{res}}(i,j) \in L_{s_L} \\ f_{Ll_r}(i,j) & otherwise \end{cases} \tag{6.16}$$

$$f_{L_{union}} = \bigcup_{k' \in \{1,\ldots,K'\}} f_{L{k'}_{union}} \tag{6.17}$$

6.5 Experimental Results

This approach is tested on different types of document images such as bank cheques in Figures 6.6 and 6.7 and other business documents where there is a need to detect

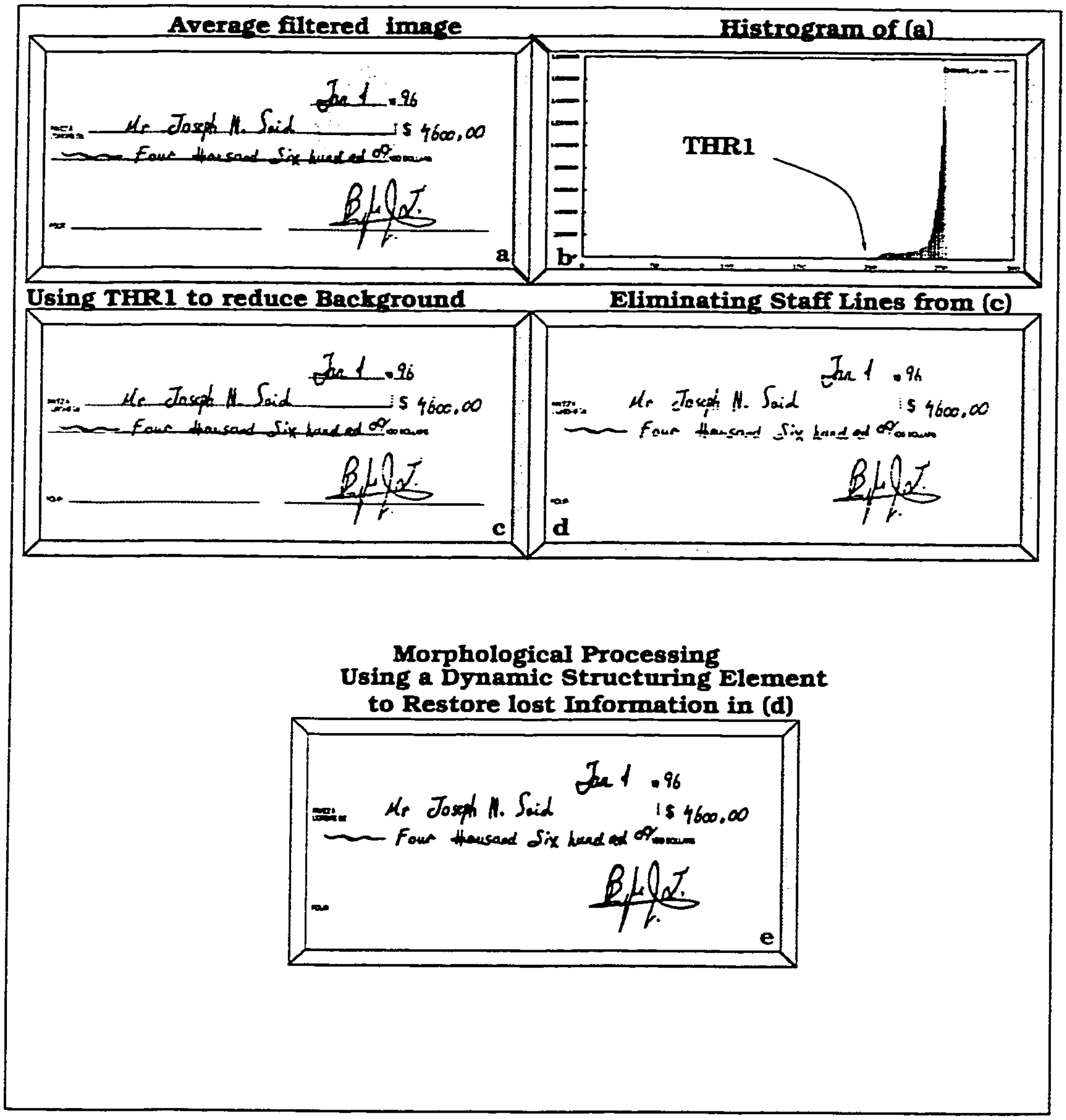

Figure 6.6: A sample cheque processing using the extended approach.

and eliminate baselines that intersect valuable information in Figures 6.8 and 6.10, as well as music notation documents in Figure 6.9.

Five hundred real life bank cheques were used to train the systems presented in Section 6.3 and Section 6.4 and 229 different cheques were used to test its performance.

TABLE V and **TABLE VI** summarize the performance of the systems as presented in Section 6.3 and Section 6.4 respectively. Moreover, **TABLE VII** compares the performance of the new approaches and the one presented in Chapter 4. From [28] it is clear that the system requires more time as the complexity of the document increases because further topological processing is required to preserve the information loss after the elimination of the baselines.

Compared with the experimental results of Chapter 4, **TABLE IV** shows a remarkable increase of 76.76% of the average processing time in the efficiency of the new approach of Section 6.3. Moreover, **TABLE VI** shows another important advancement of 35.98% in the efficiency of the system described in Section 6.4 when compared with the efficiency of the system presented in Chapter 5. In fact, this is true because the new approach which presented a new morphological closing operation that uses a dynamic structuring element acts as a feature detection and as an information preservation operation that eliminated the need for further topological processing.

TABLE V	Average Time (sec.)					
Applying dynamic morphology globally ($K' = 3$)						
Background/Foreground (f)	f_{filt}	f_{thr}	f_{lr}	f_{clos}	f_{union}	f_m
D S / {t, T} D+	17	2	1	$2K'$	K'	2
D S / {t, T} L+	17	2	1	$2K'$	K'	2
D S / {t, T} L	17	2	1	$2K'$	K'	2
D+ S / {t, T} {D+, L+, L }	17	2	1	$2K'$	K'	2
D C / {t, T} D+	17	5	1	$2K'$	K'	2
D C / {t, T} L+	17	5	1	$2K'$	K'	2
D C / {t, T} L	17	5	1	$2K'$	K'	2
D+ C / {t, T} {D+, L+, L }	17	5	1	$2K'$	K'	2
D C+ / {t, T} D+	17	5	1	$2K'$	K'	2
D C+ / {t, T} L+	17	5	1	$2K'$	K'	2
D C+ / {t, T} L	17	5	0	0	0	0
D+ C+ / {t, T} {D+, L+, L }	17	5	0	0	0	0

TABLE VI	Average Time (sec.)						
Applying dynamic morphology locally ($k' = 3$)							
Background/Foreground (f)	f_{filt}	f_{thr}	f_k	$f_{k_{thr}}$	f_{kl_r}	$f_{k_{k'clos}}$	$f_{k_{union}}$
D S / {t, T} D+	17	2	1	1	1	K'	1
D S / {t, T} L+	17	2	1	1	1	K'	1
D S / {t, T} L	17	2	1	1	1	K'	1
D+ S / {t, T} {D+, L+, L }	17	2	1	1	1	K'	1
D C / {t, T} D+	17	5	1	1	1	K'	1
D C / {t, T} L+	17	5	1	1	1	K'	1
D C / {t, T} L	17	5	1	1	1	K'	1
D+ C / {t, T} {D+, L+, L }	17	5	1	1	1	K'	1
D C+ / {t, T} D+	17	5	1	1	1	K'	1
D C+ / {t, T} L+	17	5	1	1	1	K'	1
D C+ / {t, T} L	17	5	0	1	1	K'	1
D+ C+ / {t, T} {D+, L+, L }	17	5	0	1	1	K'	1

TABLE VII illustrates the fact that the new approach saved a large amount of processing time as the complexity of the document increases. This, in fact, highlights an important phenomenon of such an approach especially in systems that deal with large document images like musical notations and map documents where detection and extraction of baselines with the challenge of preserving the information that intersects these lines is a very important issue to consider.

In **TABLE VII**, T_1 is the total processing time of the system in Chapter 4 and T_2 is the total processing time of the system presented in Section 6.3, and P is the percentage of saving in the processing time between these systems. Moreover, T_3 is the total processing time of the system in Chapter 5 and T_4 is the total processing time of the system presented in Section 6.4, and P' is the percentage of saving in the processing time between these systems.

TABLE VII	Average Time (sec.)					
Background/Foreground (f)	T_1	T_2	P	T_3	T_4	P'
D S / {t, T} D+	129	31	75.97	37	26	29.73
D S / {t, T} L+	134	31	76.87	38	26	31.58
D S / {t, T} L	139	31	77.70	41	26	36.58
D+ S / {t, T} {D+, L+, L }	143	31	78.32	41	26	36.58
D C / {t, T} D+	135	34	74.81	41	28	31.71
D C / {t, T} L+	139	34	75.54	42	28	36.36
D C / {t, T} L	145	34	76.55	44	28	39.13
D+ C / {t, T} {D+, L+, L }	153	34	77.78	46	28	36.36
D C+ / {t, T} D+	145	34	76.55	44	28	39.13
D C+ / {t, T} L+	151	34	77.48	46	28	36.36
D C+ / {t, T} L	0	0	0	46	28	39.13
D+ C+ / {t, T} {D+, L+, L }	0	0	0	46	28	39.13

6.6 Discussion and Motivation to Increase the Throughput of the System

Chapter 4 introduced a novel approach for processing the gray-scale images of bank cheques towards the extraction of specific fields located inside these documents. Chapter 5 is presented to increase the productivity and Chapter 6 introduced a very powerful dynamical morphological processing technique that increased the efficiency of the system and eliminated the need for further topological processing.

As a step to increase the throughput of the system, it is necessary to develop intelligent techniques to extract the targeted information for the gray-scale bank cheques when the background is very dark and contains multi-objects that should be segmented against a lighter handwritten information. As pointed out in **TABLE I - TABLE V**, the systems accuracy is 0% in such cases. In the following Chapter, a new bottom-up approach that tackled this problem is presented.

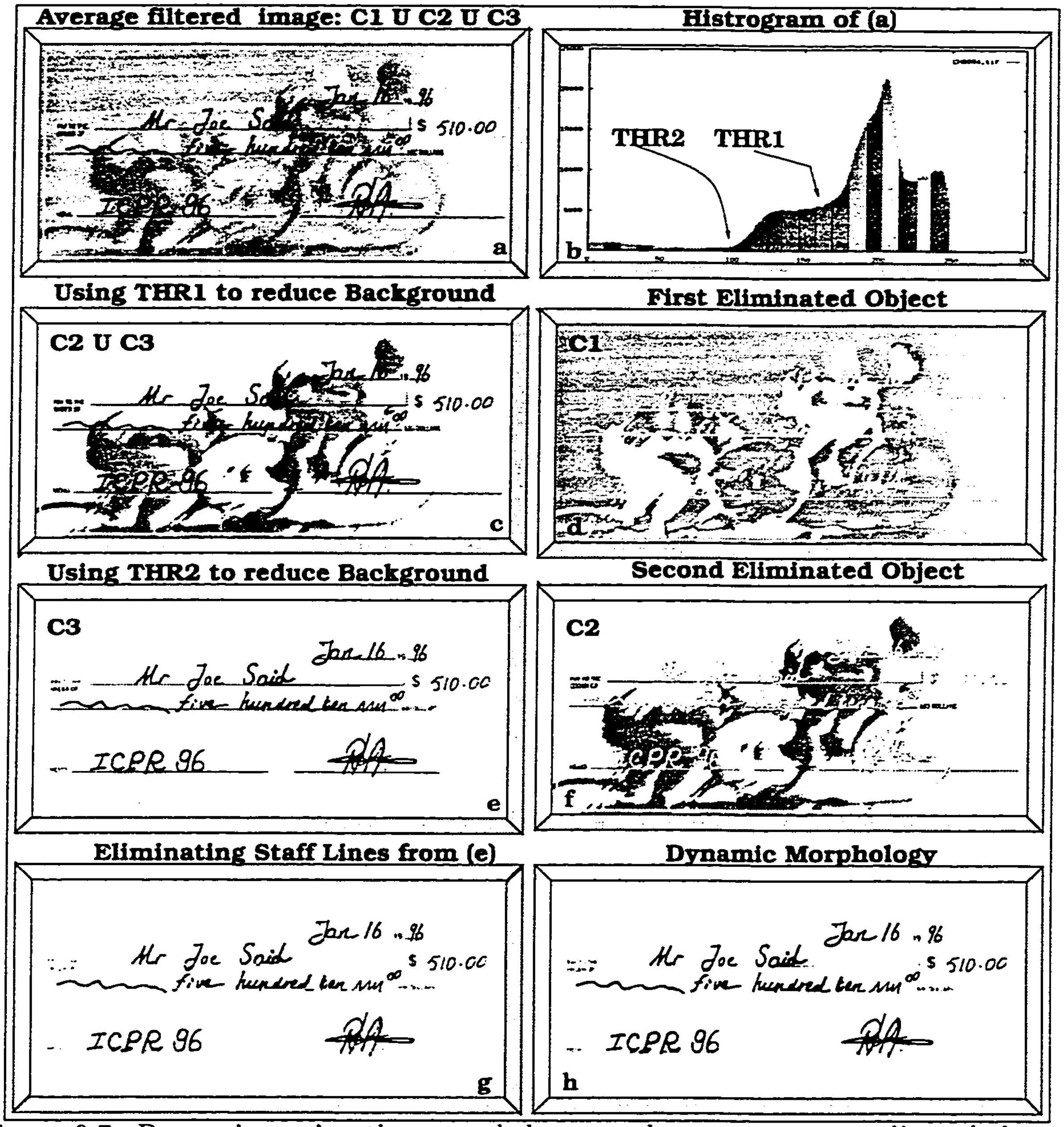

Figure 6.7: Processing using the extended approach on a more complicated cheque image.

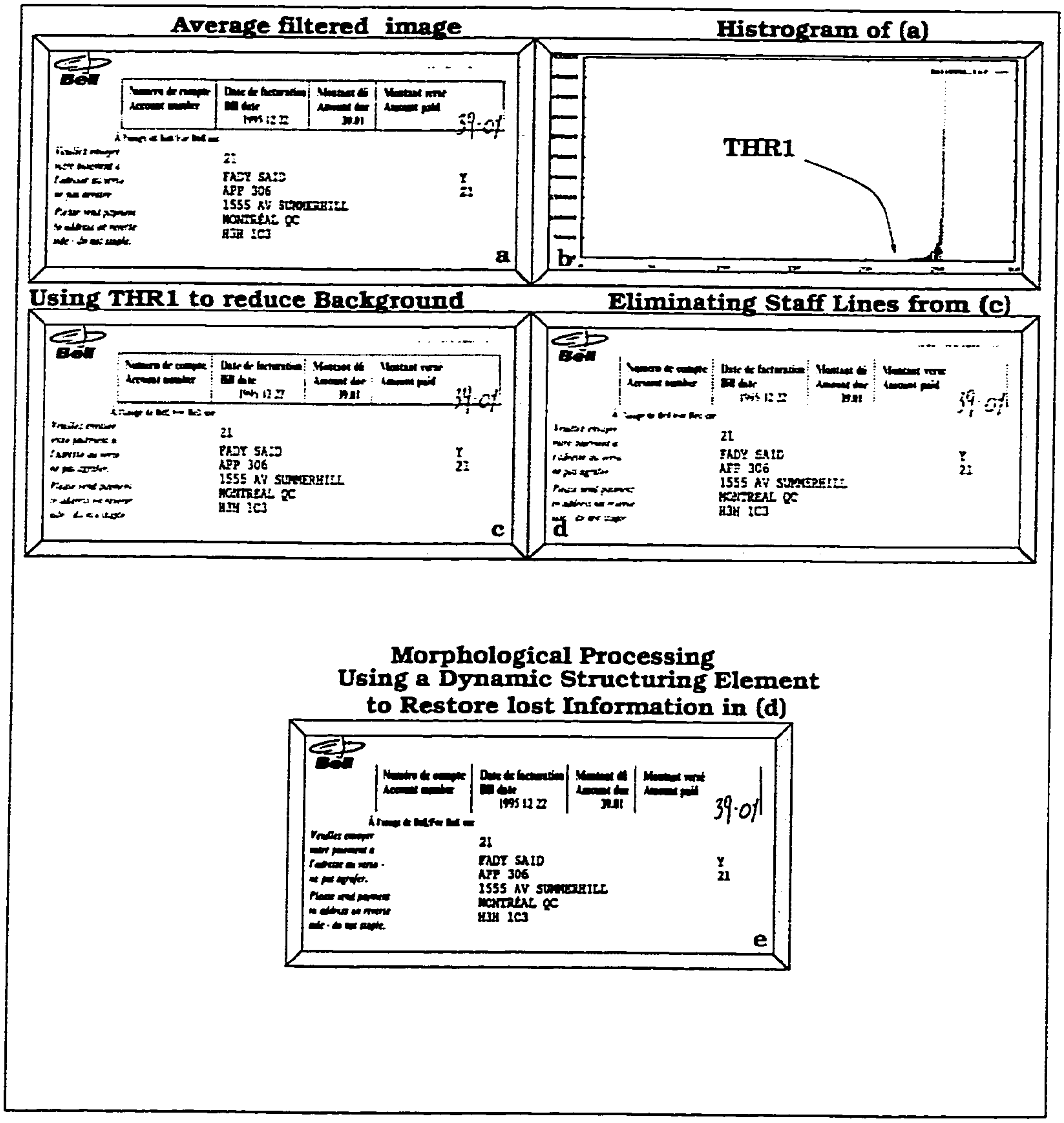

Figure 6.8: A sample document processing using the extended approach.

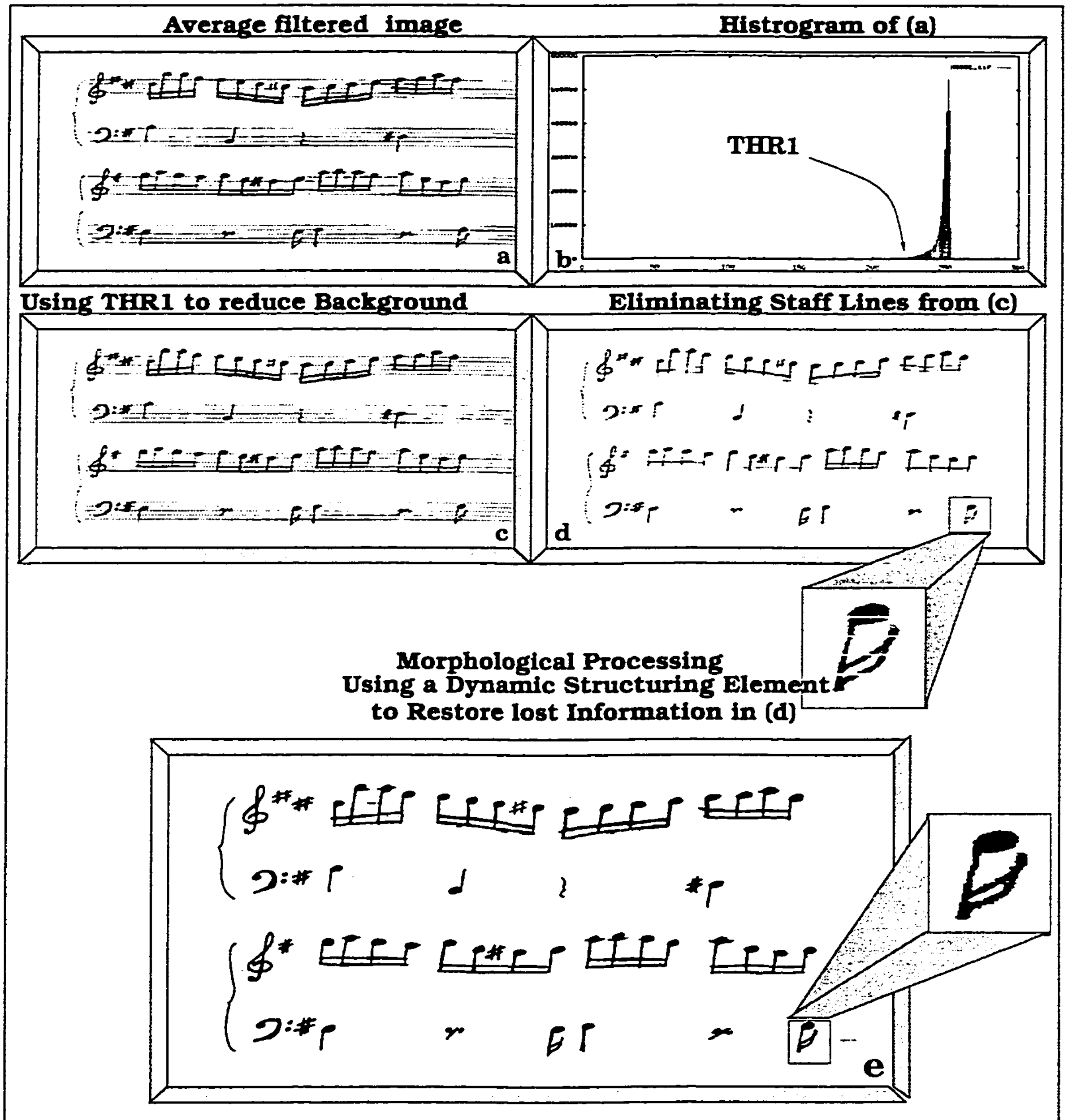

Figure 6.9: Another document processing using the extended approach.

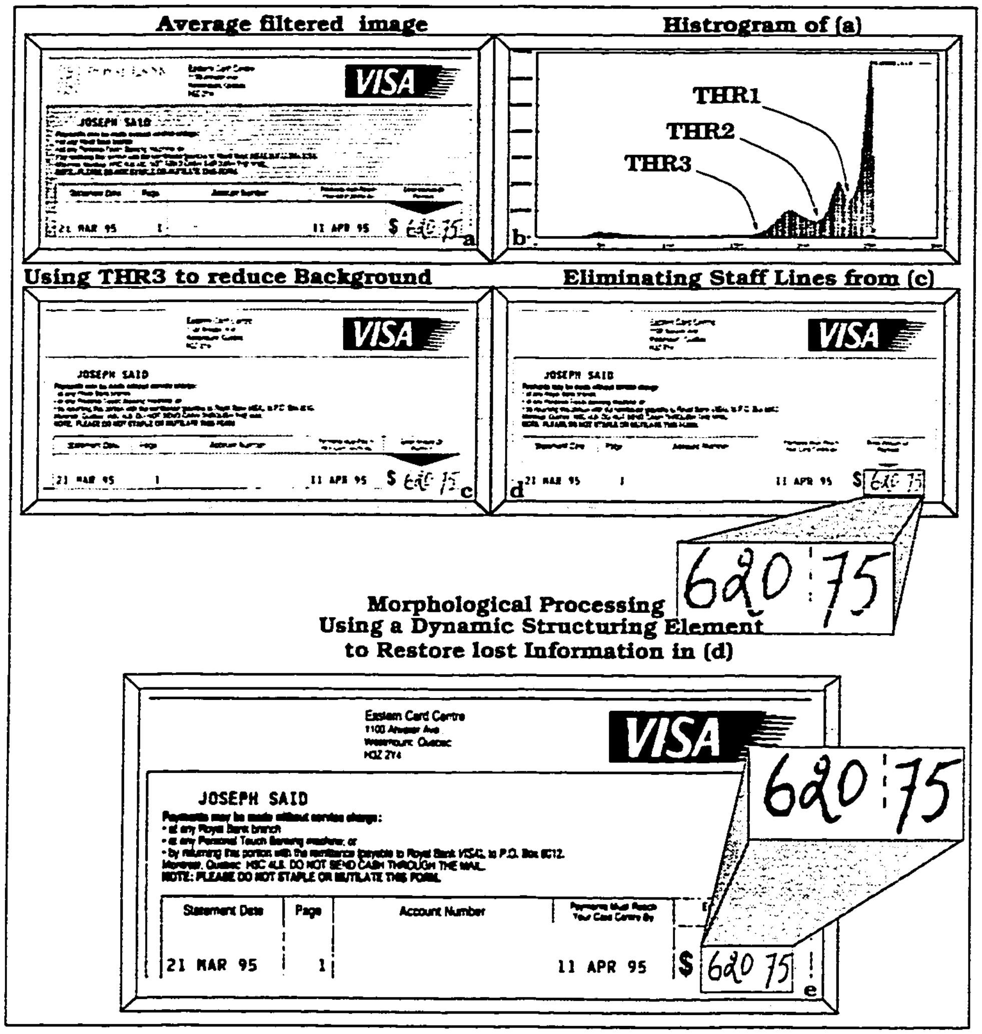

Figure 6.10: Another document processing using the extended approach.

Chapter 7

A Bottom-up Approach for Automatic Item Extraction from Cheque Images

A bottom-up approach.

7.1 Automatic Extraction of Baselines and Data from Images

In this Section we will briefly introduce the work done in [113], where a new bottom-up approach to extract data from cheque images is proposed. The new approach is based on the a-priori information about the position of data on cheques; this a-priori information will help in the determination of printed baselines on cheques and facilitates the process of layout understanding to properly extract different items from the cheque images. This approach viewed document images starting by the pixel value in a step to distinguish between the pixels that belong to the background and the pixels that belong to the foreground. Having recognized the foreground objects. the system recognizes the baselines that must be eliminated and extracts the required items. The experimental results performed on the CENPARMI real life testing bank cheque images obtained from Bell Quebec showed that the system is efficient and robust.

7.2 Methodology

After scanning the cheque image and converting it to a grey image, as shown in Figure 7.1 (a), preprocessing is applied to produce two more images, (1) the image of candidate edge points and, (2) the image of edge directions. Based on the two new images and the original grey scale image, baselines on the cheque image are located and properly extracted. Then, using apriori information about the layout of Canadian cheques the system can locate and extract the appropriate baselines from the scanned cheque image. This will help to get a clear description of the cheque by analyzing the extracted baselines. Once the layout analysis is performed, the system proceeds to extract the legal amount, courtesy amount and the date. In extracting these three items, the system will determine the sub-image regions which contain these items, extract them, remove the touching baselines, and pass the extracted items to other recognition models such as [181].

In the following Section, we will briefly present the proposed approach.

7.3 Document Image Segmentation

In this approach, we aim at describing the structure of the processed cheque image to locate the zones which contain the courtesy amount, legal amount and date. For that reason, our first goal is to look for the baselines, i.e., the preprinted straight lines on cheques. The method of extracting these baselines is based on edge images.

As pointed out in [113], preprocessing is completed through two steps: first an edge image is applied to the cheque image to calculate its gradient images (both edge strength and normal direction images), then a thresholding technique is used to select the candidate edge point and to calculate the modified normal direction of edges. The well-known Sobel operator is employed in this system because it is simple and effective in the presence of noise, and can provide gradient directions of edge points.

The key in using the edge detectors, including Sobel operator, is to determine a threshold which differentiates between edge and non-edge pixels in a given image.

In tackling this problem, a local thresholding technique is preferred as opposed to a global thresholding technique because a global thresholding technique will produce thick edges in some regions and thin and broken edges in some other regions. However, a local thresholding technique will be much more reliable because each region will have a threshold value that is different than the threshold value of other regions.

As a result of preprocessing, the output images of candidate edge points and the image of edge direction are produced for further processing.

7.4 Extraction of Baselines

Canadian cheques have lines pre-printed on them. Information is expected to be written or printed above and around these lines. These lines are always parallel to each other on the cheque image. Therefore, these baselines will be extracted based on the analysis of the line segments of edge points as calculated in Section 7.3.

First, we apply the least square fitting technique to the line segments extracted from the edge image. This helps to determine an initial estimation of the parameters of the straight line segments and the direction of baselines. Second, to refine the calculated parameters and the direction angle of baselines in a global manner, a proposed method based on Hough transform technique is presented. Moreover, a special technique is employed in order to re-connect those broken straight line segments that belong to the same baselines. The broken straight line segments exist because of touching or crossing strokes of handwritings.

7.5 Extraction of Targeted Information

Having located and identified the location, size and orientation of each straight line segment in the cheque image, it is very important at this stage to analyze the layout structure of the cheque image. Layout analysis is facilitated by having apriori knowledge about the layout structure of the real life cheque images. Based on the apriori knowledge and the knowledge gathered by the system about the straight line segments, layout analysis is performed to identify the regions in the cheque image

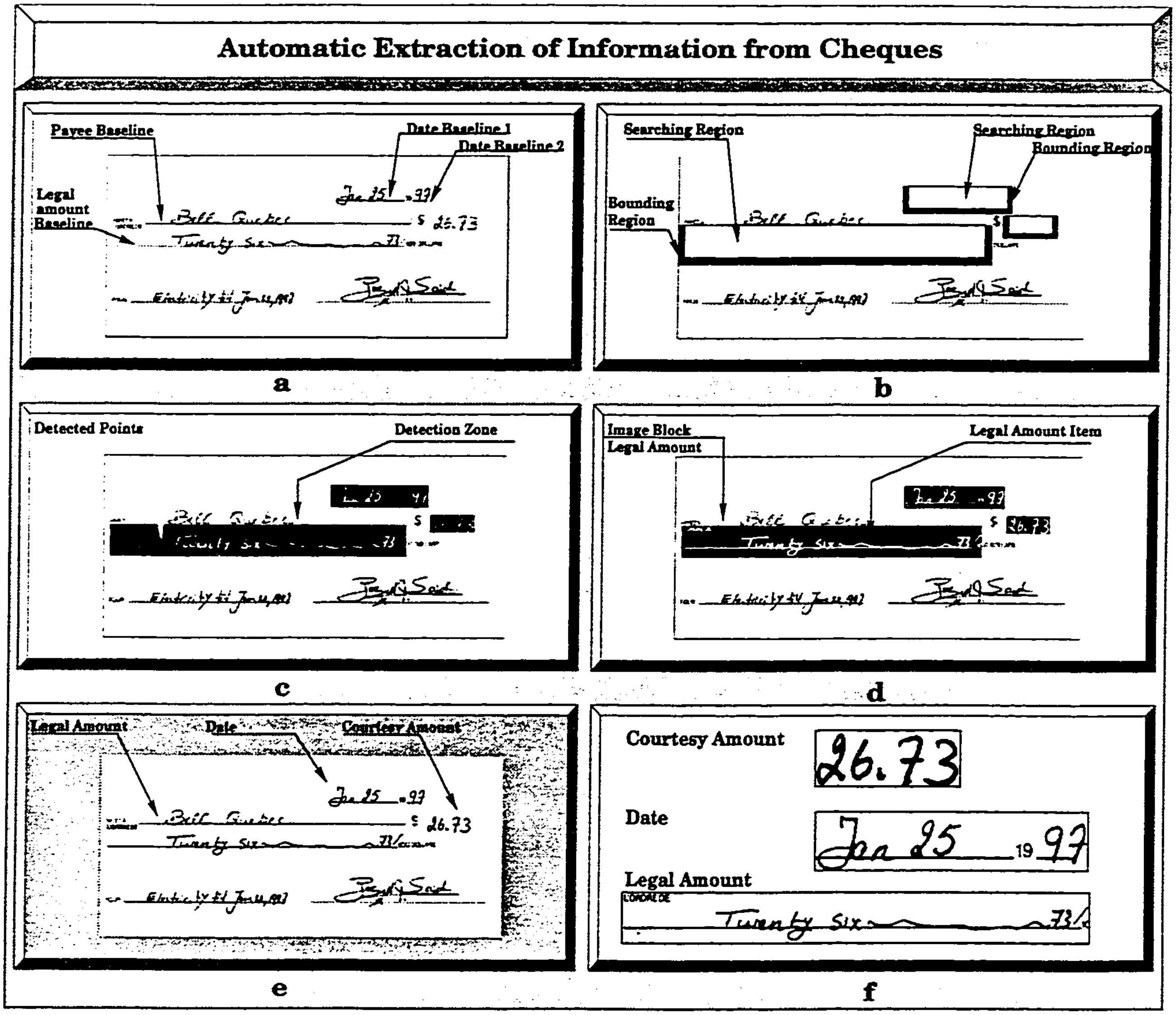

Figure 7.1: A sample output to demonstrate the approach.

that contains the information to be extracted.

After performing the layout analysis of the cheque image, the bounding and searching regions of each item to be extracted are determined, see Figure 7.1 (b). In real-life cheques, the strokes of a legal amount frequently touch or cross the baselines. Based on this research, it can be concluded that for the legal amount, touching and crossing mainly occur between its strokes and the baseline. The strokes may also touch the payee baseline, but the probability of the strokes touching the payee baseline or a baseline other than the legal amount baseline is very low. Therefore, based on this reasoning, the searching region and bounding region for the item to be extracted are determined. Of course, the searching and bounding regions of different items are different because different items are located in different regions in the cheque image; however, the methodology of determining the searching and bounding regions for these items follows the same concept.

After the searching and bounding regions of each targeted item are determined, the system will consider these regions for each item to estimate the grey level distributions for each item Figure 7.1 (c). This helps to distinguish between the grey level distributions of the handwritten data and other pixels. Once the grey level distribution of each item is determined, thresholding is performed in order to segment the handwritten data from the background as shown in Figure 7.1 (d).

Now, extraction of the items is done by finding the connected components of which the grey values of their connected points are lower than a threshold T within a restricted region. All the initial points for tracing out the connected points must be from the searching region and all the connected points must be restricted within the bounding region. A constraint based on the concepts of *row region straight line segments* and *column region straight line segment* can be incorporated in the above connected component tracing to enhance the quality of block images of items. See results shown in Figure 7.1 (e) and (f).

7.6 Separation of Strokes from Connected Lines

The connected components extracted from a cheque image may contain both strokes and lines which either touch or cross each other, or the baselines, or occasionally the bounding box lines of the courtesy amount (these cases can be verified by checking if there are baselines inside the block regions corresponding to courtesy amount items.) Therefore, post-processing is needed to remove such lines which do not belong to the strokes of the legal amount and date from the connected components. An effective technique of separating strokes from connected baselines has been developed based on the morphological closing operation and topological processing [28]. The basic idea to separate baselines from an extracted item is that the whole baseline is first removed from the item image and then the lost information corresponding to the strokes of data is restored by using both morphological and topological processing.

7.7 Experimental Results

An automatic cheque item extraction system has been developed in the X-windows environment of a Sun-Sparc/SunOS. The system accepts as an input a scanned grey level image as a 'tiff' image format. As an output, the system provides three extracted images: (1) the legal amount item , (2) the courtesy amount item, and (3) the date item. A total of 500 real-life bank cheque images, scanned at Bell Quebec is used in the training phase of the system. Experimental results indicate that the proposed approach is promising and its performance is encouraging.

Figure 7.2 provides the results of processing two sets **TEST 1** and **TEST 2** of real cheques from Bell Quebec. Each set contains 200 grey cheque images different from the training data set. The experiments showed that all the baselines on these 400 cheque images were correctly extracted. Figure 7.2 illustrates the results of layout analysis for the cheque images of **TEST 1** and **TEST 2**. The structural description of 5 cheques in **TEST 1** and 4 cheques in **TEST 2** were not derived by the system due to broken, missing or distortion of baselines on these cheque images.

In Figure 7.2, the results of extracting the items from 195 cheques of **TEST 1** and

Experimental Results

Structural Description	Correct	Substitution	Rejection	Reliability
200 Cheques (TEST 1)	97.50 %	0	2.50 %	100 %
200 Cheques (TEST 2)	98.00 %	0	2.00 %	100 %

Item Extraction	TEST 1 195 (200 X 97.5 %)		TEST 2 196 (200 X 98.0 %)	
	Citems	Mitems	Citems	Mitems
Legal Amount	99.49 %	0.51 %	99.49 %	0.51 %
Courtesy Amount	97.95 %	2.05 %	98.98 %	1.02 %
Date	99.49 %	0.51 %	97.96 %	2.04 %

Figure 7.2: Experimental results of TEST1 and TEST2. A courtesy amount when processed by the courtesy amount recognition system is considered correct if it is equal to the courtesy amount as observed by a human.

196 cheque images of **TEST 2**, respectively, are provided, where **Citems** and **Mitem** indicate that the items which can or cannot be correctly located and extracted. It can be concluded from Figure 7.2 that the proposed approach [113] is very effective in locating and extracting different items from the cheque images.

Chapter 8

Experimental Results and Future Directions

Performance Analysis of the Experimental Results and Future Research Direction.

To evaluate the efficiency and the reliability of any item extraction system, it is very important to consider presenting the extracted items to different professional item processing systems that are able to recognize and explicitly give quantitative information that help in judging the quality of an item extraction system. In following up with this line, in this Chapter we will present the experimental results obtained by the date processing system whose input is the extracted date item. Moreover, the experimental results obtained by the legal amount processing system whose input is the extracted legal amount will also be presented. As a final step, a more detailed quantitative performance analysis technique will be presented on the experimental results obtained after presenting the extracted courtesy amount to the courtesy amount processing system. The quantitative performance analysis technique used to quantify the goodness and the reliability of the extracted courtesy amount will be used in our future direction research to produce quantitative measures to quantify the goodness and the reliability of the extracted date and legal amount.

In what follows, we will briefly describe the novelty of these item processing systems as presented by Suen *et al.* [199].

8.1 Date Zone Processing System

Developing an effective date processing system is very challenging owing to the high degree of variability and uncertainty present in the dates handwritten on standard bank cheques which sometimes include complicated background images which are difficult to eliminate. Some of these variations and problems are shown in Figures 8.1 - 8.5.

The contents of a date can be either pure numeric or a mixture of letters (for month) and numerical (for day and year). Month can be written before or after the day as illustrated in Figures 8.2 and 8.3. Since no a priori knowledge is available about the format or style used, it is very difficult and computationally inefficient to develop a system to process the entire date zone at the same time. Therefore, segmentation is introduced so that the problem can be reduced to the separate processing of cursive or handprinted words and numerals.

In processing the date zone images, work should be focused on segmenting a given handwritten date zone image into three subimages, and creating hypotheses to assign such subimages to the day, month, or year field, and to decide the writing style of the month word.

One segmentation has been implemented effectively by passing numeric fields to the recognizer of Section 8.3. Alphabetic fields, when written cursively, can be processed by the cursive word recognizer used for legal amount processing (Section 8.2), after this classifier has been trained on the lexicon of month words and their abbreviations.

When the day and month fields are both numeric, then a parser would be applied to select the acceptable interpretations of the candidate day and month (since these can be written in either order). Naturally, some ambiguities cannot be resolved by this process; for example, the interpretation of "5-10-1995", "03-10-1995" must be left to the user as Figure 8.4 illustrates.

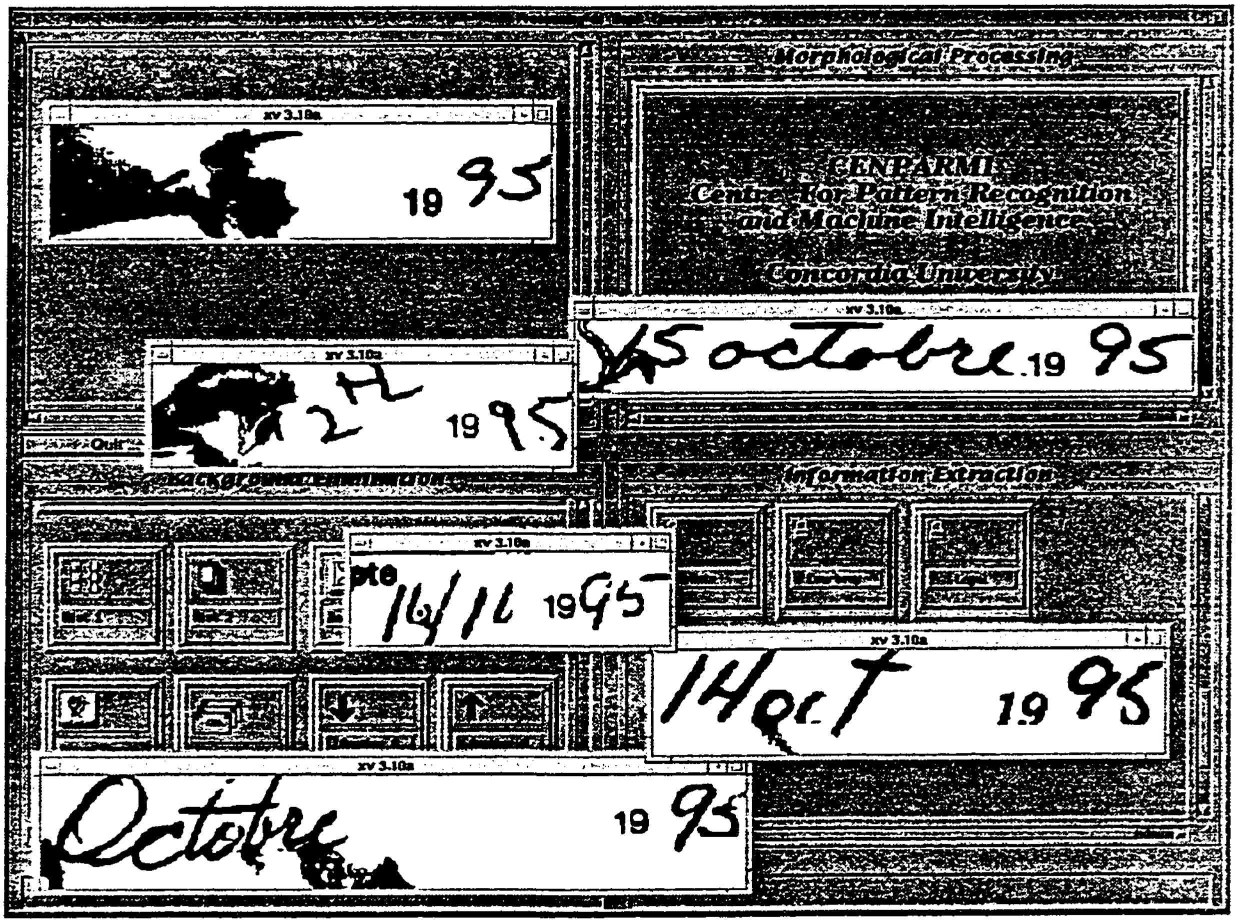

Figure 8.1: Extracted date zones with noise due to light handwriting against a dark background.

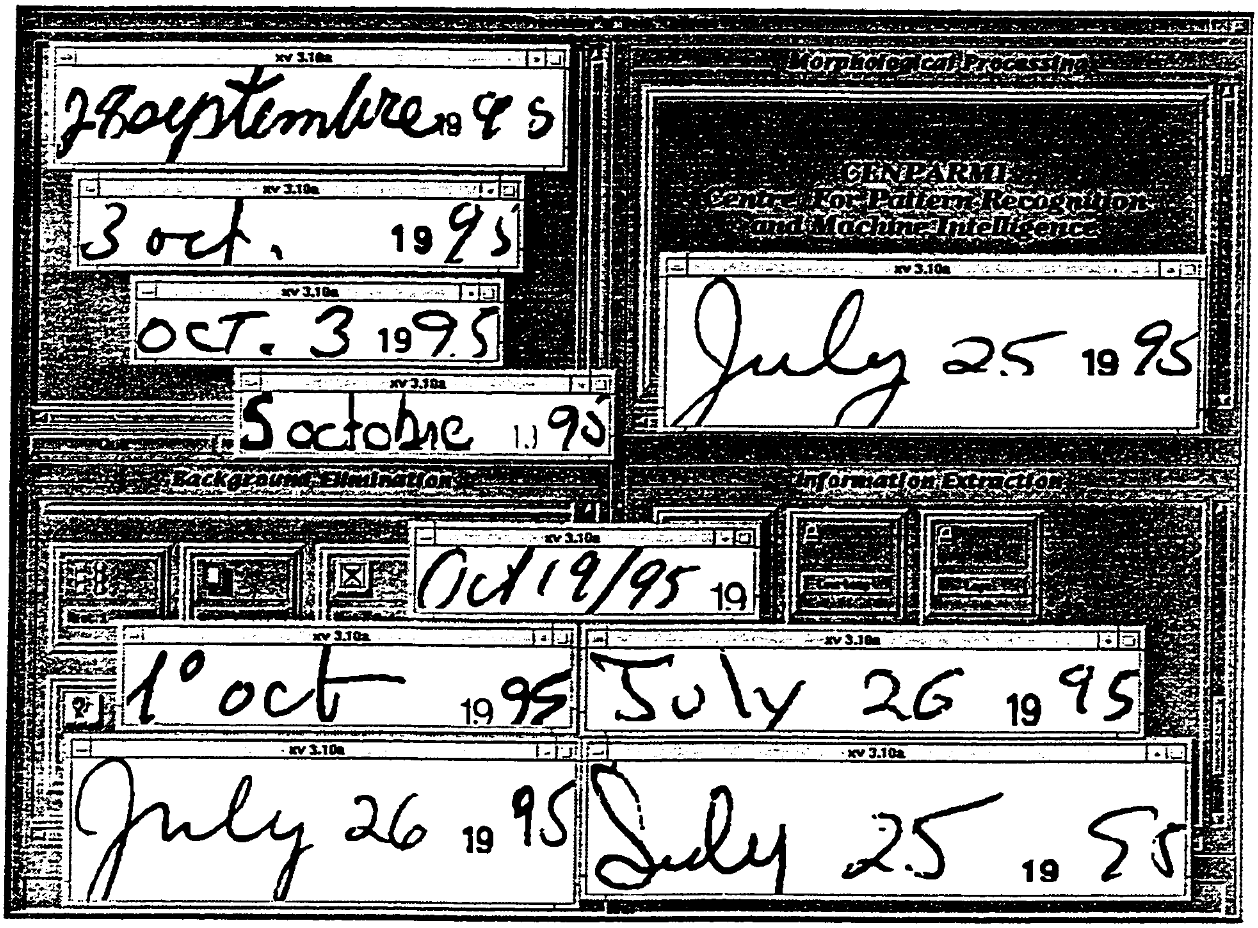

Figure 8.2: Variations and difficulties with date zone processing.

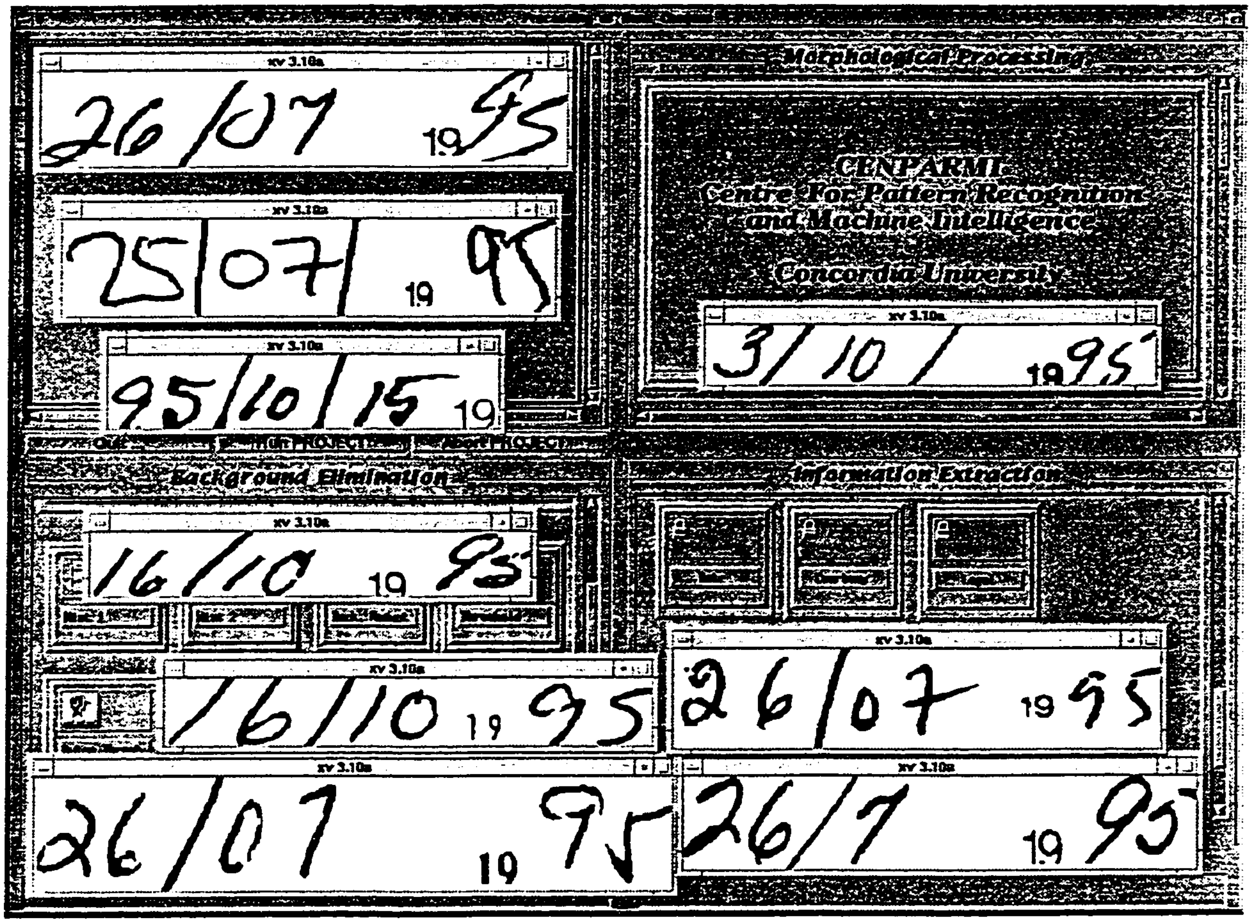

Figure 8.3: Variations and difficulties with date zone processing (continue).

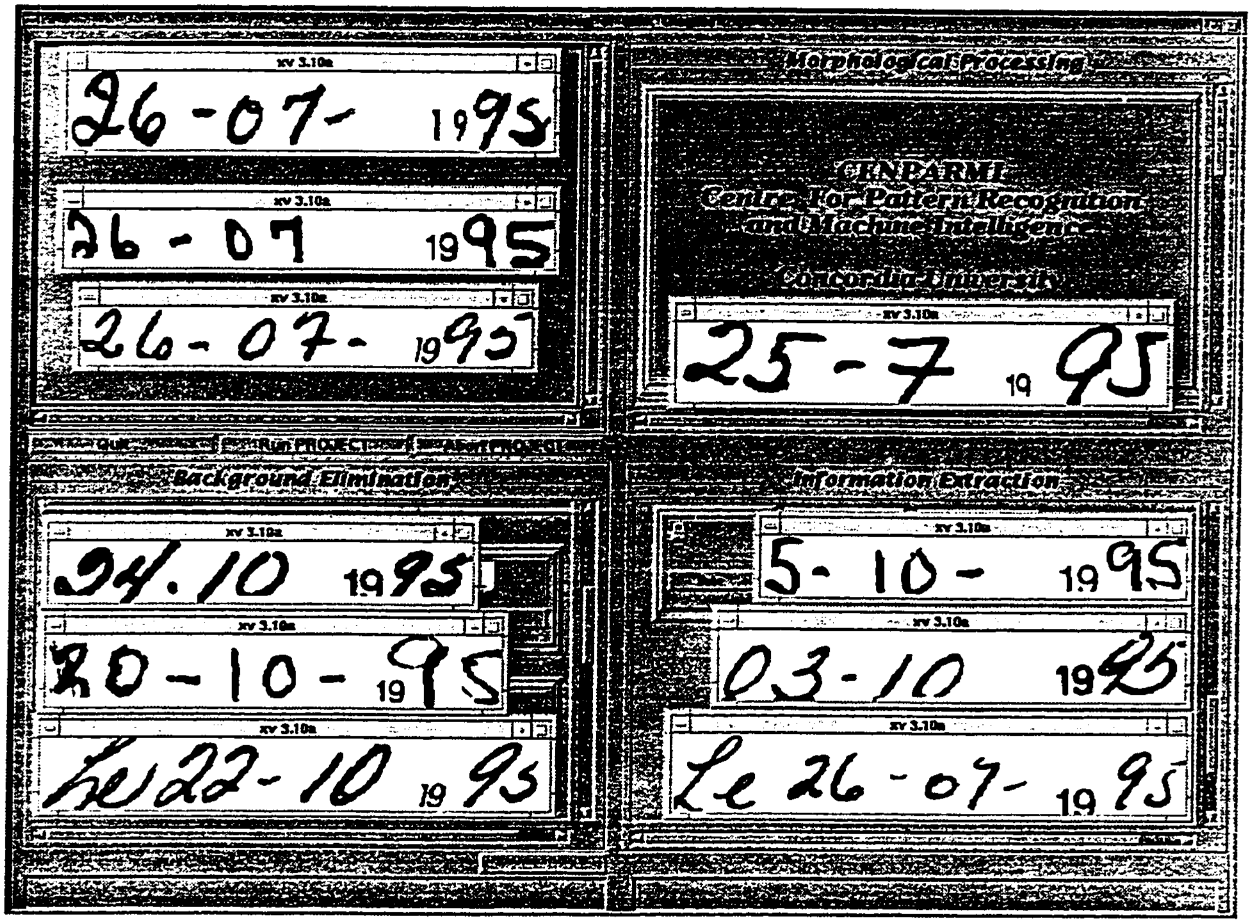

Figure 8.4: Variations and difficulties with date zone processing (continue).

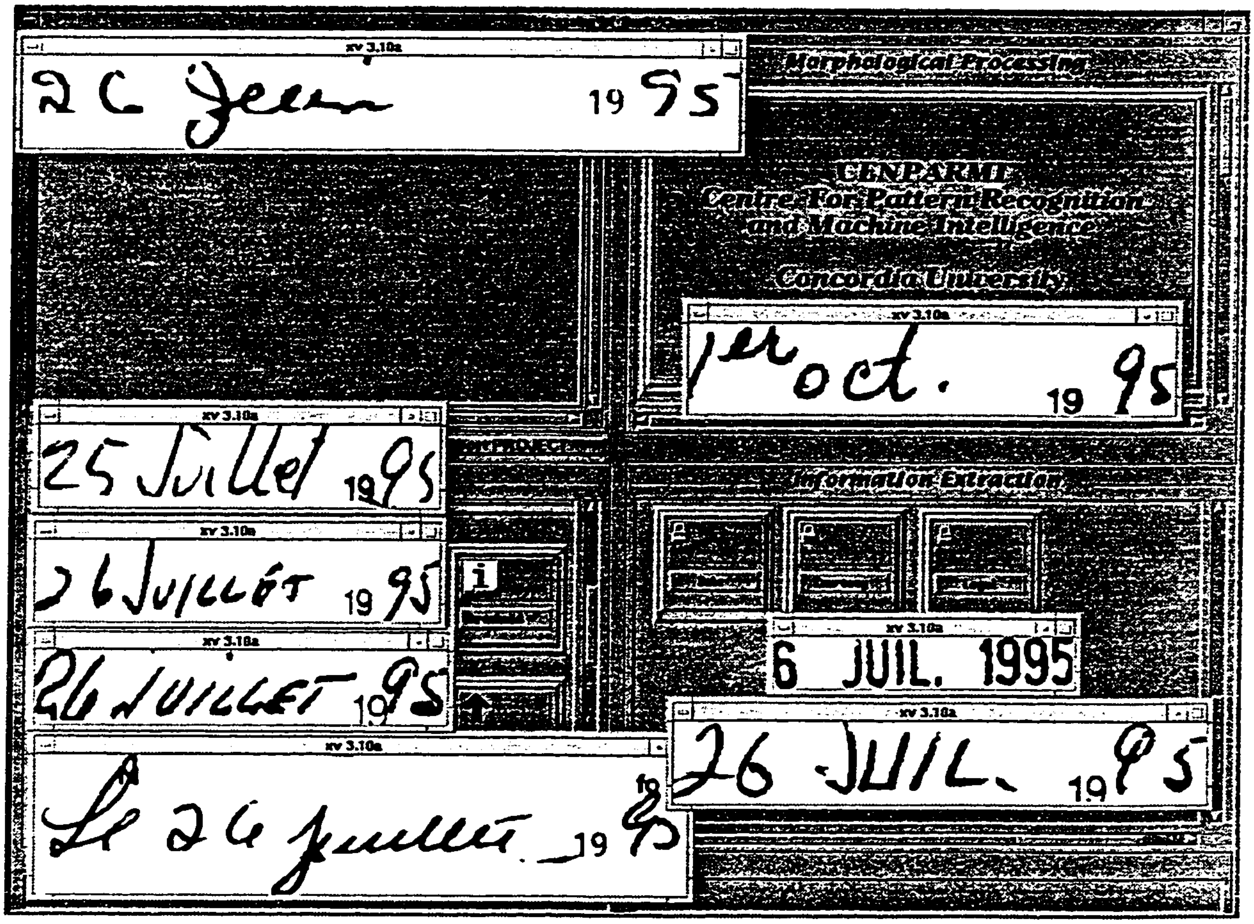

Figure 8.5: Variations and difficulties with date zone processing (continue).

8.1.1 Date Zone Segmentation

Because of the variations described in Section 8.1, date zone image segmentation is a complicated problem for which the following solution according to Lam *e al.* [124] and Suen *et al.* [199] is presented:

1. Given that the year is always written after the printed "19" on all bank cheques, by locating the printed "19," the date zone image can be initially segmented into two subimages, one for the year field (named *Year subimage*), and the other for the day and month fields (denoted by *Day&Month subimages*).

2. Punctuations such as slash ("/"), hyphen ("-"), period ("."), and comma (",") that appear inside the Day&Month subimage can function as separators for the day and month fields. They can also help to indicate the writing style of the date zone image and the type of the month field. Therefore, by detecting punctuations, a segmentation can be attempted, and hypotheses can also be generated of each field.

3. If no punctuation is detected inside the Day&Month subimage (except at the end), then the maximum gap must be located for the purpose of segmentation.

4. If punctuations are detected in the Day&Month subimages, different strategies for segmentation and hypotheses are generated according to the number and types of punctuations detected as well as their positions within the subimage.

8.1.2 Punctuation Detection

The features used in [199] to describe punctuation can be divided into shape and spatial features according to Seme *et al.* [185]. The former describes the shape of the connected component itself, while the latter indicates its location in the Day&Month subimage.

1. *Shape Features:* The shape features used are the following:

- *high_density* : That is the ratio of the number of black pixels to the area of the bounding box. Usually hyphens and periods return high values.
- *narrow:* This is measured by the maximum width of a component divided by its length.
- *flat:* The opposite of *narrow*.
- *slope:* This measures whether the majority of contour pixels have approximately the same slope.
- *small:* The size of the component is compared with both the average component size on the same image and the average period and comma sizes encountered in the training set.
- *Simple_curve* : This is true when "most" rows and columns of the connected component contain only one run of black pixels.
- *no_innerloop* : This condition is used in testing all punctuations and the printed "1," to narrow the scope for punctuation detection.

2. The location of a connected component in the entire date image as well as its position relative to its neighbours are of significant importance for punctuation detection. The four punctuations considered in this application normally appear at different locations. Slashes usually extend from the top to the bottom of the entire image, hyphens tend to appear in the middle, and periods and commas occur at relatively low positions.

   To implement spatial features, it is necessary to define the upper-half and lower-half reference lines of the date image. In this approach, the number of runs is used to determine them. First, the maximum number of runs, *max_run*, is found for the entire *date_zone* image. Then, starting from the top of the image, the first scan line having *max_run*/2 runs is defined as the upper-half reference line, and the first line reaching *max_run*/2 the bottom line defined as the *lower_half* reference line.

   The spatial features used are:

- *exceed_neighbor* (for slashes),
- *in_middle_zone* and *mid_to_neighbor* (for hyphens),
- *below_lower_half* and *low_to_left* (for periods and commas).

3. *Procedure for Punctuation Detection* As the four punctuations considered have different characteristics, different feature combinations are used to detect them:

   - *narrow* and *exceed_neighbor* for slash.
   - *high_density*, *flat*, *at_middlezone*, and *mid_to_neighbor* for hyphen.
   - *high_density, small* and *below_lowerhalf* for period.
   - *narrow, small,* and *below_lowerhalf* for comma.

For efficiency, the *no_innerloop* and *simple_curve* features are used to exclude all unlikely candidates from punctuation detection. In addition, only those components with a height taller than that of the image middle zone are checked for slashes, while only those having the *low_to_left* attributes are tested for periods and commas. The rest are checked for hyphens only.

To compute the likelihood of a connected component being a certain punctuation, we currently use the average confident value obtained for each feature set.

8.1.3 Maximum Gap Detection

Many algorithms have been published to compute the distance between pairs of connected components. However, most of them become more complicated when the connected components overlap each other. Suen *et al.* in [199] introduced an effective and computationally efficient algorithm for locating the maximum gap between connected components. Its performance is very good when connected components are horizontally overlapping or skewed.

The procedure works as follows:

1. Scan each line within the middle zone from left to right.

2. During each scan, record the two connected components where the distance between them is the maximum for this line.

3. After scanning all lines, the two connected components between which the maximum distance occurs most frequently are considered to be separate by the maximum gap.

8.1.4 Segmentation by Hypothesis

According to the number of punctuations detected within the date image, their types and their positions, a set of heuristic rules [199] have been developed to segment the subimage into two parts to assign the parts to the day and month fields, and to hypothesize whether the month field is written in letters or digits.

1. If no punctuation is detected, the image is segmented according to the maximum gap. By comparing the width of the two segments, the one with the larger width is assigned to the month field written in letters, and the one with a smaller width is assigned to the day field.

2. Suppose one punctuation is detected. When this occurs at the end of the subimage, it is removed if it is either a period or comma, and it is ignored if it is either a slash or a hyphen. When the punctuation occurs inside the Day&Month subimage and it is either a slash or a hyphen, then the subimage is segmented at the position of that punctuation. Comparing the widths of the two segments, the one that is at least twice as wide as the other one is assigned the month field written in letters. Otherwise, the segment before the punctuation is assigned the month field written in numeric, and the one after it is assigned the day field.

3. Suppose two punctuations are detected. If they have the same type and one of them occurs at the end of the subimage, then it is removed; otherwise the one with higher confidence value is used, and the other is ignored. If a slash is detected after a period or comma, this slash is probably confused with the

digit "1." Therefore, we ignore the slash, and proceed with segmentation and hypotheses as in Step 2. Otherwise, we select the more likely candidate between the two punctuations, using their confidence values and corresponding weights. The weight for each punctuation type is the correct recognition rate of this punctuation on the training set.

8.1.5 Experimental Results of Segmentation

According to [199] one thousand date images obtained from CENPARMI cheques have been used to train the segmentation system. The experimental results are presented in **TABLE VIII**.

TABLE VIII

Date Zone Experimental Results

Date Zone Training Set of 1000 images						
	Reject 1	*Reject 2*	*Reject 3*	*Total*	*Correct 1*	*Correct 2*
Number	14	49	1	64	758	758
Rate	1.4 %	4.9 %	0.1 %	6.4 %	80.98 %	75.80 %
Date Zone Testing Set of 261 images						
Number	0	15	0	15	201	201
Rate	0.00 %	5.75 %	0.0 %	5.75 %	81.71 %	77.01 %

In **TABLE VIII**, *Reject 1* represents the rejection cases due to improper binarization, as illustrated in Figure 8.1; *Reject 2* represents the rejection cases due to printed "19" that is connected to neighbouring handwritten data as illustrated by the handwritten dates "26 / 07 1995" in Figure 8.3, and "26 JUILLET 1995" in Figure 8.2; *Reject 3* represents the rejection due to complete touching of Day&Month subimages as illustrated by the handwritten date "28 septembre19 95" in Figure 8.2; *Correct 1* excludes the images rejected and *Correct 2* includes all images.

A total of 261 date images have been used to test the segmentation system, based on the statistics (used to obtain thresholds for feature algorithms) collected from the training set. The results are shown in **TABLE VIII**.

8.2 Legal Amount Processing System

The processing of legal amount involves the recognition of unconstrained handwritten cursive words, which remains an extremely challenging problem owing to the great variability in handwriting styles and handwriting devices. Figures 8.6 - 8.8 illustrate some of the extracted legal amounts. Current research aims at developing systems for limited domain applications where a defined lexicon plus a well constrained syntax help provide a feasible solution to the problem. The method proposed by Guillevic in [45] is global in nature and avoids the difficult segmentation stage of common word recognition techniques.

8.2.1 Preprocessing

The most significant preprocessing step that is needed in [45] and [199] is to perform a slant normalization. The average slant of a word is computed by the analysis of slanted vertical histograms at various angles. The heuristic for finding the average slant is to look for the greatest positive derivative in all of the slanted histograms computed. The image is then corrected through a shear transformation.

8.2.2 Global Word Features

Seven different types of global features, according to [45], are extracted: ascenders, descenders, loops, an estimate of the word length, as well as vertical, horizontal and diagonal strokes. The system will look at the ascenders (descenders) by following the upper (lower) outer contour of the word. For coding into a feature vector. as ascender is said to occur at the midpoint between its starting and finishing points. Loops are detected as inner contours. The system estimates of the word "length" is the number of "central threshold crossings". Strokes in four directions are extracted using mathematical morphology operators. The size of the structuring element has been determined empirically relative to the average stroke thickness.

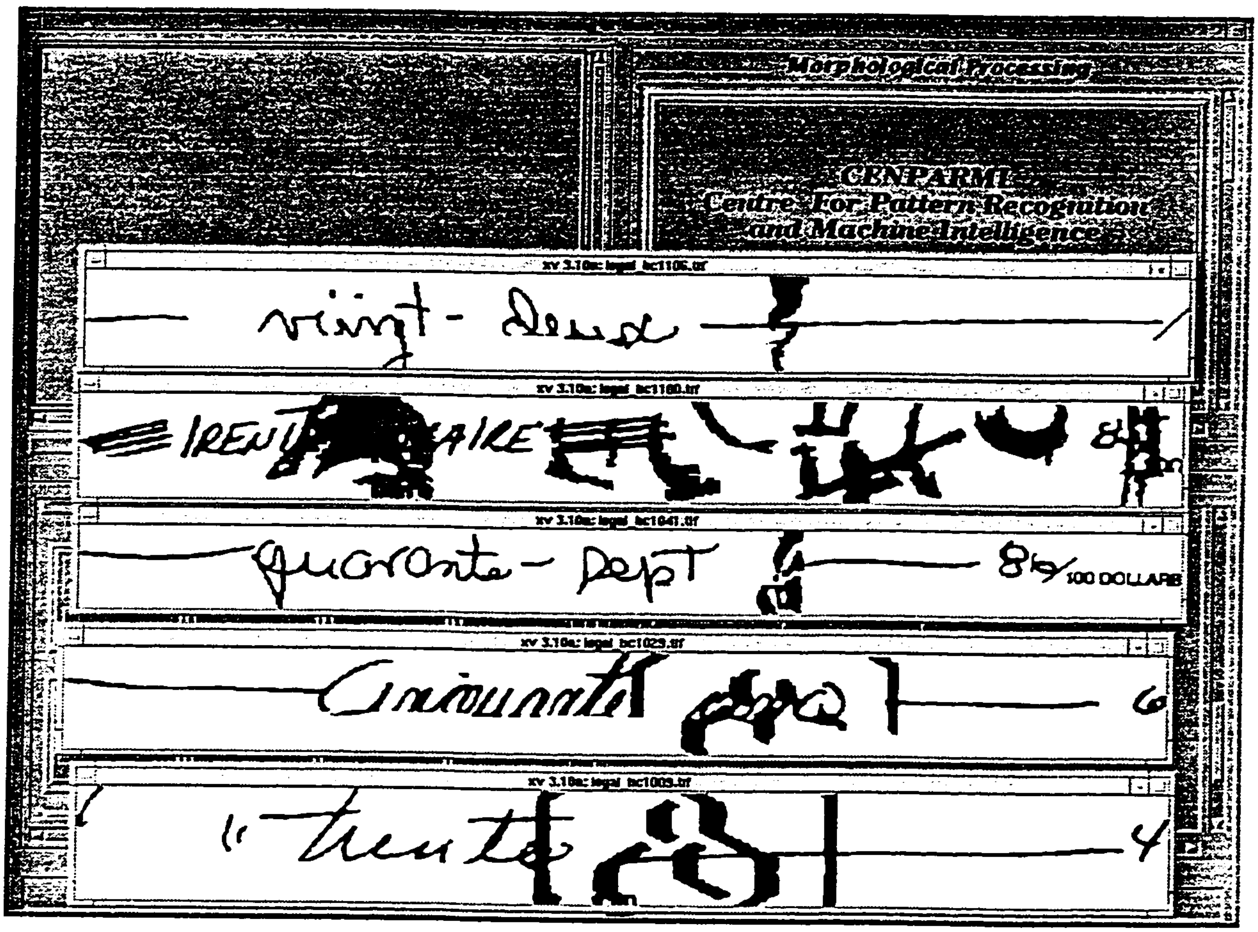

Figure 8.6: Extracted legal amounts with noise due to light handwriting against a dark background.

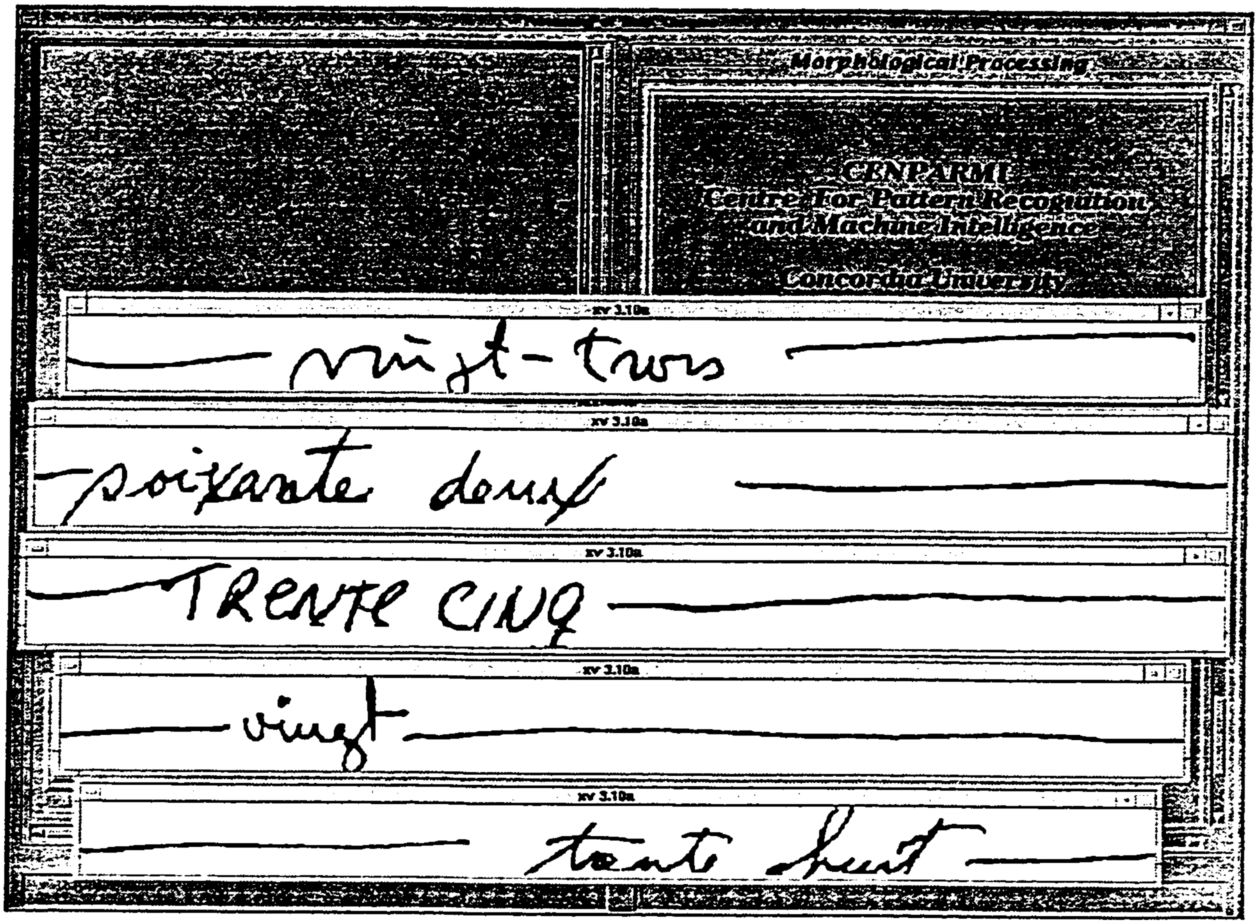

Figure 8.7: Variations and difficulties with French legal amount processing.

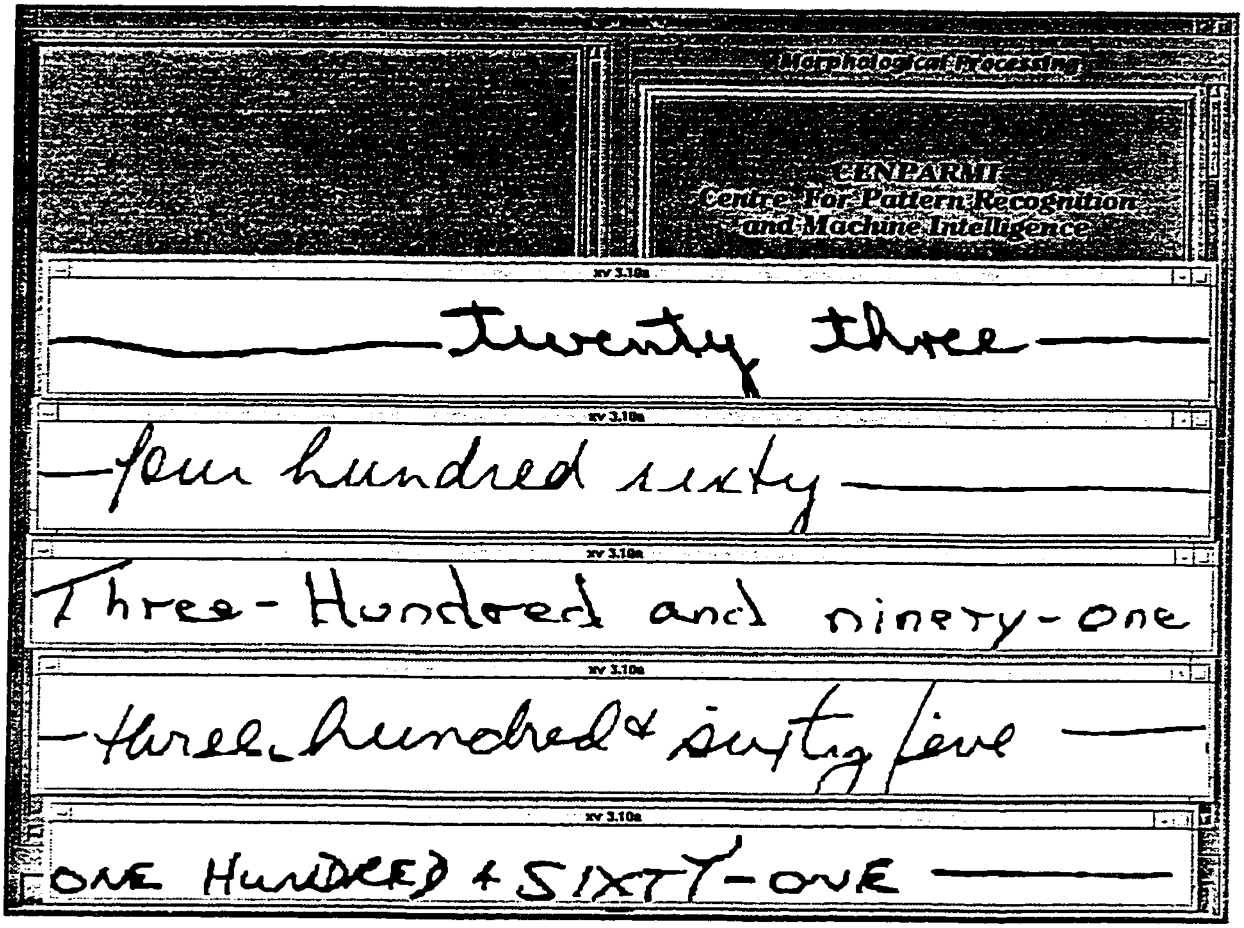

Figure 8.8: Variations and difficulties with English legal amount processing.

8.2.3 Feature Vector

In the classification scheme, the system as presented in [45] and [199] differentiates between input and class feature vectors. The input feature vector is computed when features are extracted from some input image. When the system attempts to classify the input word, for each class in the lexicon used the system converts this input feature vector to the class feature vector specified to the class being considered. The word feature vector consists of 11 components, namely, the relative position of ascenders, descenders, loops, strokes as well as the number of ascenders, descenders and loops, and finally the word length.

When an ascender is detected in some input image, its position relative to the input width is computed. For example, suppose an ascender is detected at the position 85% in some input word. When it is matched to the class "and," which is a three-letter word, the detected ascender will be converted into an ascender at the third position in the word.

8.2.4 Classification

According to [45] the feature vector of each image consists of 11 components. To compute the distance between two feature vectors the system considers 11 sub-distances between the corresponding sub-vectors. The final distance measure is taken as a weighted sum of those sub-measures. A minimum shift distance, which measures the distance of the shifting needed to have the two sub-vectors match is designed. This heuristic incorporates the intrinsic notion of position features, and it is sufficient for this purpose.

The distance from one input vector to a training feature vector is then computed as a weighted average of the 11 sub-distances. The 11 weights have been optimized by a genetic algorithm [70]. For an input image, its distance from each lexicon class is determined as the average distance from the k nearest samples of that class, and the classes are ranked according to distance.

A grammar has been designed to parse the results of the word recognizer. For each word in the legal amount, the word recognizer outputs a ranked list of the top 10 possibilities, from which the parser produces a ranked list of possible legal amounts with corresponding distance measures.

8.2.5 Experimental Results

As stated in [199], the word recognizer has been trained on a set of 1496 legal amounts (5322 words) obtained from the CENPARMI database of real life bank cheque images, and tested on a different set of 712 amounts (2515 words). The results of the system global classifier on those images are shown in **TABLE IX** [199]. Note that these results correspond to the words taken in isolation, and can result from the use of the parser.

TABLE IX

Legal Amount Experimental Results
Legal Amount Training Set of 1496 images

	Language	*Lexicon Size*	*1 %*	*2 %*	*5 %*	*10 %*
Guillevic and Suen	English	32	72.6	84.0	94.3	98.4
Gilloux et al.	French	27	79	87	95	98

Based on the analysis done on the results, it is noted that the classifier performs better on those words that have distinctive global features such as "hundred", "thousand" or "and". The distinction between some word classes is problematic. For example, "Nine" and "One" have similar features: an ascender at the beginning of the word and a loop at the end, the loop in the "O" not being a robust feature. Therefore, there is a need to extract another important feature for a fast reader [71], namely, the characters adjacent to blank spaces.

8.3 Courtesy Amount Processing System

According to [199], the input to this system is a binarized image that represents the extracted courtesy amount from a given bank cheque image. The output could be a numerical number that corresponds to the correctly recognized input, a rejected output image, or an output error image. In what follows we will briefly present the steps required to process the input image.

- Removal of spurious connected components;
- Grouping together of related components;
- Splitting apart of connected characters;
- High-performance digit recognition with noise detection.

It is understood that the digit recognizer is the last step after extracting all required information. It must be able to recognize its target classes with high accuracy as well as reject wrong segmented data. Its role is therefore not just as a specialist in its target classes, but also as a specialist in rejecting all other classes.

The common strategy of repeated segmentation to obtain a confident recognition result has been used in the present system, especially in the connected digit splitter. and to a lesser extent in the grouping of broken characters.

The steps in the courtesy amount recognition procedure as presented by Suen *et al.* in [199] are:

- Locate connected components.
- Remove components with perimeter below a minimum length.
- Reject images with more than a threshold number of components and images with dissproportunately wide components.
- Sort components left to right.
- Do a context-free classification of each object into one of 14 classes chosen from the 10 digits, dash, period comma, or other.
- Group components of potentially broken characters together.
- Submit multicomponents groups to the digit classifier.
- Disband certain groups rejected by the digit classifier.
- Submit previously rejected single components to the digit splitter/classifier.

- Reject any courtesy amount that yields one or more rejected symbols or that yields a splitting result of more than two digits for a single connected component.

Connected components are found by tracing contours using the algorithm given in [197]. Sorting of components is by the average value of contours perimeter horizontal ordinates. Digit recognition is done by extracting features for classification by neural networks to be described later.

Low-level grouping is done by grouping together components with stroke tips that are within a threshold distance of each other. Higher-level grouping is done by grouping dashes in the upper part of the image with their neighbouring component on the left; and grouping any "1" together with any upper image component on its left (for possible "4"s). These criteria can result in over grouping; hence, the need for some backtracking. At the current stage of development disbanding of a group in step 8 is done only if the group has two components, and if at least one of those components had a confident recognition result from step 5.

Recognition of punctuation marks is based entirely on three features of component contours: length of contour cumulative straightness of contour segments between stroke tips in the component, and cumulative slopes of these segments.

8.3.1 Digit Splitter

The digit splitter operates by attempting to segment and recognize the leftmost digit from a connected component as presented by Strathy *et al.* [198]. If successful, it then attempts to recursively recognize the remainder of the digit string. For each digit according to [199], it tries up to five possible segmentations where each segmentation is associated a confidence recognition value to separate each digit in the string. In other words, the procedure works as follows:

1. Attempt to recognize the component as a single digit. If recognition is successful, return the recognizer result.

2. Otherwise, generate a list of possible segmentations.

3. Attempt up to five segmentations until a confident recognition of the left component is achieved. If unsuccessful then return a rejection result.

4. If the leftmost digit has been recognized, recursively process the remaining rightmost components.

5. If the rightmost component yields a confident recognition, concatenate it to the result for the left hand digit and return the symbol string; otherwise, resume step 3.

8.3.2 Neural Network Classifier

As stated in [199], digits are recognized by a combination of three backpropagation neural networks using different sets of input features. The combined decision is obtained from the sum of the outputs. Of the three networks, network A uses the pixel distance features described below, while networks B *and* C use the size-normalized image pixels.

1. Pixel Distance Features: These consist of two features for each pixel, measured in the horizontal and vertical directions. The horizontal pixel distance feature (PDF) of a given pixel gives the direction and distance from that pixel to the nearest black pixel in the same horizontal scan line. The direction is indicated by the sign: negative if the nearest black pixel is to the right, positive if to the left, as shown in Figure 8.9. Moving away from a black pixel in either direction, the absolute PDF values count upward. At the midpoint between two black pixels the absolute PDF value is maximum. Vertical PDFs are analogous.

2. Preprocessing: Network A uses the following preprocessing steps:

- Thin the input image.

- Extract horizontal and vertical PDFs.

- Average each PDF array to 12×14 high such that values lie in the range $[-1, +1]$.

-1	0	0	0	0	1
0	1	2	-2	-1	0
-1	0	1	2	-1	0
-2	-1	0	0	0	1
-1	0	1	-1	0	1
0	1	2	-2	-1	0
0	1	2	-1	0	1
-1	0	0	0	1	2

Figure 8.9: An example to illustrate the horizontal pixel distance feature array of the character "8".

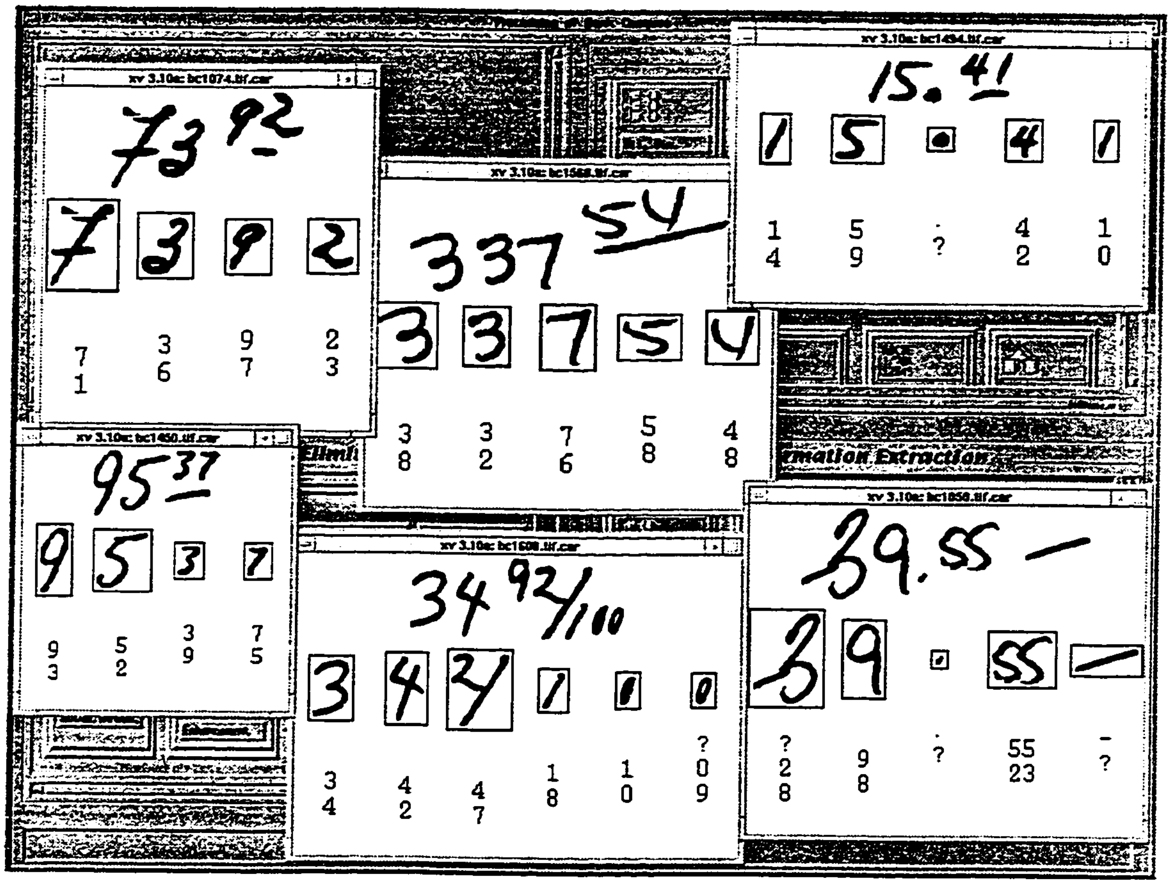

Figure 8.10: Examples of courtesy amount recognition.

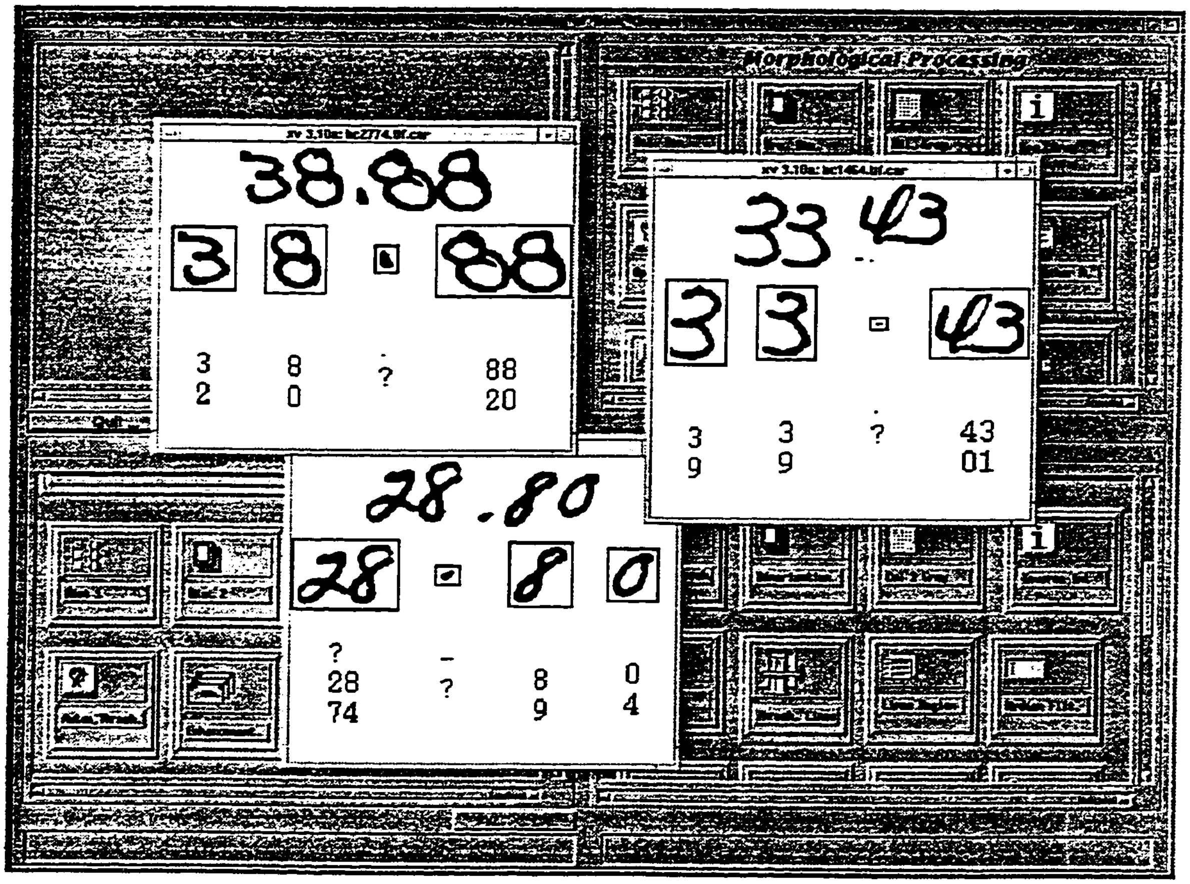

Figure 8.11: Examples of courtesy amount recognition (continue).

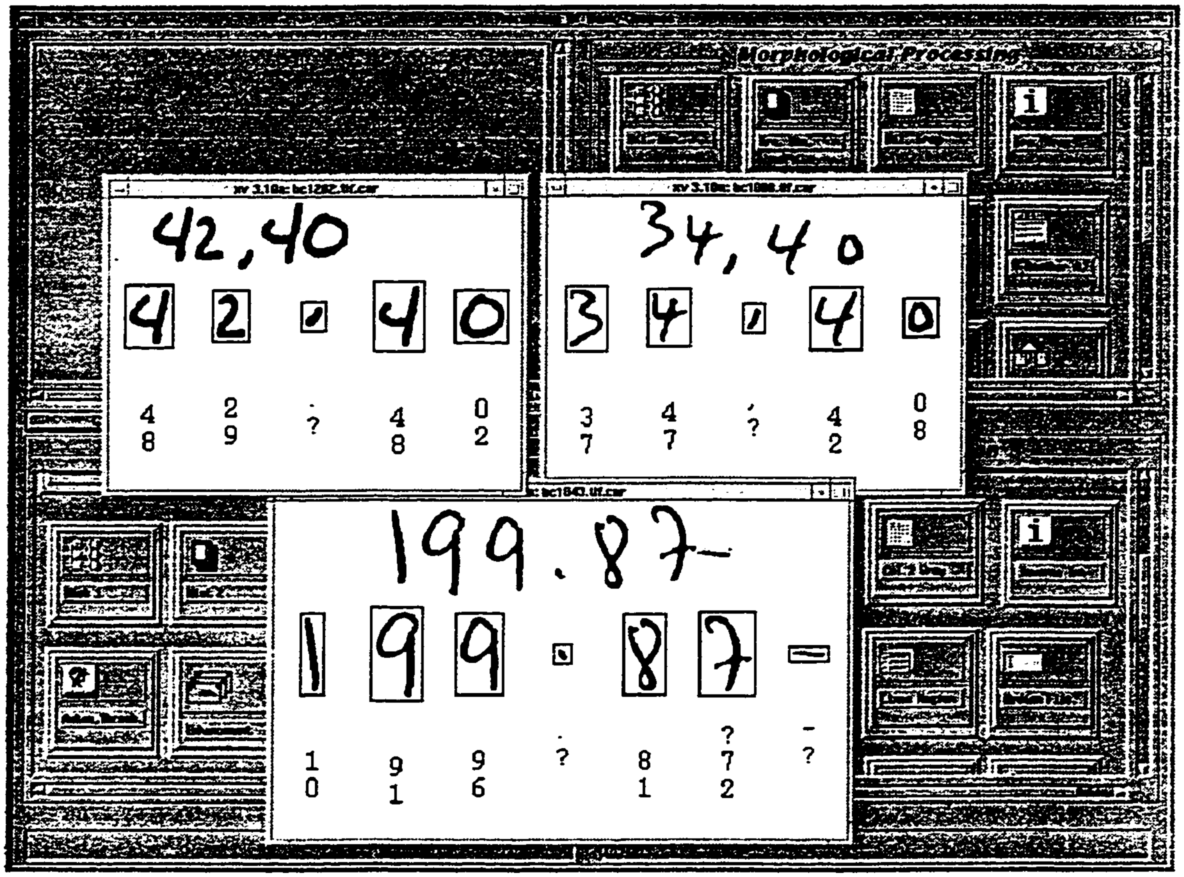

Figure 8.12: Examples of courtesy amount recognition (continue).

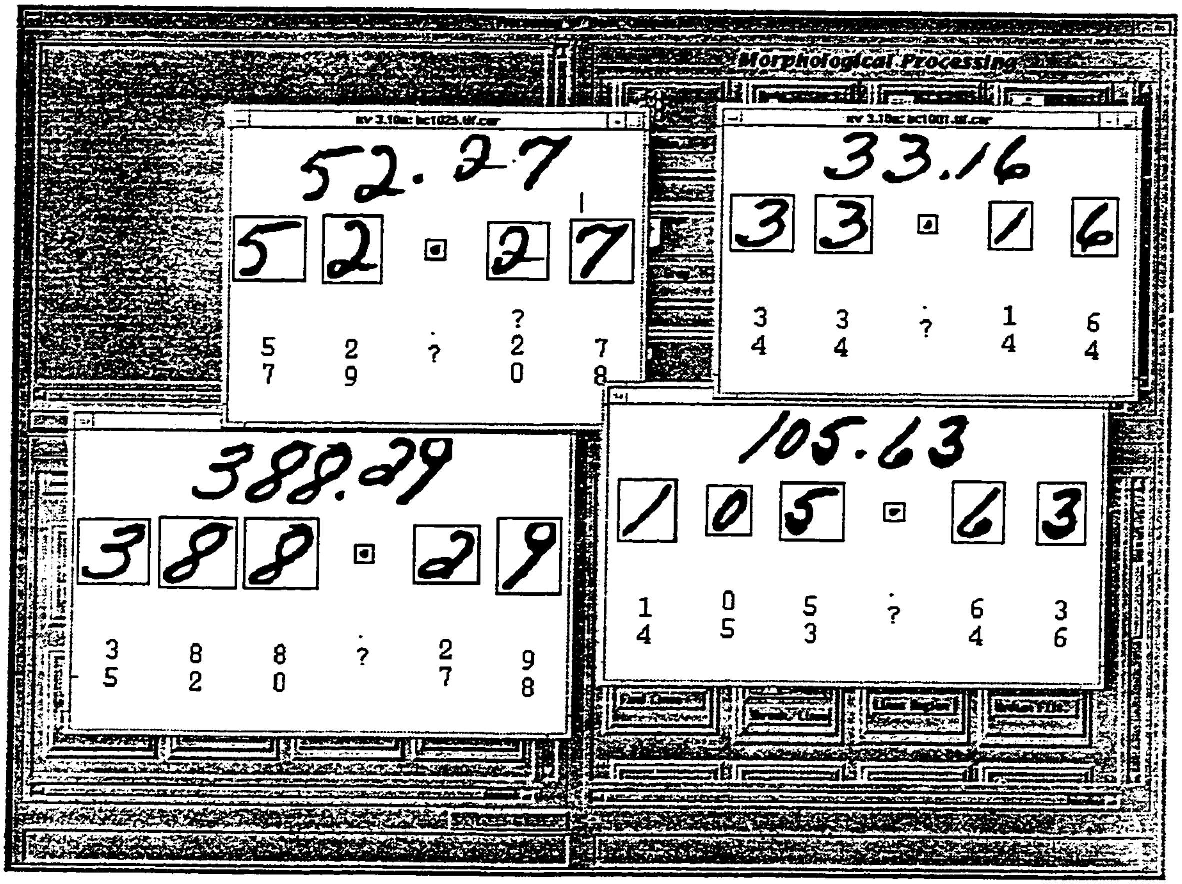

Figure 8.13: Examples of courtesy amount recognition (continue).

Preprocessing for net B is very simple: The input image is scaled to 16×16. For net C the input image is rotated by 45^o before scaling as in net B.

3. Architecture: Each network has three layers. Net A is fully connected, having 336 inputs, 70 hidden, and 10 output units. Nets B *and* C are not fully connected, and they have the same architecture, with 256 input, 560 hidden, and 10 output units. Each block of 20 hidden units is connected to a triplet of contiguous rows or columns of the input layer. Half of the hidden units are connected to row triplets of the input, and the other half are connected to column triplets. The triplets are allowed to overlap. The hidden layer is fully connected to the output layer.

4. Training and Results: Each network was trained on the 18,468 training samples of the CEDAR-CD-ROM [84] to higher than 99.8% recognition. Recognition results on the 2711 numerals in the CEDAR BS and the 2213 samples in the GoodBS sets are given in **TABLE V**. These results include the zero rejection recognition rates for the individual networks as well as their combination by output summation, and they are comparable to the results reported in [125].

The training and testing sets represent the quality of data encountered in a real application. Extremely noisy images that result from binarization problems are normally detected at the segmentation stage, from the presence of too many or oversize objects, and these objects are usually not passed to the classifier. Pepper noise is removed before classification. Experience with this classifier, according to [199], indicates that reasonable results can be expected even from badly broken digits. At present, high accuracy recognition produces a high number of no-decision (rejection) results, i.e., the system chooses to reject a digit rather than risk a wrong decision. Correct results can be expected from the current prototype if the input consists of clearly written and well separated characters chosen from the class “0”-“9”, “.”, “.” and “-”.

TABLE X				
Recognizer's Experimental Results CEDAR BS and GoodBS Test Sets				
	A %	*B %*	*C %*	*ABC %*
BS	96.27	96.24	95.68	97.31
GoodBS	98.51	98.46	97.97	99.19

In conclusion, extremely high-digit recognition performance is a practical necessity for a system such as the present one. Substantial increases in reliability have been observed during the development of the system as a result of improving the performance of the digit classifier. By continuing this process, there is a good reason to expect signification improvements of the system.

8.4 Performance Analysis of the Experimental Results

As mentioned earlier in this Chapter, to evaluate the efficiency, reliability, and performance of any item extraction system, it is very important to consider presenting the extracted items to different professional item processing systems that are able to recognize and explicitly give quantitative information that helps in evaluating the quality of an item extraction system. In pursuing this line, we will present the extracted (courtesy amount) by the top-down and the bottom-up systems presented in Chapters 6 and 7 respectively to the item processing system presented in Section 8.3.

8.4.1 Courtesy Amount Error Classification

To obtain quantitative statistical data from these experiments it is helpful to classify the output results of item processing systems at its finest granularity. For this reason. in what follows, we will classify the resulting images that are outputed by the courtesy amount processing system of Section 8.3 into two classes: (1) extraction difficulties and (2) recognition difficulties. Each class describes finer details on the types of difficulties observed. These errors will later be of great importance to study the future advancements that the future approach should undertake in developing a combined general item extraction system.

I Extraction Problems

a. Failure to eliminate too many handwritten intruders belonging to other baselines that intersect the courtesy amount as illustrated in Figures 8.14 and 8.15.

b. Failure to eliminate rectangular boxes that have the same intensity as the courtesy amount. Figures 8.16, 8.17, and 8.18.

c. Failure to extract the courtesy amount due to difficulty in segmentation when the handwritten data are lighter than the background.

d. Failure to eliminate noise due to rotated cheque images. Figure 8.19.

e. Failure to remove noise due to very dark background. Figure 8.20 and 8.21 illustrate.

f. Failure to segment items due to non-uniform cheques. Figure 8.22.

II Recognition Problems

a. Recognition errors (even in case of very good extraction)

1 A "," is interpreted as a "1". Figure 8.23.

2 A zero "0" is recognized as a decimal point ".". Figure 8.24.

3 A decimal point "." is not recognized. Figure 8.25.

4 Sometimes the recognizer does not correctly recognize characters even in the case of clear and clean extraction. (the handwritten letter "7" is recognized as a "1" in most of the cases.) Figure 8.26.

b. Writer's Behaviour

1 Draw a dashed line under the decimal part. The dashed lines might be connected to the handwritten courtesy amount causing a recognition error. Figure 8.27.

2 Draw dashed lines before or after the courtesy amount. These dashed lines could some times be connected to the courtesy amount and will cause recognition errors. Figure 8.28.

3 Connected digits are causing a segmentation problem and therefore a recognition error. Figure 8.29.

4 Some symbols that belong to and are disconnected from a certain digit could be connected to neighbouring digits. In this case, the symbol is considered as part of the neighbouring digits and therefore causes a segmentation and a recognition problem. Figure 8.30.

8.4.2 Courtesy Amount Training and Testing Results

A training set of 500 real life bank cheque images obtained from the CENPARMI database is used to train both systems presented in Chapters 6 and 7. An exclusive two exclusive testing sets: Set1 and Set2 with 200 images each are also used to test the performance of these systems and compare their experimental results.

TABLE XI and **TABLE XII** present the experimental results done after presenting the extracted courtesy amounts of the training bank cheque images of both systems to the courtesy amount processing system. In fact, the training set is used to train both systems which are tested with another exclusive testing set Set1. The experimental results are presented in **TABLE XIII** and **TABLE XIV**. In fact Set1 is used to collect feedback from the output images provided by the courtesy amount recognizer to enhance the item extraction systems before using testing set Set2 to obtain the real performance of both systems. The experimental results of testing Set2 are presented in **TABLE XV**. It is clear from **TABLE XV** that the reliability of both systems is increasing as more feedback is collected from Set1.

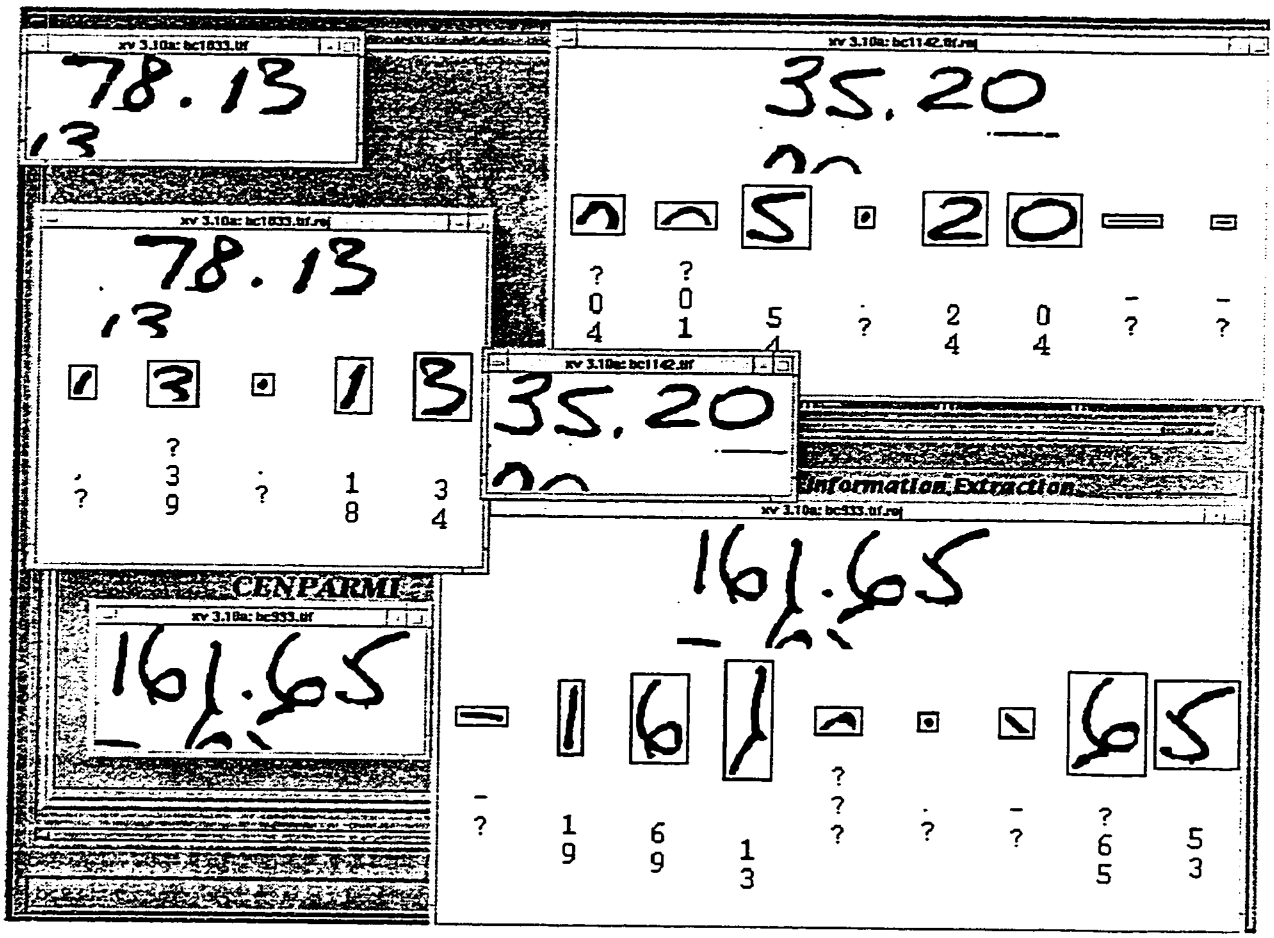

Figure 8.14: Extraction Difficulty: Failure to eliminate intruders (I.a).

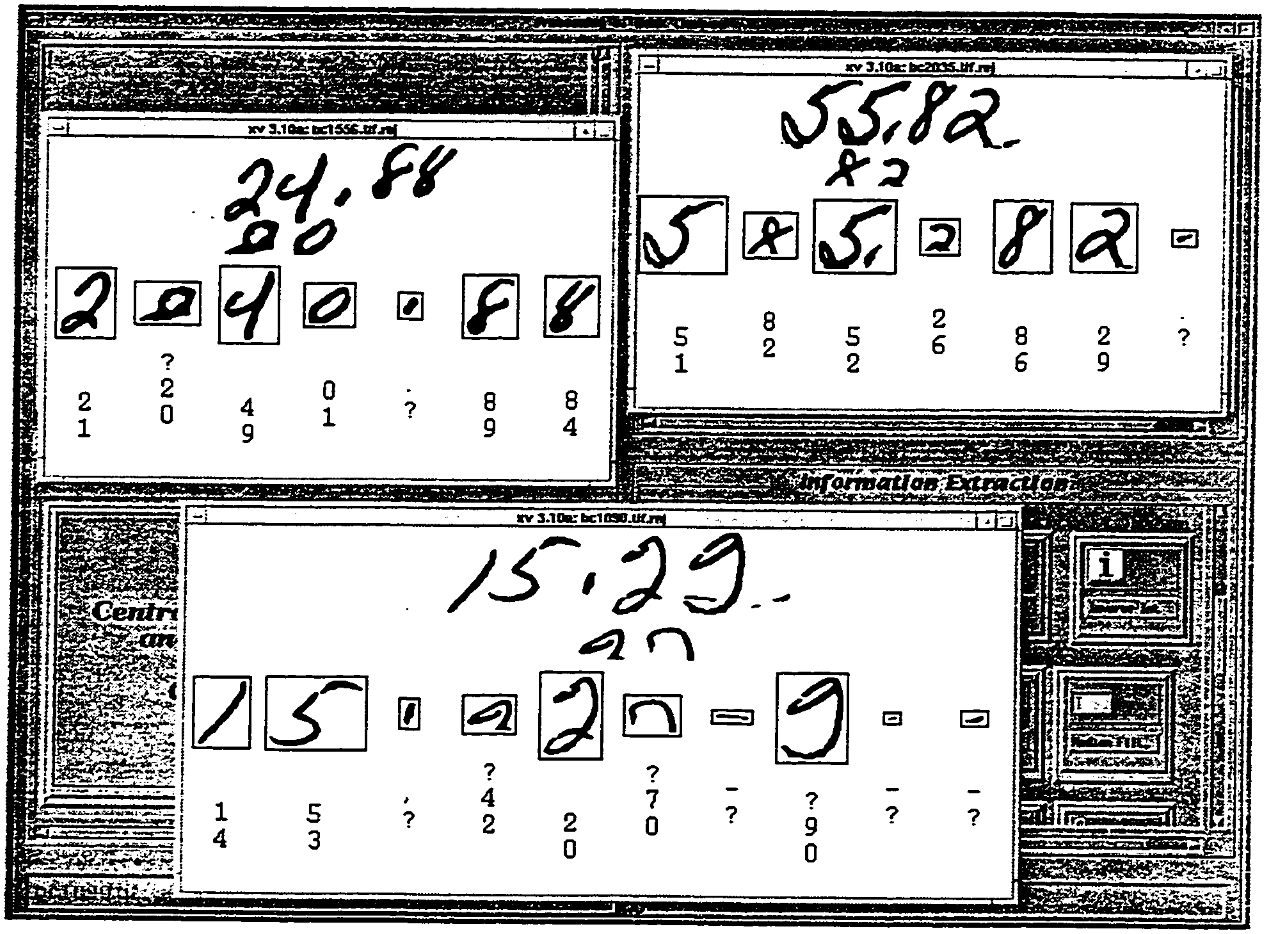

Figure 8.15: Extraction Difficulty: Failure to eliminate intruders (I.a) (continue).

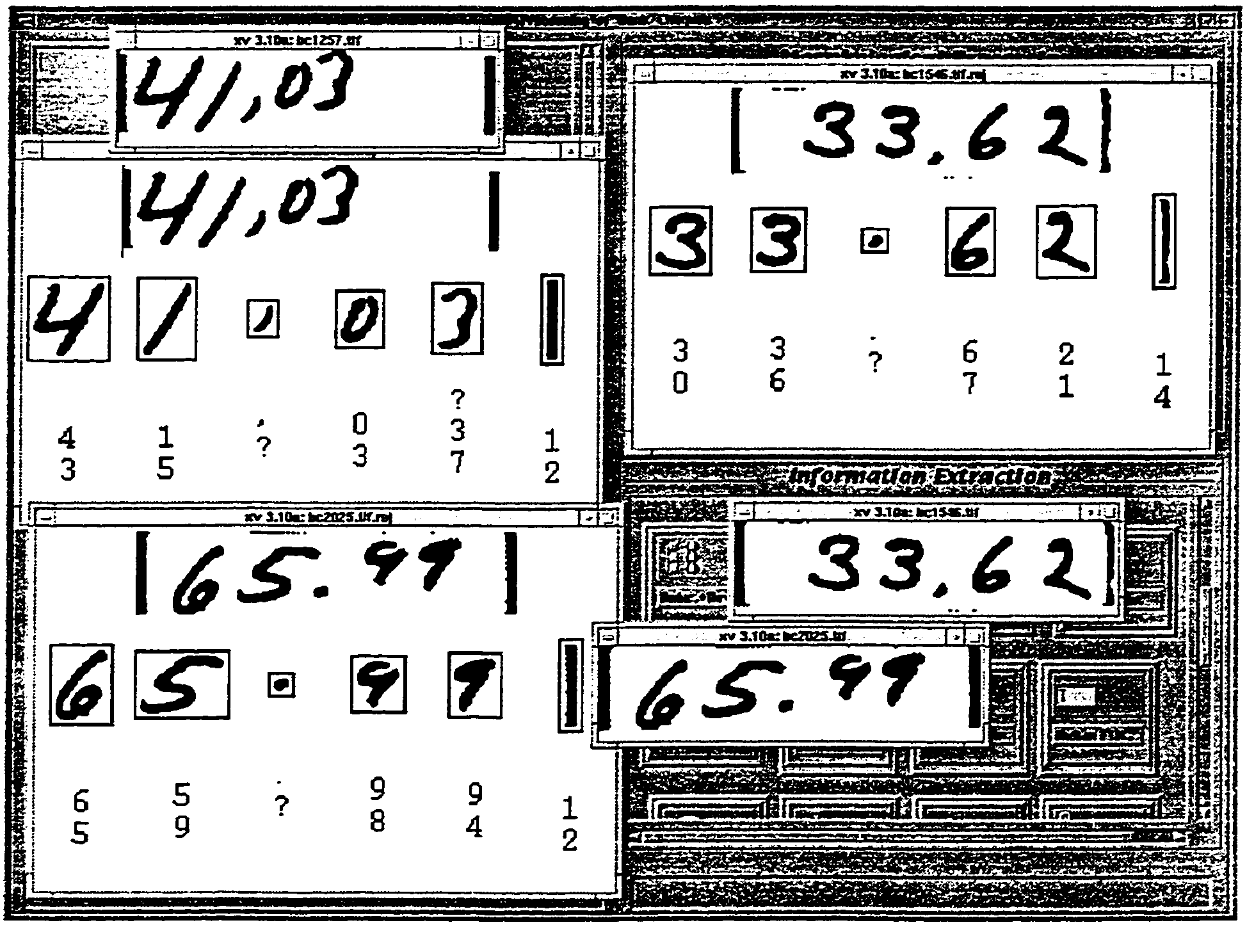

Figure 8.16: Extraction Difficulty: Failure to eliminate rectangular boxes (I.b).

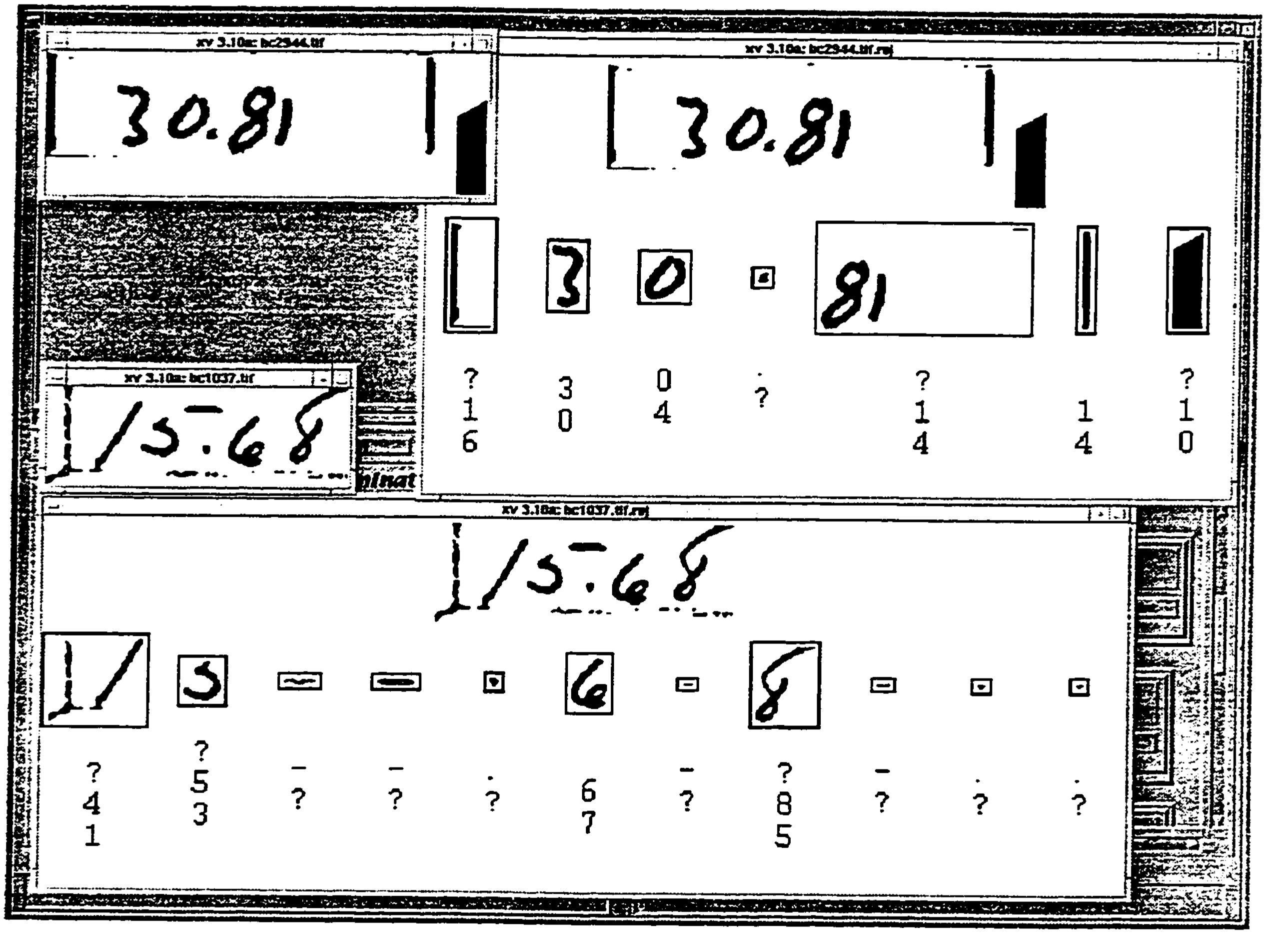

Figure 8.17: Extraction Difficulty: Failure to eliminate rectangular boxes (I.b) (continue).

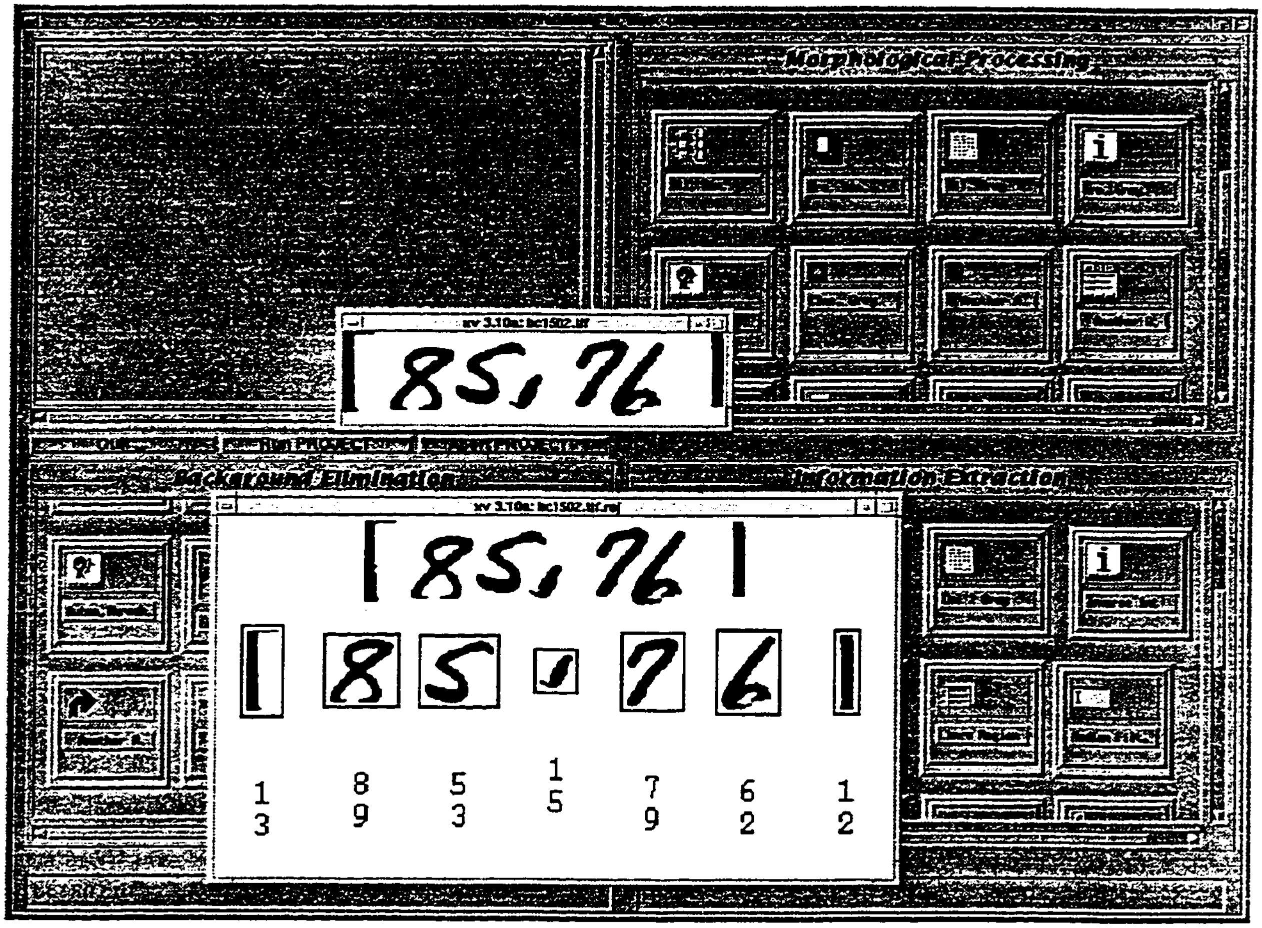

Figure 8.18: Extraction Difficulty: Failure to eliminate rectangular boxes (I.b) (continue).

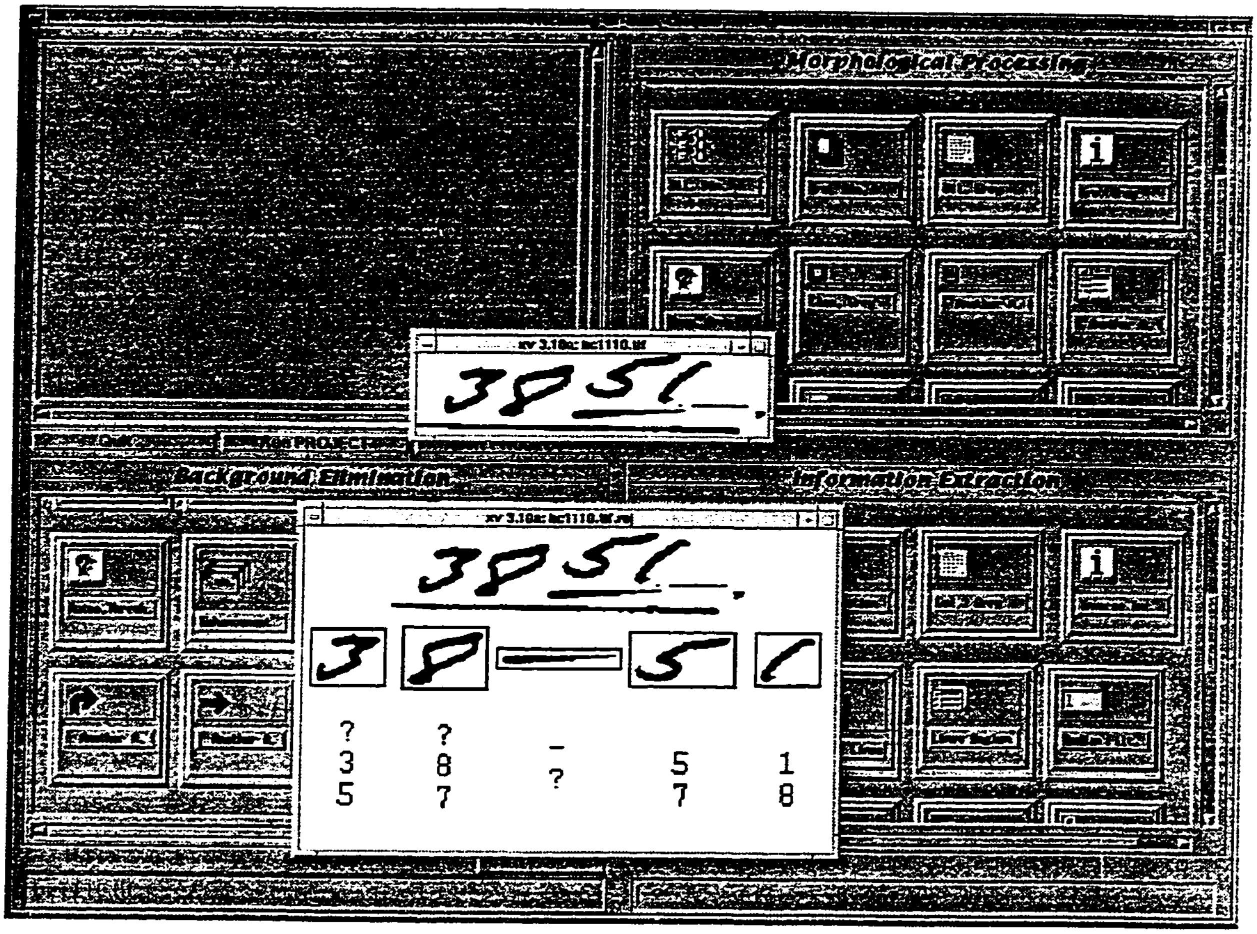

Figure 8.19: Extraction Difficulty: Failure to eliminate noise due to rotated cheque images (I.d).

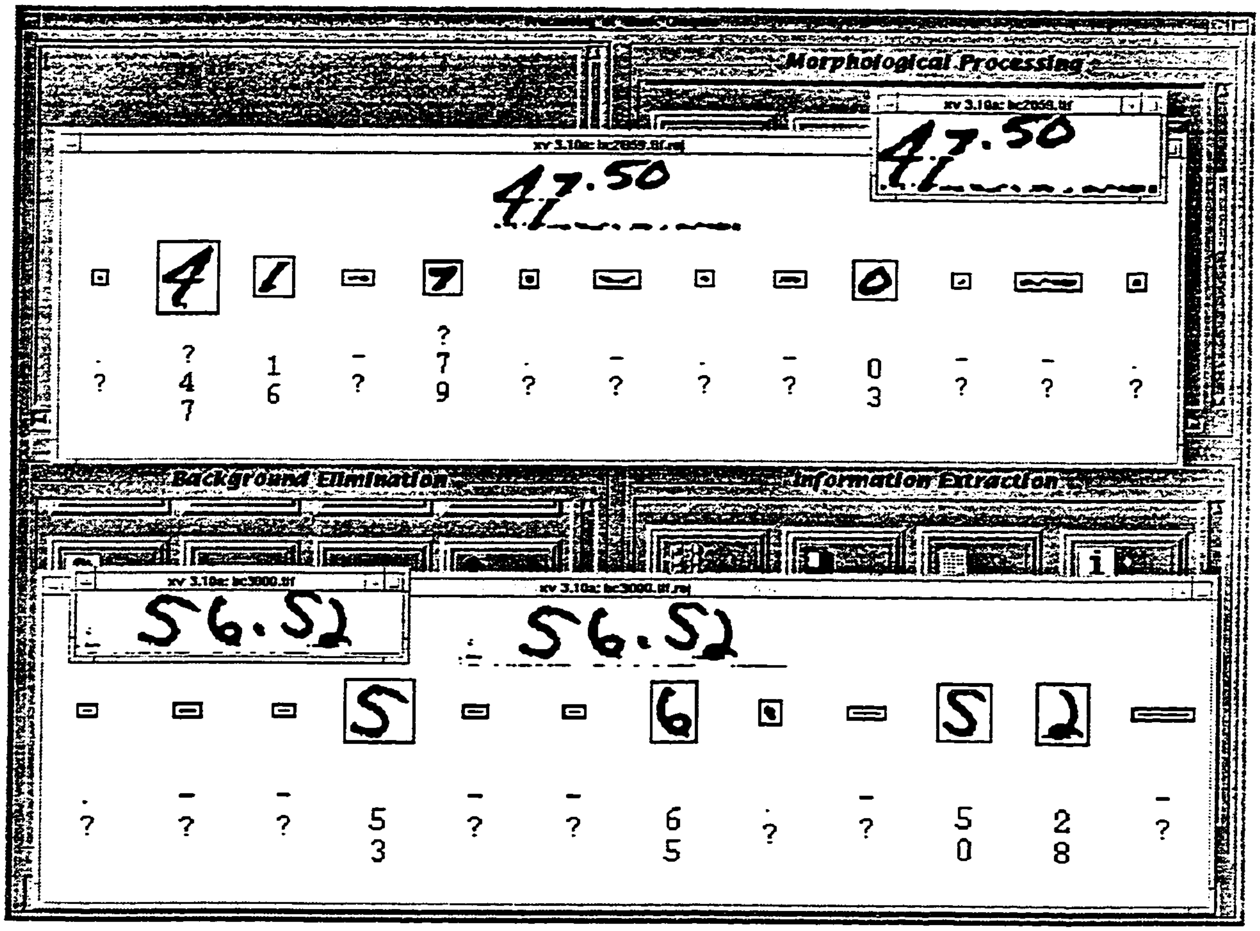

Figure 8.20: Extraction Difficulty: Failure to eliminate noise due to very dark background (I.e).

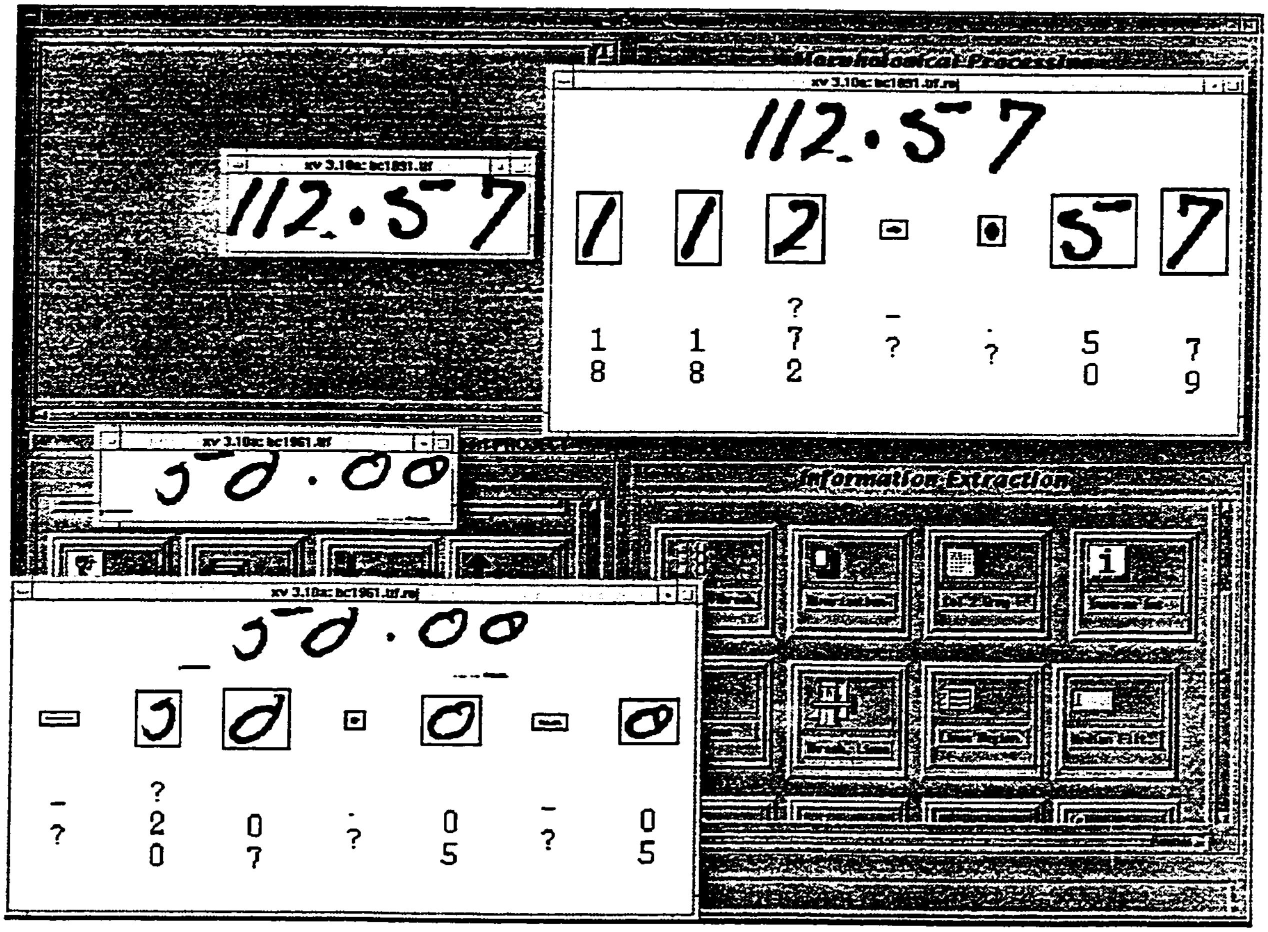

Figure 8.21: Extraction Difficulty: Failure to eliminate noise due to very dark background (I.e) (continue).

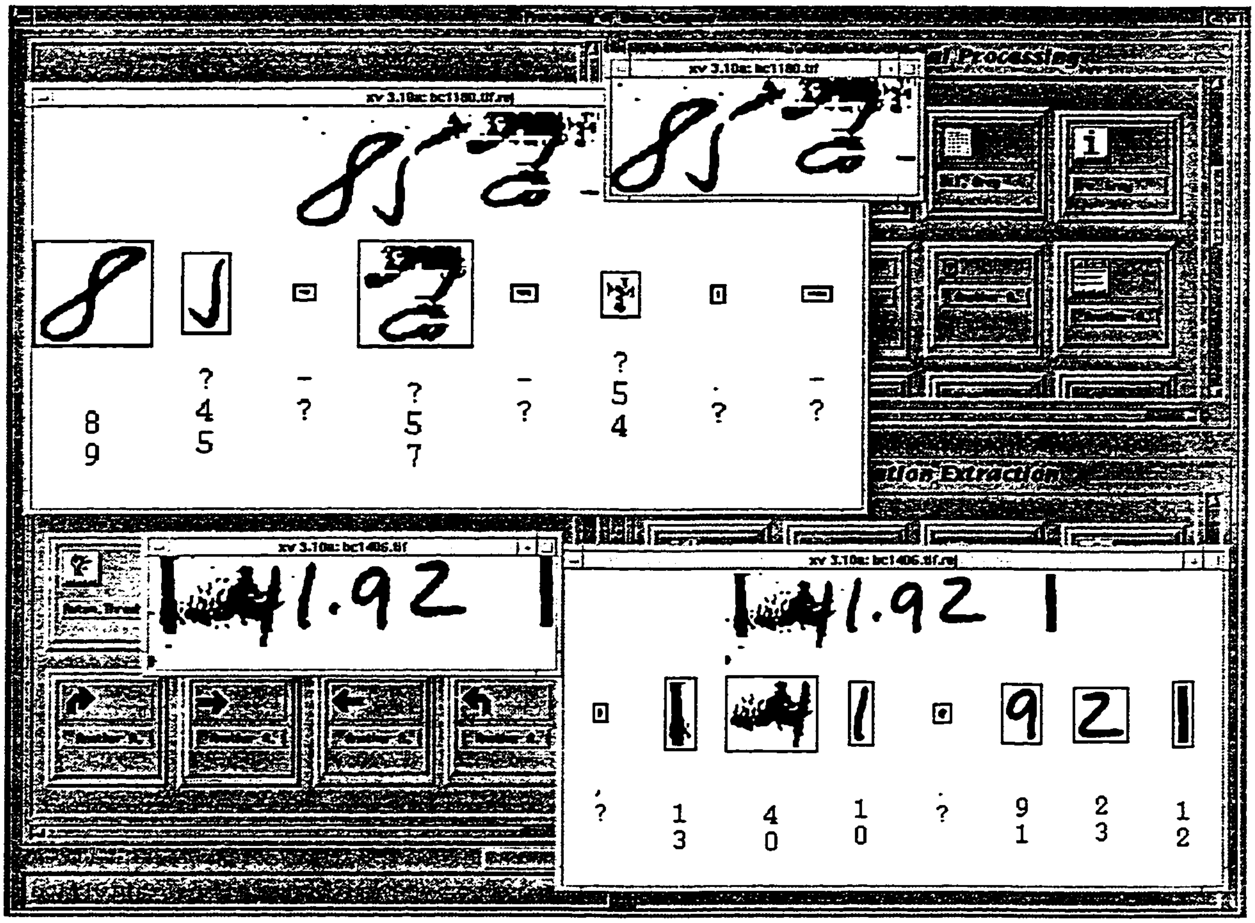

Figure 8.22: Extraction Difficulty: Failure to eliminate noise due to non-uniform cheques (I.f).

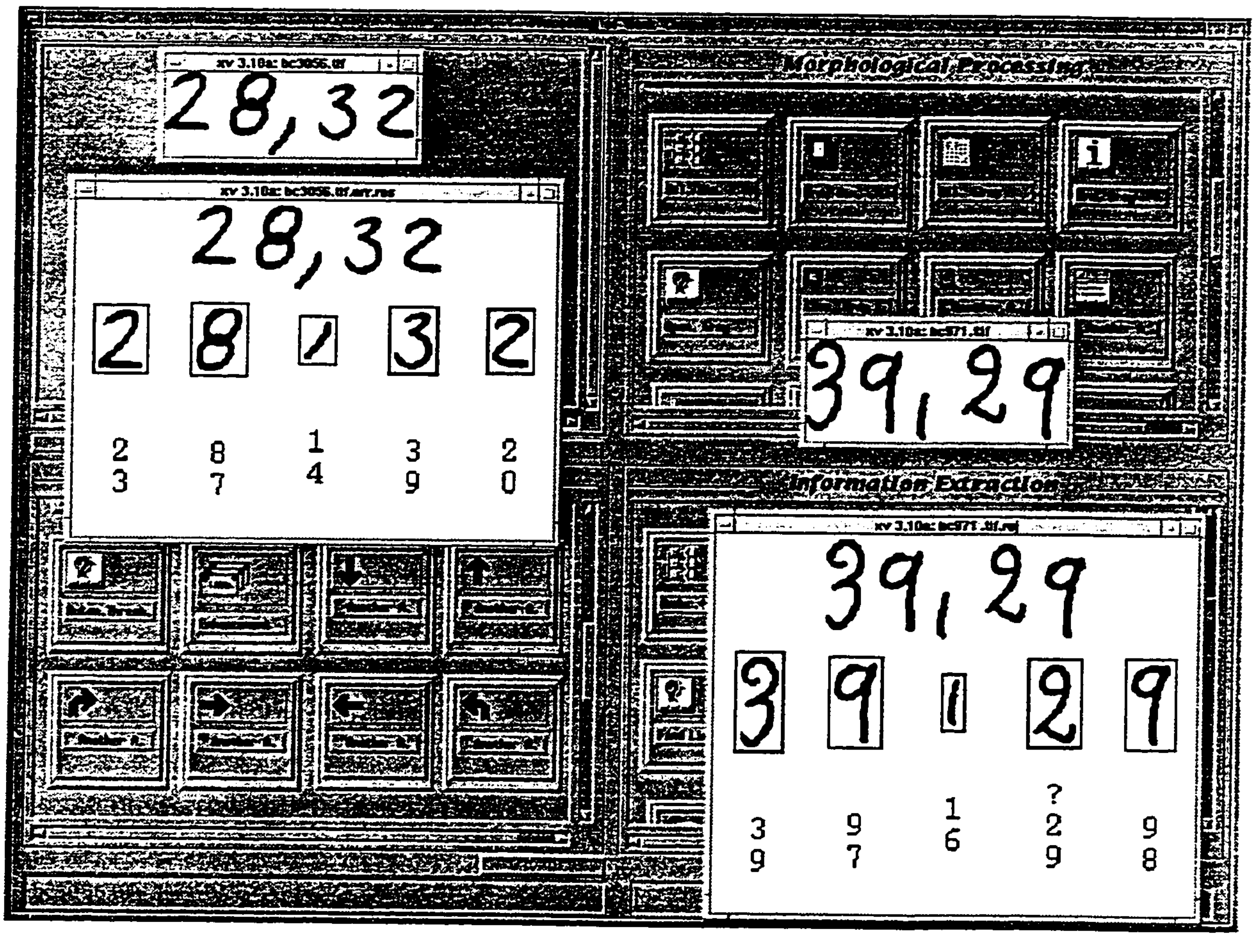

Figure 8.23: Recognition Difficulty: A "," is interpreted as a "1" (II.a.1).

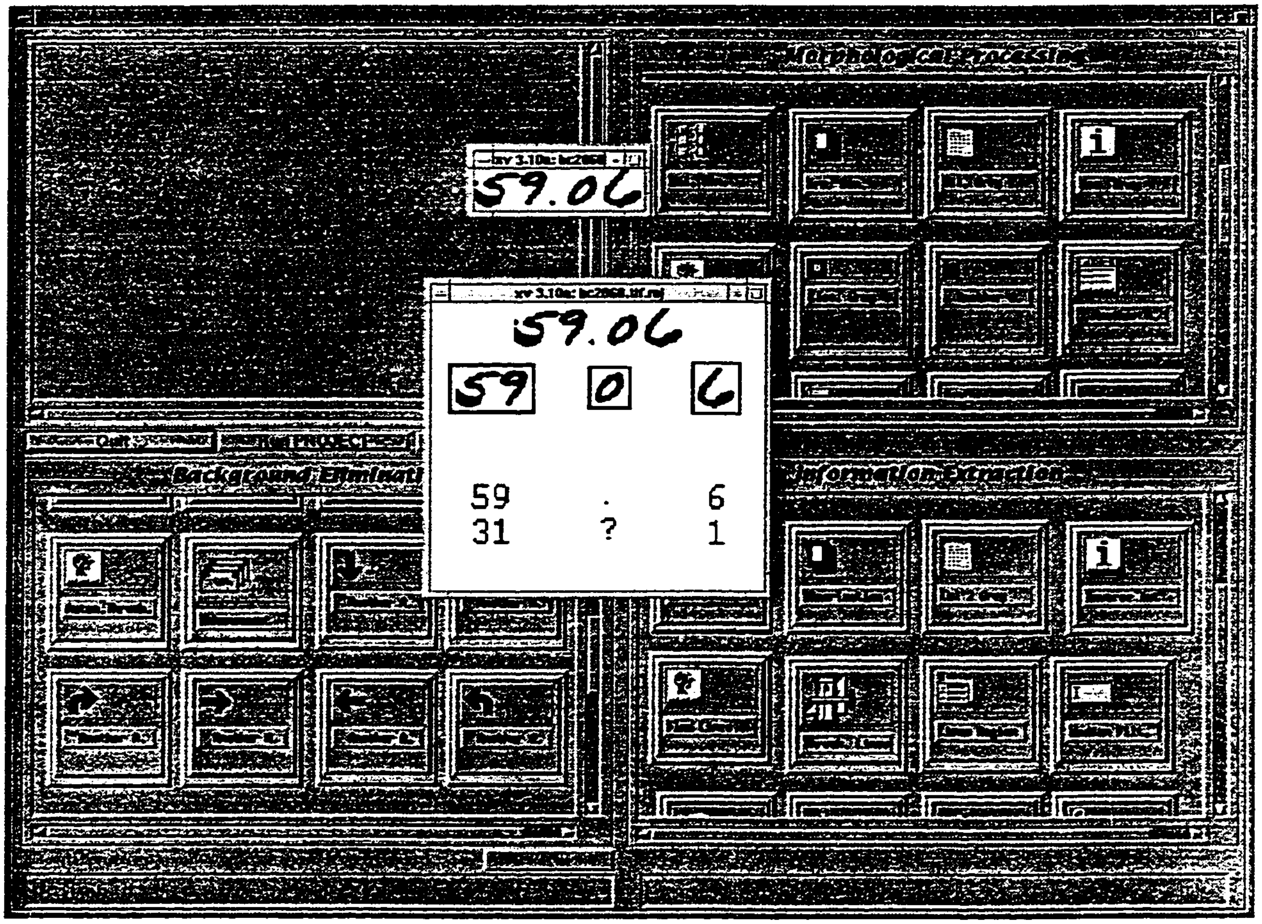

Figure 8.24: Recognition Difficulty: A "0" is interpreted as a "." (II.a.2).

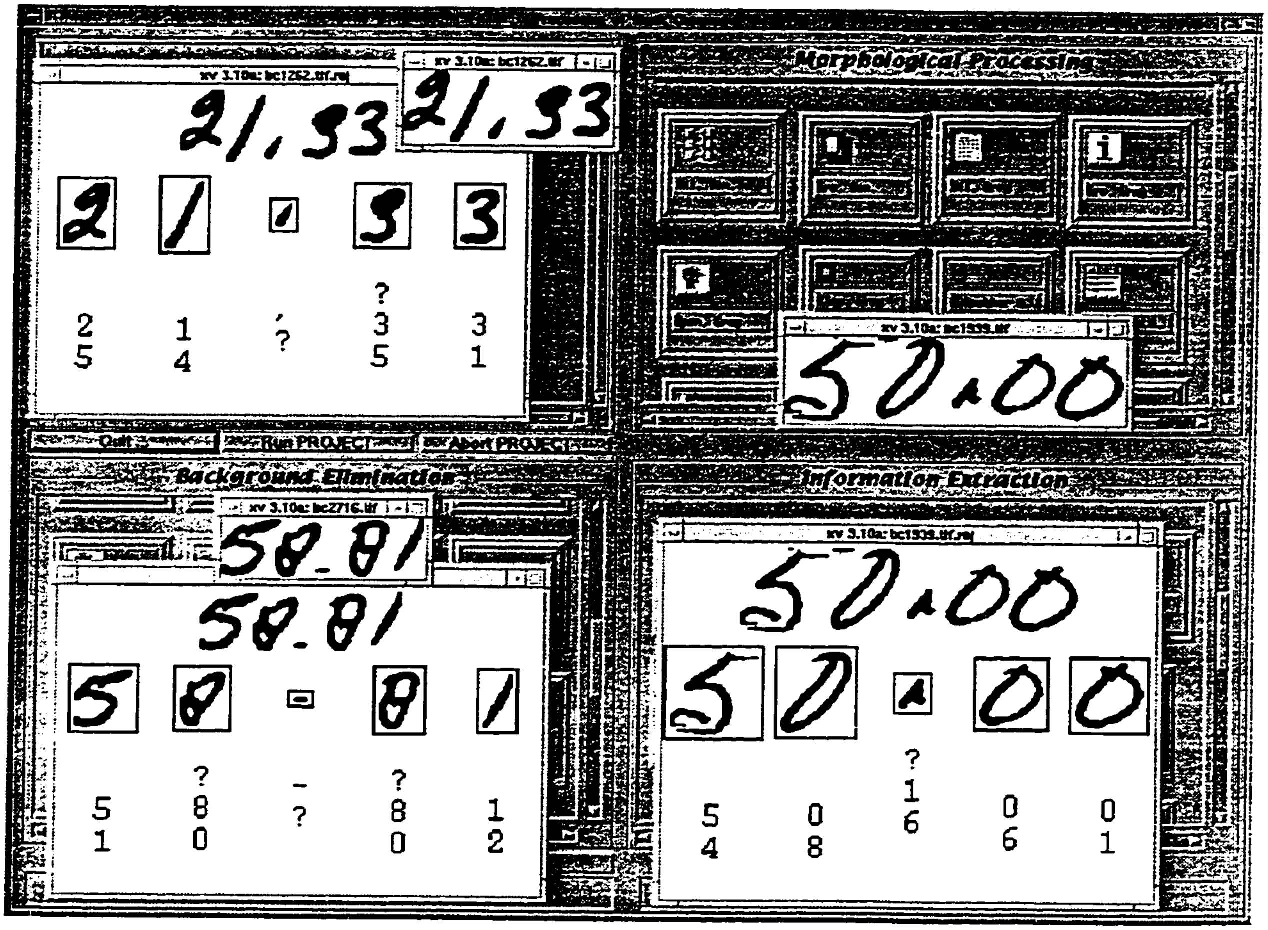

Figure 8.25: Recognition Difficulty: A "." is not recognized (II.a.3).

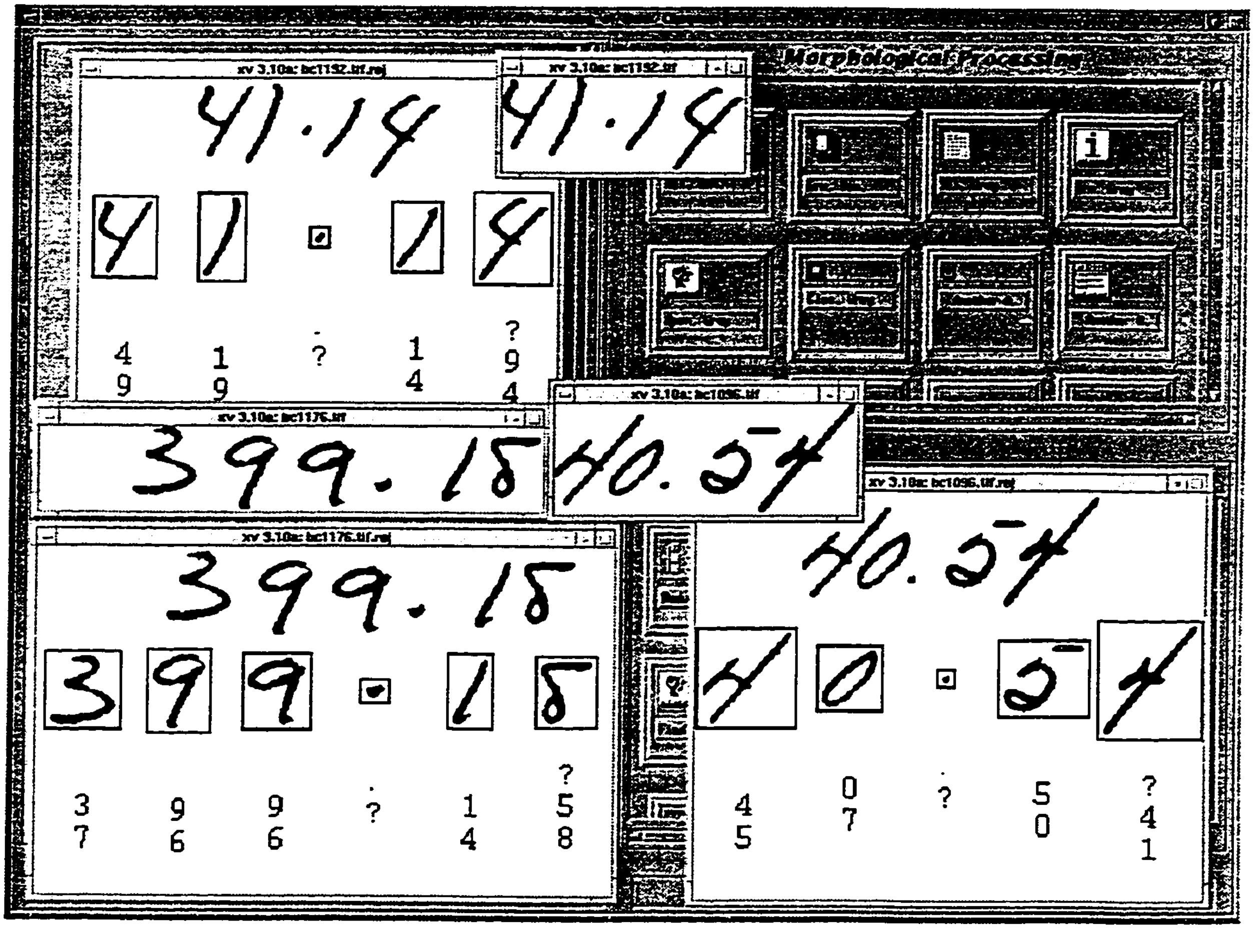

Figure 8.26: Recognition Difficulty: Characters are not properly recognized (II.a.4).

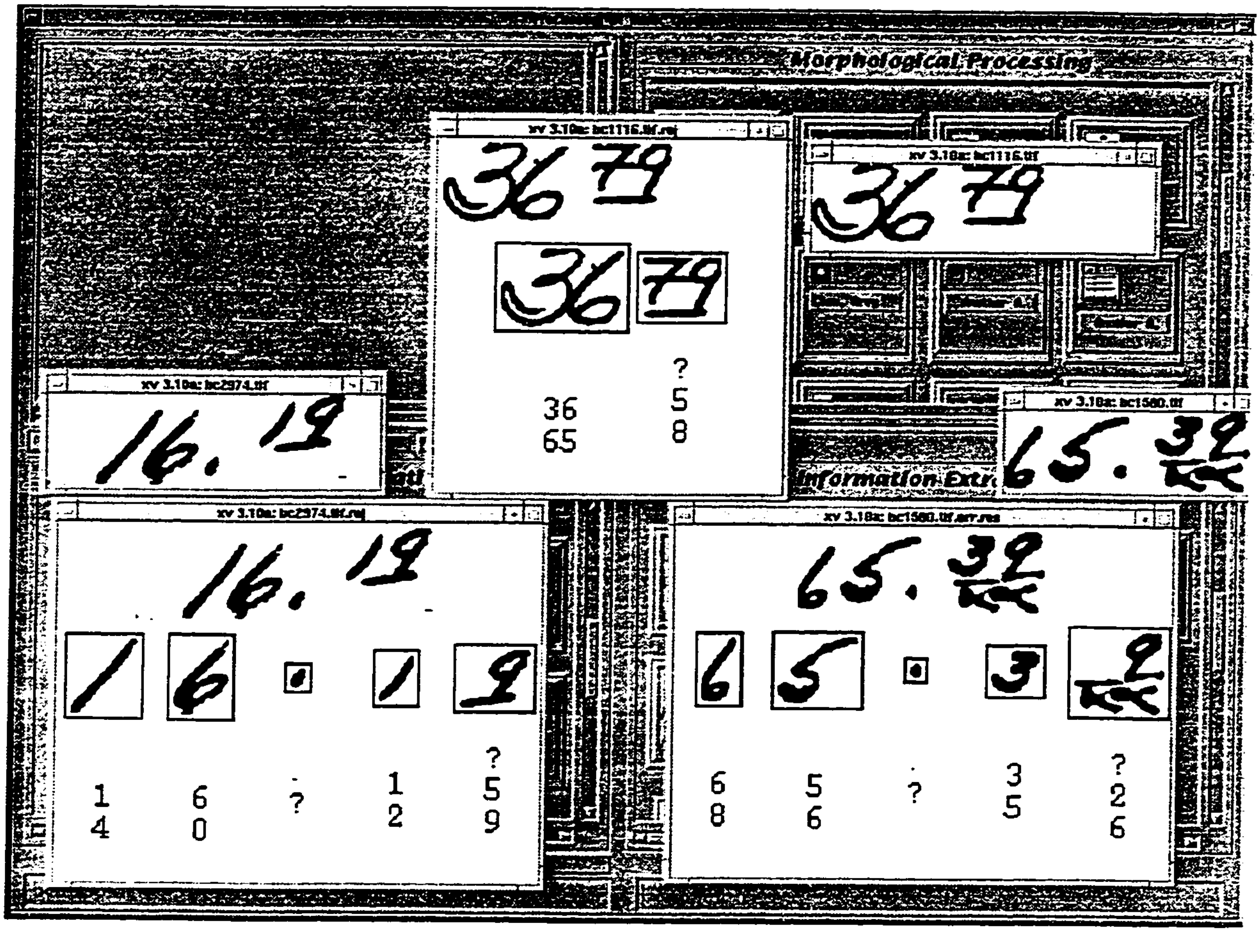

Figure 8.27: Recognition Difficulty: Dash lines under the decimal section (II.b.1).

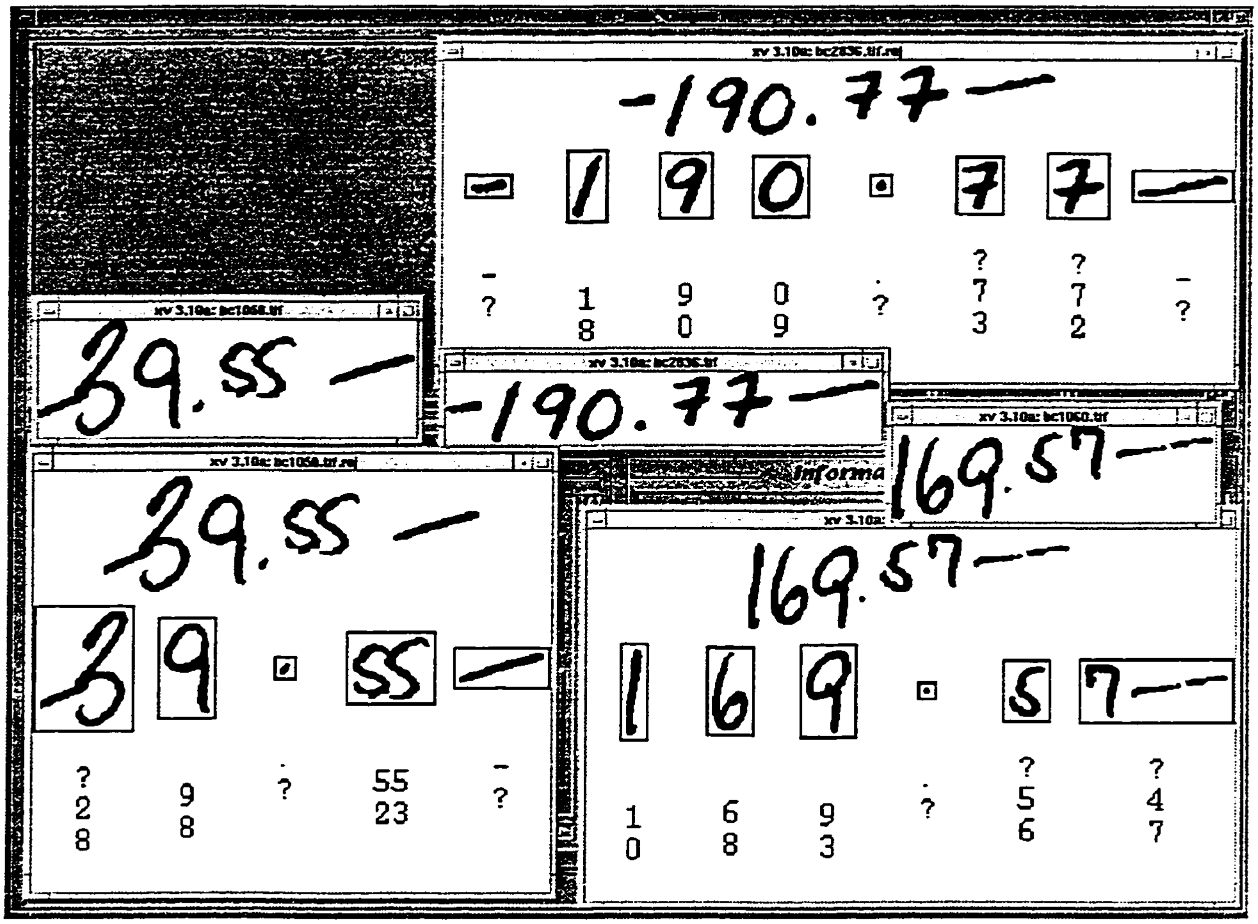

Figure 8.28: Recognition Difficulty: Dash lines before or after the courtesy amount (II.b.2).

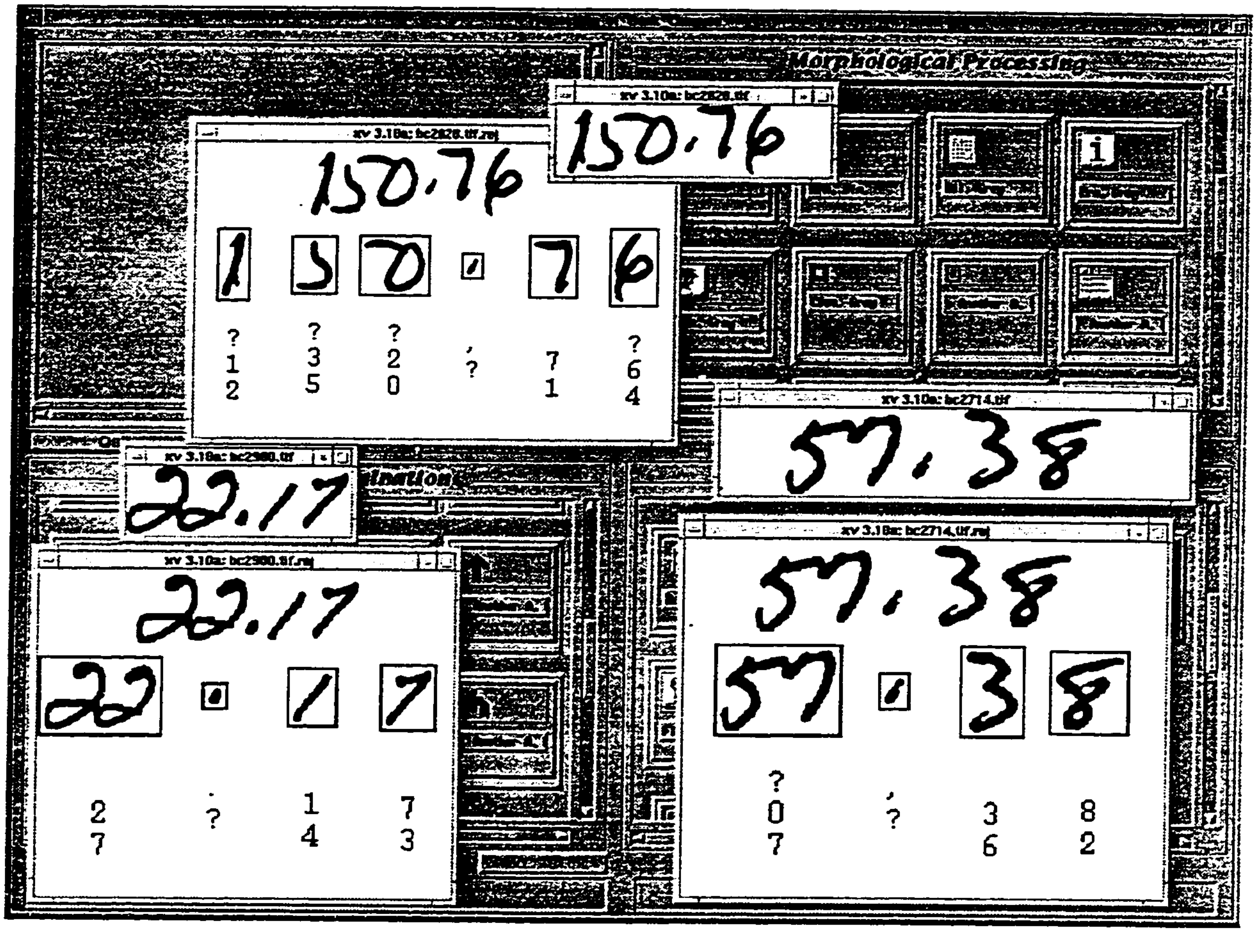

Figure 8.29: Recognition Difficulty: Error in segmenting touching digits (II.b.3).

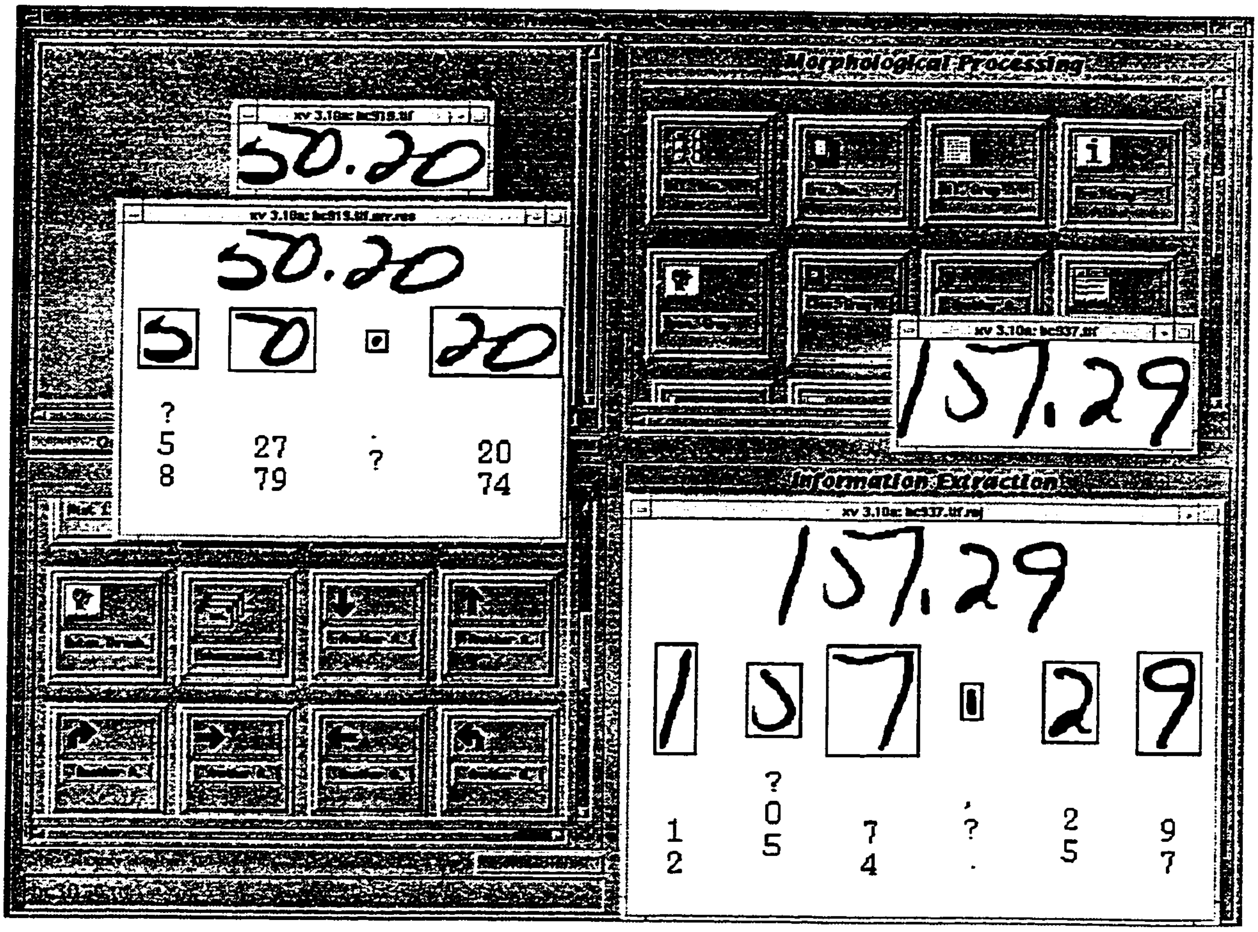

Figure 8.30: Recognition Difficulty: Error in interpreting disconnected digits fraction(s) (II.b.4).

TABLE XI						
Top-down Approach (Chapter 6) **Courtesy Amount Training Set of 500 images** **# Processed: 484** **Reliability: 89.20%** **Correct: 46.07% Wrong: 5.58% Rejected: 48.35%**						
Classification of Extraction Results						
	Common Errors %		Exclusive Errors %		Total %	
Error type	*Errors*	*Rejects*	*Errors*	*Rejects*	*Errors*	*Rejects*
I.a	0.00	2.48	0.83	1.45	0.83	3.93
I.b	0.20	2.27	0.21	1.45	0.41	3.72
I.c	0.00	2.07	0.00	1.65	0.00	3.72
I.d	0.20	2.48	0.41	1.45	0.61	3.93
I.e	0.00	1.24	0.00	0.82	0.00	2.06
I.f	0.63	4.13	0.00	0.41	0.63	4.54
I(Total)	1.03	14.67	1.45	7.23	2.48	21.9
Classification of Recognition Results						
II.a.1	0.00	0.82	0.21	0.20	0.21	1.02
II.a.2	0.00	0.00	0.00	0.00	0.00	0.00
II.a.3	0.00	1.44	0.00	0.20	0.00	1.64
II.a.4	1.03	9.51	1.42	2.69	2.45	12.2
II.a (Sub-total)	1.03	11.77	1.63	3.09	2.66	14.86
II.b.1	0.41	4.13	0.21	1.65	0.62	5.78
II.b.2	0.00	1.65	0.00	0.20	0.00	1.85
II.b.3	0.00	3.31	0.00	0.00	0.00	3.31
II.b.4	0.00	0.62	0.00	0.00	0.00	0.62
II.b (Sub-total)	0.41	9.71	0.21	1.85	0.62	11.56
II (Total)	1.44	21.48	1.84	4.94	3.28	26.42
Total (I+II)	2.47	36.15	3.29	12.17	5.76	48.32

TABLE XII						
Bottom-up Approach (Chapter 7) **Courtesy Amount Training Set of 500 images** **# Processed: 492** **Reliability: 91.35%** **Correct: 53.66% Wrong: 5.08% Rejected: 41.26%**						
Classification of Extraction Results						
	Common Errors %		Exclusive Errors %		Total %	
Error type	*Errors*	*Rejects*	*Errors*	*Rejects*	*Errors*	*Rejects*
I.a	0.40	2.03	0.41	0.20	0.81	2.23
I.b	0.40	2.03	0.20	0.00	0.60	2.03
I.c	0.00	2.23	0.00	0.00	0.00	2.23
I.d	0.00	2.64	0.00	0.20	0.00	2.84
I.e	0.00	1.22	0.00	0.20	0.00	1.42
I.f	0.82	3.46	0.00	0.41	0.82	3.87
I (Total)	1.62	13.61	0.61	1.01	2.23	14.62
Classification of Recognition Results						
II.a.1	0.21	0.61	0.00	0.00	0.21	0.61
II.a.2	0.00	0.00	0.00	0.00	0.00	0.00
II.a.3	0.00	1.88	0.00	0.61	0.00	2.49
II.a.4	1.01	9.95	0.40	3.86	1.62	13.78
II.a (Sub-total)	1.22	12.43	0.40	4.44	1.62	16.87
II.b.1	0.61	3.86	0.00	1.01	0.61	4.87
II.b.2	0.21	1.71	0.00	0.20	0.21	1.91
II.b.3	0.00	3.75	0.00	0.00	0.00	3.75
II.b.4	0.42	0.41	0.00	0.82	0.42	1.23
II.b (Sub-total)	1.23	9.73	0.00	2.03	1.23	11.76
II (Total)	2.45	22.17	0.40	4.47	2.85	26.64
Total (I+II)	4.07	35.78	1.01	5.48	5.08	41.26

TABLE XIII						
Top-down Approach (Chapter 6) **Courtesy Amount Testing Set1 of 200 images** **# Processed: 196** **Reliability: 87.91%** **Correct: 40.82% Wrong: 5.61% Rejected: 53.57%**						
Classification of Extraction Results						
	Common Errors %		Exclusive Errors %		Total %	
Error type	*Errors*	*Rejects*	*Errors*	*Rejects*	*Errors*	*Rejects*
I.a	0.00	3.57	1.02	3.57	1.02	7.14
I.b	0.00	2.55	0.00	1.53	0.00	4.08
I.c	0.00	2.04	0.00	1.02	0.00	3.06
I.d	0.00	1.53	0.00	2.55	0.00	4.08
I.e	0.00	1.53	0.51	0.51	0.51	2.04
I.f	0.00	2.04	0.00	1.53	0.00	3.57
I (Total)	0.00	13.26	1.53	10.71	1.53	23.97
Classification of Recognition Results						
II.a.1	0.00	1.55	0.00	0.00	0.00	1.55
II.a.2	0.00	0.00	0.00	0.00	0.00	0.00
II.a.3	0.00	1.53	0.00	0.51	0.00	2.04
II.a.4	1.02	9.69	0.51	1.02	1.53	10.71
II.a (Sub-total)	1.02	12.77	0.51	1.53	1.53	14.3
II.b.1	0.51	7.65	0.00	0.00	0.51	7.65
II.b.2	0.51	1.02	0.51	1.53	1.02	2.55
II.b.3	0.51	2.55	0.00	1.53	0.51	4.08
II.b.4	0.51	0.51	0.00	0.51	0.51	1.02
II.b (Sub-total)	2.04	11.73	0.51	3.57	2.55	15.3
II (Total)	3.06	24.50	1.02	5.10	4.08	29.6
Total (I+II)	3.06	37.76	2.55	15.81	5.61	53.57

TABLE XIV						
Bottom-up Approach (Chapter 7) **Courtesy Amount Testing Set1 of 200 images** **# Processed: 198** **Reliability: 91.30%** **Correct: 53.03% Wrong: 5.05% Rejected: 41.92%**						
Classification of Extraction Results						
	Common Errors %		Exclusive Errors %		Total %	
Error type	*Errors*	*Rejects*	*Errors*	*Rejects*	*Errors*	*Rejects*
I.a	0.50	2.52	0.00	1.02	0.50	3.54
I.b	0.50	2.02	0.00	0.00	0.50	2.02
I.c	0.00	2.02	0.00	0.00	0.00	2.02
I.d	0.00	1.51	0.00	0.00	0.00	1.51
I.e	0.00	1.51	0.00	0.50	0.00	2.01
I.f	0.00	2.02	0.00	0.50	0.00	2.52
I (Total)	1.00	11.60	0.00	2.02	1.00	13.62
Classification of Recognition Results						
II.a.1	0.51	2.03	0.00	0.00	0.51	2.03
II.a.2	0.00	0.00	0.00	0.00	0.00	0.00
II.a.3	0.00	1.52	0.00	0.50	0.00	1.52
II.a.4	1.01	8.08	0.50	3.55	2.02	11.63
II.a (Sub-total)	1.52	11.63	0.50	4.05	2.02	15.68
II.b.1	0.52	7.07	0.00	0.00	0.52	7.07
II.b.2	0.50	1.01	0.00	0.00	0.50	1.01
II.b.3	0.00	3.03	0.00	0.50	0.00	3.53
II.b.4	1.01	1.01	0.00	0.00	1.01	1.01
II.b (Sub-total)	2.03	12.12	0.00	0.50	2.03	12.62
II (Total)	3.55	23.75	0.50	4.55	4.05	28.3
Total (I+II)	4.55	35.35	0.50	6.57	7.08	41.92

TABLE XV					
Top-down Model (Chapter 6)			Bottom-up Model (Chapter 7)		
Courtesy Amount Training Set of 500 images 500 images					
Reliability: 89.20 %			Reliability: 91.35 %		
Correct %	*Errors %*	*Rejection %*	*Correct %*	*Error %*	*Rejection %*
46.07	5.58	48.35	53.66	5.08	41.26
Courtesy Amount Testing Set1 200 images					
Reliability: 87.91 %			Reliability: 91.30 %		
Correct %	*Errors %*	*Rejection %*	*Correct %*	*Error %*	*Rejection %*
40.82	5.61	53.57	53.03	5.05	41.92
Courtesy Amount Testing Set2 200 images					
Reliability: 90.10 %			Reliability: 93.10 %		
Correct %	*Errors %*	*Rejection %*	*Correct %*	*Error %*	*Rejection %*
47.89	5.26	46.84	54.82	4.06	41.12

8.5 Future Research Directions

In this thesis, a top-down system for the processing of bank cheques has been developed and qualitatively and quantitatively compared with another bottom-up system using the same training and testing data sets to verify its reliability and correctness. In testing the performance of the developed system we presented the extracted items, namely, the date, legal amount, and courtesy amount, to the date processing system, the legal amount processing system, and the courtesy amount processing system respectively.

8.5.1 A Feasibility Study Towards a New System

As a future step, a combined system must be developed. The combined system should be able to overcome the exclusive difficulties of both approaches that are presented in Chapters 6 and 7.

The system in Chapter 6 is fast and very efficient when a light simple or complex background exists against a light or dark handwriting. However, it is very sensitive to dark and complex backgrounds against a light handwriting. The system is very useful especially when the size of a document image is large.

The system in Chapter 7 is slower specially when larger document images are considered. Its extraction results are less sensitive to a light simple or complex background against a light or dark handwriting. However, the quality of its extracted items has less features for the recognizer. The line removal procedure adopted is very efficient and preserves the topological properties of the extracted items only at the intersection between the handwriting and the baselines that should be eliminated. In fact, better processing must be applied to preserve the topological properties of the handwritten information after the background is eliminated.

Having presented both systems and analyzed their corresponding quantitative and qualitative experimental results, it is very helpful to study the feasibility of combining these systems into one system. A good approach in pursuing this line is to conduct a study with the purpose is to find the set of common and exclusive difficulties of both item extraction systems. If both systems have the same extraction difficulties under the same cases, then combining them is not favourable because none of them can help to reduce the mistakes of the other. However, when each system overcomes partially or completely the exclusive difficulties of the other, combining them is favourable and will lead to a more reliable system that is able to overcome the exclusive difficulties of both systems. Of course, there remains the set of common difficulties that both item extraction systems do not produce successful results.

TABLE XVI illustrates a feasibility study that shows an expected performance of the combined systems. Of course, the expected reliability of the combined system is higher than the reliability of both systems in separate. This is true because these separate systems have a good number of exclusive errors and most of these errors could be overcome when both systems collaborate. From **TABLE XVI** one can conceive the set of difficulties that both item extractions have and this requires further research.

TABLE XVI				
Top-down & Bottom-up (combined) Courtesy Amount Training Set of 500 images Reliability: 97.09% Correct: 40.45% Wrong: 1.21% Rejected: 58.33%				
Courtesy Amount Testing Set1 of 200 images Reliability: 96.06% Correct: 36.87% Wrong: 1.51% Rejected: 61.62%				
Classification of Extraction Results				
	Training Set		Testing Set1	
Error type	*Errors %*	*Rejects %*	*Errors %*	*Rejects %*
I.a	0.00	4.47	0.00	7.58
I.b	0.00	4.06	0.00	3.54
I.c	0.00	3.87	0.00	3.03
I.d	0.00	4.26	0.00	4.54
I.e	0.00	2.24	0.00	3.03
I.f	0.61	4.88	0.00	3.54
I (Total)	0.61	23.78	0.00	25.26
Classification of Recognition Results				
II.a.1	0.00	1.01	0.00	1.52
II.a.2	0.00	0.00	0.00	0.00
II.a.3	0.00	2.64	0.00	2.52
II.a.4	0.20	17.08	0.51	14.65
II.a (Sub-total)	0.20	20.73	0.51	18.69
II.b.1	0.20	6.51	0.00	7.58
II.b.2	0.00	1.83	0.50	2.52
II.b.3	0.00	4.26	0.00	5.05
II.b.4	0.20	1.22	0.50	2.52
II.b (Sub-total)	0.40	13.82	1.00	17.67
II (Total)	0.60	34.55	1.51	36.37
Total (I+II)	1.21	58.33	1.51	61.62

To produce the expected reliability from the feasibility study conducted we considered the judgement of the courtesy amount processing system. The courtesy amount processing system takes as input an extracted courtesy amount and gives as output as one of the following three possible results: (1) a correct and well recognized image, (2) an error image, and (3) a rejected image. In combining the two item extraction system we assumed the following rules:

- An extracted courtesy amount image is considered a correct image if the courtesy amount processing system classified this image as being correct for both item extraction systems.

- An extracted courtesy amount image is considered an error image if the courtesy amount processing system classified this image as being an error image for both item extraction systems.

- An extracted courtesy amount image is considered a rejected image if the courtesy amount processing system classified this image as being a rejected image for any item extraction system.

8.5.2 A Need for Further Quantitative Analysis

As to the extracted date and legal amounts, a quantitative analysis must be done in order to identify the set of extraction difficulties caused by the item extraction system and the set of recognition difficulties caused by the corresponding item processing system. In this sense, one can quantitatively define the reliability of an item extraction system based on the feedback from the corresponding recognizer adopted. This, of course, will be very helpful for both the extractor and its corresponding recognizer. In other words, the item extraction system will be better trained to extract its items and enhance its quality and the corresponding item processing system adopted will be better trained to enhance its recognition ability.

From such a feasibility study, one can conceive many challenges. The detection of intruding strokes (from adjacent text lines) is one difficulty that should be overcome. The different recognizers are poorly positioned to determine the origins of strokes in each view window, and can only recognize or reject the entire object based on their respective criteria. Ultimately, some intruding strokes are expected to be removed at the item-extraction level, where the procedures have to be improved and also made efficient. The courtesy amount processing system should be better trained to enhance its recognition ability to differentiate between a "1" and a "7" and enhance its ability to segment touching digits. Feedback from the recognizers must be available to the item extraction systems. The segmentation system must be able to provide alternative strategies upon receiving feedback or rejection from the recognizers.

Underlying all these problems are other human factors-the variability in writing styles and the large assortment of cheques for different users. Consequently, the problem has considerable magnitude, encompassing different forms of handwriting in different contexts, and it is clear that much effort would be needed to arrive at a satisfactory solution to a problem of such a scope.

Bibliography

[1] L. Abele, F. Wahl, and W. Scheri, "Procedures for an Automatic Segmentation of Text Graphic and Halftone Regions in Documents," *Proc. 2nd Scandinavian Conf. on Image Analysis,* 1981, pp. 177-182.

[2] A. Agarwal, L. Granowetter, K. Hussein, A. Gupta, and P.S.P. Wang. "Detection of Courtesy Amount Block on Bank Checks," *Proc. of the Third International Conference on Document Analysis and Recognition (ICDAR'95,* Montreal, Canada, 1995, pp. 748 751.

[3] T. Akiyama and I. Masuda, "A Segmentation Method for Document Images Without the Knowledge of Document Formats," *Trans. Japanese Institute of Electronics and Communication Engineerings,* Vol. J66-D, No. 2, pp. 111-118, 1982.

[4] T. Akiyama and N. Hagita, "Automated Entry System for Printed Documents," *Pattern Recognition,* Vol. 23, No. 11, 1990, pp. 1141-1151.

[5] M. Amadasun and R. King, "Textural Features Corresponding to Textural Properties," *IEEE Trans. on Systems, Man, and Cybernetics,* Vol. 19, No. 5, pp. 1264-1274, 1989.

[6] T. Amano *et al.*, "DRS: A Workstation-based Document Recognition System for Text Entry," *Computer*, Vol. 25, No. 7, July 1992, pp. 67-71.

[7] M. Ammar, Y. Yoshida and T. Fukumura, "Automatic Off-line Verification of Signatures Based on Pressure Features," *Proc. Tech. Group. meeting of Pattern Recognition Learn. of IECEJ,* Tokyo (1985).

[8] A. Andronico and A. Ciampa, "On Automatic Pattern Recognition and Acquisition of Printed Music," *In Proceedings, International Computer Music Conference,* pp. 245-278, Venice, Italy 1982, Computer Music American Publications.

[9] R. N. Ascher, G. M. Koppelman, M. J. Miller, G. Nagy, and G. L. Shelton Jr, "An Interactive System for Reading Unformatted Printed Text," *IEEE Trans. on Computers,* Vol. C-20, No. 12, pp. 1527-1543, 1971.

[10] H. S. Baird and K. Thompson, "Reading chess," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 12, no 6, June 1990, pp. 552-559.

[11] H. S Baird, "Anatomy of a Page Reader," *Proc. IAPR Workshop on Machine Vision Applications*, Tokyo, Japan , 1990.

[12] D. H. Ballard and C. M. Brown, Computer Vision, Prentice-Hall, Englewood, Cliffs, N. J., 1982.

[13] A. Belaid, J. C. Anigbogu, and Y. Chenvoy, "Qualitative Analysis of Low-level Logical Structures," *Electronic Publishing Organization, Dissemination and Design*, 6(4): 435-446, December 1993.

[14] A. Bleau, J. De Guise, and A. Robert Le Blanc, "A New Set of Fast Algorithms for Mathematical Morphology," *CVGIP: Image Understanding*, vol. 56, No. 2, pp. 178-209, September 1992.

[15] D. S. Bloomberg, "Multiresolution Morphological Approach to Document Image Analysis," *Proc. First Int. Conf. on Document Analysis and Recognition*, Saint-Malo, France, Sept. 30 - Oct. 2, 1991, pp. 963-971.

[16] S. Boukharouba, J. M. Rebordao, and P. L. Wendel, "An Amplitude Segmentation Method Based on the Distribution Function of an Image," *Compt. Vision Graphics Image Process.*, vol. 29, pp. 47-59, 1985.

[17] M. Brooks, "Rationalizing Edge Detection," *Computer Graphics Image Processing* vol. 8, pp. 277-285, 1978.

[18] H. Brown, "Standards for Structured Documents," *The Computer Journal*, 32(6):505-514, 1989.

[19] M. Bryan, SGML: An Author's Guide to the Standard Generalized Markup Language, Addison - Wesley, 1988.

[20] H. Bunke and A. Sanfeliu, Syntactic and Structural Pattern Recognition Theory and Applications. World Scientific, 1990.

[21] M. J. Carlotto, "Histogram Aalysis Using Scale-space Approach," *IEEE Trans. Pattern Anal. Mach. Intell.* vol. 9 (1), pp. 121-129, 1987.

[22] R. Casey and G. Nagy, "Recursive Segmentation and Classification of Composite Character Patterns," *Proc. 6th Int. Conf. Pattern Recognition*, (Munich, Germany), 1982, pp. 1023-1026.

[23] R. G. Casey and G. Nagy, "Document Analysis: A Broader View," *Proc. First Int. Conf. on Document Analysis and Recognition*, Saint-Malo, France, Sept. 30 - Oct. 2, 1991, pp. 839-850.

[24] R. G. Casey, D. R. Ferguson, K. M. Mohiuddin, and E. Walach, "An Intelligent Forms Processing System," *Machine Vision and Applications*, vol. 5, no. 3, pp. 143-155, 1992.

[25] N. B. Carter, "Automatic Recognition of Printed Music in Context of Electronic Publishing," PhD thesis, Univ. of Surrey, Depts. of Physics and Music, February 1989, 174 pages.

[26] N. B. Carter and R. A. Bacon, "Automatic Recognition of Printed Music," *Structured Document Image Analysis*, 1992, pp. 456-465.

[27] Y. Chenevoy and A. Belaid, "Hypothesis Management for Structured Document Recognition," *Proceedings of the first International Conference on*

Document Analysis and Recognition, pp. 121-129, Saint-Malo, France, September 1991.

[28] M. Cheriet, J. N. Said, and C. Y. Suen, "A Formal Model for Document Processing of Bank Cheques," *third International Conference on Document Analysis and Recognition, Proc. ICDAR'95*, Montreal, Aug 24-26, 1995 pp. 210-213.

[29] M. Cheriet, J. N. Said, and C. Y. Suen, "A Recursive Approach for Image Segmentation: Extended Otsu's Approach," to appear in *IEEE Trans. on Image Processing.* (accepted)

[30] D. K. Y. Chiu, C. Y. C. Bie, and H. C. Shen. "Hierarchical Maximum Entropy Partitioning of Visual Feature Frequency Matrices for Texture Classification," *Proc. Vision Interface'92*, 1992, pp. 9-14.

[31] D. K. Y. Chiu, B. Cheung, and A. K. C. Wong. "Information Synthesis Based on Hierarchical Maximum Entropy Partitioning," *J. Expt. Theor. Artif. Intell.* Vol. 2, pp. 117-129, 1990.

[32] G. Ciardiello *et al.*, "An Experimental System for Office Document Handling and Text Recognition," *Proceedings of the International Conference on Pattern Recognition*, page 739-743, 1988.

[33] M. Clark and A. C. Bovik. "Experiments in Segmenting Text on Pattern Using Localized Spatial Filters," *Pattern Recognition*, Vol. 22, No. 6, pp. 707-717, 1989.

[34] M. Cohen and G. Toussaint, "On the Detection of Structures in Noisy Pictures," *Pattern Recognition*, vol. 9, pp. 95-98, 1977.

[35] D. Cooper and H. Elliot, "A Maximum Likelihood Framework for Boundary Estimation in Noisy Images," *Proc. IEEE Computer Society Conf. on Pattern Recognition and Image Processing*, (Chicago), May 31-June 2, 1978, pp. 25-31.

[36] R. C. Gonzalez and M. G. Thomason, Syntactic Pattern Recognition - An Introduction, Addison-Wesley Publishing Company, 1978.

[37] M. Coster and J. L. Chermant, "Precis d'Analyse d'Images," Presses du CNRS, Paris, 1989.

[38] L. Davis, M. Clearman, and J. K. Aggarwal, "A Comparative Texture Classification Study Based on Generalized Cooccurrence Matrices," *Proc. IEEE Conf. on Decision and Control*, (Miami), Dec. 12-14, 1979.

[39] L. Davis, S. Johns, and J. K. Aggarwal, "Texture Analysis Using Generalized Cooccurrence Matrices," *Proc. IEEE Trans. Pattern Anal. Machine Intelligence*, vol. PAMI-1, pp. 251-258, 1979.

[40] A. Dengel *et al.* "From Paper to Office Document Standard Representation." *Computer*, Vol. 25, No. 7, July 1992, pp. 63-67.

[41] A. Dengel, "IIODA: The Paper Interface to ODA. Technical Report," The German Search Centre in Artificial Intelligence, 1992.

[42] A. Dengel, "A Step Towards Understanding Paper Documents. Technical Report," German Search Centre in Artificial Intelligence, 1990.

[43] A. Dengel, "Document Image Analysis Expectation Driven Text Recognition," *Proc. of Syntactic and Structural Pattern Recognition (SSPR90)*, 1990, pp. 78-87.

[44] A. Dengel and G. Barth. "High Level Document Analysis Guided by Geometric Aspects," *International Journal of Pattern Recognition and Artificial Intelligence,* Vol. 2, No. 4, 641-655, 1988.

[45] D. Guillevic, "Unconstrained Handwriting Recognition Applied to The processing of Bank Cheques," PhD thesis, Univ. of Surrey, Depts. of Physics and Music, February 1989, 174 pages.

[46] W. Doster, "Different States of a Document's Content on its Way From The Gutenburgian World to The Electronic World," *Proc. 7^{th} Int. Conf. on Pattern Recognition,* 1984, pp. 872-874.

[47] E. R. Doughert, An Introduction to Morphological Image Processing, Vol. TT9, SPIE Press, Bellingham, Washington, U.S.A., 1992

[48] A. C. Downton and C. G. Leedham, "Pre-processing and Presorting of Envelope Images for Automatic Sorting Using OCR," *Pattern Recognition*, Vol. 23, No. 3/4, pp. 347-362, 1990.

[49] D. G. Elliman and I. T. Lancaster. "A Review of Segmentation and Contextual Analysis Techniques for Text Recognition," Pattern Recognition, Vol. 23, No. 3/4, pp. 337-346, 1990.

[50] F. Esposito, D. Malerba, and G. Semerato, "An Experimental Page Layout Recognition Systems for Office Document Automatic Classification: an Integrated Approach for Inductive Generation," *Proc. Pattern Recognition* (Atlantic City, NJ), 1990, pp. 557-562.

[51] F. Farrokhnia, "Multi-channel Filtering Techniques for Texture Segmentation and Surface Quality Inspection," Ph. D. thesis, Dept. of Electrical Eng., Michigan State University, 1990.

[52] F. Farrokhnia and A. K. Jain. "A multi-channel Filtering Approach to Texture Segmentation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* Maui, 1991, pp. 364-370.

[53] J. L. Fisher. "Logical Structure Description of Segmented Document Images," *Proc. First Int. Conf. on Document Analysis and Recognition*, Saint-Malo, France, Sept. 30 - Oct. 2, 1991, pp. 302-310.

[54] L. A. Fletcher and R. Kasturi, "A Robust Algorithm for Text String Separation From Mixed Text/Graphics Images," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* Vol. 10, No. 6, pp. 910-918, 1988.

[55] J. L. Fisher, S. C. Hinds, and D. P. D'Amato, "A Rule-Based System for Document Image Segmentation," *Proc. 10th Int'l Conf. Pattern Recognition (ICPR),* IEEE CS Press, Los Alamitos Calif., 1990, pp. 567-572.

[56] K. S. Fu, Syntactic Pattern Recognition and Applications. Prentice Hall Inc., 1982.

[57] K. Fukunaga, *Introduction to Statistical Pattern Recognition.* New York:Academic, 1972, pp. 260-267.

[58] H. Fujihara and A. Mukerjee. "Document Recognition Using Qualitative Reasoning," *Proc. Symposium on Document Analysis and Information Retrieval,* March 16-18, 1992, pp. 316-331.

[59] I. Fujinaga, "Optical Music Recognition Using Projections," Master's thesis, McGill University, Faculty of Music, Montreal, Canada, September 1988. For an M.A. im Music Theory.

[60] I. Fujinaga, B. Alphonce, and B. Pennycook, "Issues in the Design of an Optical Music Recognition System," *Proceedings, 1989 International Computer Music Conference.* Columbus, Ohio, 2-5 November 1989.

[61] I. Fujinaga, B. Alphonce, B. Pennycook, and N. Boisvert, "Optical Recognition of Musical Notation by Computer," *Computers in Music Research Newsletter,* Nov. 1, 1989.

[62] H. Fujisawa and Y. Nakano, "A Top-down Approach to the Analysis of Document Images," In H.S. Baird, H. Bunke, and K. Yamamoto, editors, *Structured Document Image Analysis,* pages 99 - 114, Springer - Verlag, 1992.

[63] H. Fujisawa, "Artificial Intelligence as Applied to Optical Image Filing," *Proc. Int'l Symp. On Optical Memory,* Tokyo, 1987, also in Japanese Journal of Applied Physics, Vol. 26, Supplement 26-4, 1987.

[64] R. Futrelie, M. J. Pescitelli, Jr. J. S. Alexander, M. E. Cleary, J. D. Crisman, C. E. Dunn, D. S. Ellis, J. Gauch, I. A. Kakadiaris, S. Mukherjea, D. R. Nadeau, and N. Nikolakis, "Document Analysis, Understanding and Knowledge Access," Proc. First Int. Conf. on Document Analysis and Recognition. Saint-Malo, France, Sept. 30 - Oct. 2, 1991, pp. 101-111.

[65] D. Gabor, "Theory of Communication," Journal Inst. Elect. Engr., Vol. 93, pp. 429-457, 1946.

[66] C. F. Goldfarb, *The SGML Handbook,* Clarendon Press, Oxford, United Kingdom, 1990.

[67] R. C. Gonzalez and P. Wintz, Digital Image Processing, Addison-Wesley Publishing Company, Reading, Massachusetts, 1987.

[68] T. Gude and V. Margner. "Automatic Interpretation of Graphical Drawings," *Proc. First Int. Conf. on Document Analysis and Recognition,* Saint-Malo, France, Sept. 30 - Oct. 2, 1991, pp. 656-664.

[69] D. Guillevic and C. Y. Suen, "Cursive Script Recognition Applied to the Processing of Bank Cheques," *Proc. of the Third International Conference on Document Analysis and Recognition (ICDAR'95),* Montreal, Canada, 1995, pp. 11-14.

[70] D. Guillevic and C. Y. Suen, "Cursive Script Recognition: A Sentence Level Recognition Scheme," in *International Workshop on Frontiers of Handwritten Recognition,* Taipei, Taiwan, December 1994, pp. 216-223.

[71] D. Guillevic and C. Y. Suen, "Cursive Script Recognition: A Fast Reader Scheme," in *Proc. of the International Conference on Document Analysis and Recognition,* Tsukuba Science City, Japan, October 1993, pp. 311-314.

[72] A Gupta W. V. Nagendraprasad, A. Liu, P.D.P. Wang and S. Ayyadurai, "An Integrated Architecture for Recognition of Totally Unconstrained Handwritten Numerals," *International Journal of Pattern Recognition and Artificial Intelligence,* Vol. 7, No. 4 (1994) pp. 757-773.

[73] T. M. Ha and H. Bunke, "Model-based Analysis and Understanding of Cheque Forms," *International Journal of Pattern Recognition and Artificial Intelligence,* Vol. 8, No. 5 (1994) 1053-1080.

[74] A. Habibi, "Two-dimensional Bayesian Estimate of Images," *Proc. IEEE.* vol. 60, pp. 878-883, 1972.

[75] N. Hagita I. Masuda, T. Akiyama, T. Takahashi, and S. Naito. "Approach to Smart Document Reader System," *Proc. CVPR '85,* 1985, pp. 550-557.

[76] R. M. Haralick and I. Dinstein, "A Spatial Clustering Procedure for Multi-image Data," *IEEE Trans. Circuits and Systems* CAS-22, No. 5 (1975) 440-450.

[77] D. C. He and L. Wang. "Detecting Texture Edges from Image," *Pattern Recognition,* Vol. 25, No. 6, pp. 595-600, 1992.

[78] D. C. He and L. Wang. "Texture Features Based on Texture Spectrum." *Pattern Recognition,* Vol. 24, No. 5, pp. 391-399, 1991.

[79] L. Hertz and R.W. Schafer, "Multilevel Thresholding Using Edge Matching," *Compt. Vision Graphics Image Process,* vol 44, pp. 279-295, 1988.

[80] J. Higashino, H. Fujisawa, Y. Nakano, and M. Ejiri, "A Knowledge-based Segmentation Method for Document Understanding," *Proceedings of the International Conference on Pattern Recognition,* pp. 745 - 748, 1986.

[81] F. Hones, E. G. Haffner, F. Fein, and A. Dengel. "A Hybrid Approach for Document Image Segmentation and Encoding," *Proc. First Int. Conf. on Document Analysis and Recognition,* Saint-Malo, France, Sept. 30 - Oct. 2, 1991, pp. 444-453.

[82] W. Horak, "Office Document Architecture and Office Document Interchange Formats: Current Status of International Standardization," *Computer,* Vol. 18, No. 10, Oct. 1985, pp. 50-60.

[83] M. Hueckel, "An Operator Which Locates Edges in Digital Pictures," *J. Assoc. Comput. Mach.,* vol. 18, pp. 113-125, 1971.

[84] J. J. Hull, "A Database for Handwritten Text Recognition Research," *IEEE Trans. Pattern Anal. Machine Intell.* **16,** 550-554 (1994).

[85] R. Hummel, "Feature Detection Using Basic Functions," *Computer Graphics Image Processing,* vol. 9, pp. 40-55, 1979.

[86] ICDAR, *Proc. First Int, Conf. on Document Analysis and Recognition*, Sain-Malo, (France), Sept. 30-Oct. 2, 1991.

[87] K. Inagaki, T. Hiroshima, and T. Sakai, "MACSYM: A Hierarchical Parallel Image Processing System for Event-driven Patterns Understanding of Documents," *Pattern Recognition,* vol. 17, no. 1, pp. 85-108, 1984.

[88] R. Ingold and D. Armangi, "A Top-down Document Analysis Method for Logical Structure Recognition," *Proceedings of the First International Conference on Document Analysis and Recognition*, pages 41-49, Saint-Malo, France, September 1991.

[89] ISO. 8613: Information Processing-Text and Office Systems-Office, Document Architecture (ODA) and Interchange Format, International Organization for Standardization, 1989.

[90] O. Iwaki, H. Kida, and H. Arakawa, "A Segmentation Method Based on Office Document Hierarchical Structure," *Proc. IEEE Int. Conf. Syst. Man. Cybern.,* Alexandria, VA, Oct. 1987, pp. 759-763.

[91] A. K. Jain, Fundamentals of Digital Image Processing, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 07632, USA, 1989.

[92] A. K. Jain and S. K. Bhattacharjee. "Address Block Location on Envelopes Using Gabor Filters: Supervised Method," *Proc. 11th Int. Conf. on Pattern Recognition,* 1992, pp. 264-266.

[93] A. K. Jain and S. K. Bhattacharjee. "Text Segmentation Using Gabor Filters for Automatic Document Processing," *Machine Vision and Applications,* Vol. 5, No. 3, pp. 169-184, 1992.

[94] A. K. Jain and F. Karrokhnia. "Unsupervised Texture Segmentation Using Gabor Filters," *Pattern Recognition*, Vol. 24, No. 12, pp. 1167-1186, 1991.

[95] E. G. Johnston. "Short Note: Printed Text Discrimination," *Computer Graphics and Image Processing,* Vol. 3, No. 1, pp. 83-89, 1974.

[96] V. Jolobof, Document Representation: Concepts and Standards. In J. Andre, R Furuta, and V. Quint, editors, *Structured Document,* pages 75-105. Cambridge University Press, 1989.

[97] S. Kahan, T. Pavlidis, and H. S. Baird, "On the Recognition of Printed Characters of any Font and Size," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 9, no. 2, pp. 274-788, 1987.

[98] J. N. Kapur, P. K. Sahoo, and A. K. Wong, "A New Method for Gray-level Picture Thresholding Using the Entropy of the Histogram," *Compt. Vision Graphics Image Process,* vol. 29, pp. 273-285, 1985.

[99] R. Kasturi et al., "A System for Interpretation of Line Drawings." *IEEE Trans. on Pattern Analysis and Machine Intelligence,* Vol. PAMI-12, No. 10, Oct. 1990.

[100] R. Kasturi and L. O'Gorman (guest eds.), "Special issue: Document Image Analysis Techniques," *Machine Vision and Applications*, Vol. 5, No. 3, pp. 141-248, 1992.

[101] R. Kasturi and L. O'Gorman (guest eds.), "Special Issue: Document Image Analysis Techniques," *Machine Vision and Applications*, Vol. 6, No. 2-3, pp. 67-180, 1993.

[102] D. P. M. Kelly and D. M. Abrahamson. "Document Structure Recognition," *Proc. First Int. Conf. on Document Analysis and Recognition*, Saint-Malo, France, Sept. 30 - Oct. 2, 1991, pp. 52-532.

[103] S. M. Kerpedjiev. "A Technique of Automatic data Extraction from Specialized Texts," SER-DICA, Bulg, math. public., 13, 1987, pp. 239-251.

[104] S. M. Kerpedjiev. "Automatic Extraction of Information Structures from Documents," *Proc. First Int. Conf. on Document Analysis and Recognition*. Saint-Malo, France, Sept. 30 - Oct. 2, 1991, pp. 32-40.

[105] S. M. Kerpedjiev, "Transformation of Weather Forecasts from Textual to Cartographie Form," *Computer Graphics Communications*, Vol. 61, 1990, pp. 246-256.

[106] H. Kida, O.Iwaki, and K. Kawada, "Document Recognition System for Office Automation," *Proc. 8th Int. Conf. on Pattern Recognition*, 1986, pp. 446-448.

[107] J. Kreich, A. Luhn, and G. Maderlechner, "An Experimental Environment for Model Based Document Analysis," *Proc. First Int. Conf. on Document Analysis and Recognition*, Saint-Malo, France, Sept. 30 - Oct. 2, 1991, pp. 50-58.

[108] J. Kreich, A. Luhn, and G. Maderlechner, "Knowledge Based Interpretation of Scanned Business Letters," *IAPR Workshop on CV*, 1988, pp. 417-420.

[109] S. W. Lam and S. N. Srihari, "Multi-domain Document Layout Understanding," *Proc. First Int. Conf. on Document Analysis and Recognition*, Saint-Malo, France, Sept. 30 Oct. 2, 1991, pp. 112-120.

[110] E. Lethelier, M. Lerour, and J. F. Couthouis, "An Automatic Reading System for Handwritten Numeral Amounts on French Checks," *Proc. of the Third International Conference on Document Analysis and Recognition (ICDAR'95)*, Montreal, Canada, 1995, pp. 92-97.

[111] I. Leplumey, J. Camillerapp, and C. Queguiner, "Kalman Filter Contributions Towards Document Segmentation," *Proc. of the 3rd International Conference on Document Analysis and Recognition*, Montrèal (Canada), Aug. 14-16, 1995, Vol. 3, pp. 765-769.

[112] M. D. Levine, Vision in Man and Machine, McGraw-Hill, New York, USA, 1989.

[113] K. Liu, C. Y. Suen, M. Cheriet, J. N. Said, C. Nadal, and Y. Y. Tang. "Automatic Extraction of Baselines and Data from Cheque Images", IEEE Trans. on Image Processing. (to appear)

[114] K. Liu, C. Y. Suen, and C. Nadal, "The Development of Cheque Image Segmentation Technique," Technical Report, CENPARMI, Concordia University, Montreal, Canada , Sept. 1995.

[115] K. Liu, C. Y. Suen and C. Nadal, "Automatic Extraction of Items for Payment Recognition," Technical Report, CENPARMI, Concordia University, Montreal, Canada, Dec. 1995.

[116] K. Liu, The Study of Image Processing Techniques for The Optical Testing of Flow Field, PhD Thesis, Nanjing University of Science and Technology, May 1989.

[117] H. Kato and S. Inokuchi, "A Recognition System for Printed Piano Music Using Musical Knowledge and Constraints," *Structured Document Image Analysis*, 1992, pp. 456-465.

[118] J. Kittler and J. Illingworth, "Minimum Error Thresholding," *Pattern Recognition*, vol. 19, pp. 41-47, 1986.

[119] F. Kimura and M. Shridhar "Segmentation-Recognition Algorithm for ZIP Code Field Recognition," *Machine Vision and Applications,* Vol. 5. No. 3, 1992, pp. 199-210.

[120] L. Kitchen and A. Rosenfield, "Edge Evaluation Using Local Edge Coherence," *IEEE Trans. Syst., Man. Cybern.* SMC-11, No. 9 (1981) pp. 597-605.

[121] R. Kohler, "A segmentation system based on thresholding," *Compt. Graphics Image Process,* vol. 15, pp. 319-338, 1981.

[122] R. Kooi and W. C. Lin, "An On-line Minicomputer-based System For Reading Printed Text Aloud," *IEEE Trans. Syst., Man. Cybern.,* vol. 8, no. 1, pp. 57-62, 1978.

[123] R. E. Korf, Optimal Path-finding Algorithm, In L. Kanal and V. Kumar, editors, *Search in Artificial Intelligence,* pages 223-267, Springer-Verlag, 1988.

[124] L. Lam, C. Y. Suen, D. Guillevic, N. W. Strathy, M. Cheriet, K. Liu, and J. N. Said, "Automatic Processing of Information on Cheques," *Proc. 1995 IEEE Int. Conf. on Systems, Man and Cybernetics*, Vancouver, Canada, Oct. 22-25, 1995, Vol. 3, pp. 2353-2358.

[125] D. S. Lee and S. N. Srihari, "Handprinted Digit Recognition: A Comparison of Algorithms," in *Proceedings of the* 3^{rd} *International Workshop on Frontiers in Handwritten Recognition*, Buffalo, May 1993, pp. 153-162.

[126] S. U. Lee, S. Y. Chung, and R. H. Park, "A Comparative Performance Study of Several Global Thresholding Techniques for Segmentation," *Comput. Vision Graphics Image Process.* Vol. 52 1990, pp. 171-190.

[127] M. Leroux, J. C. Salome and J. Badard, "Recognition of Cursive Script Words in a Small Lexicon," *Proc. of the First International Conference on Document Analysis and Recognition (ICDAR'91)*, Saint-Malo, France, 1991, pp. 774-782.

[128] *Machine Vision and Appl.*, Special Issue: Document Image Analysis Techniques, vol. 5, no., 3, 1992.

[129] J. V. Mahoney, "Automatic Analysis of Musical Score Image," B. S. thesis, Dept of Computer Science and Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02129, May 1982.

[130] J. Malik and P. Perona. "Preattentive Texture Discrimination with Early Vision Mechanisms," Journal Opt. Soc. Amer. A., Vol. 7, No. 5, pp. 923-932, 1990.

[131] J. C. Mao and A. K. Jain. "Texture Classification and Segmentation Using Multiresolution Simultaneous Autoregressive Models," *Pattern Recognition*, Vol. 25, No. 2, pp. 173 -188, 1992.

[132] D. Marr, "Early Processing of Visual Information," *Trans. Royal Society*, vol. B-275, pp. 483-524, 1976.

[133] I. Masuda, N. Hagita, and Akiyama, "Approach to Smart Document Reader System," *Proc. 1985 Comput. Vision Pattern Recognition*, (San Francisco. CA), 1985, pp. 550-557.

[134] H. Masuzaki, Takahashi, and Y. Kuroso, "HIFFILE 650E Optical Disk Filing System," *Hitachi Review*, Vol. 38, No. 5, pp. 257-264, 1989.

[135] O. Matan *et al.*, "Reading Handwritten Digits: A ZIP Code Recognition System," *Computer*, Vol. 25, No. 7, July 1992, pp. 59-63.

[136] G. Matheron, Random sets and Integral Geometry, Wiley, New York, 1975.

[137] T. Matsushina, "Automatic High Speed Recognition of Printed Music," *Proceedings. 1985 International Conference on Advanced Robotics*, pages 477-482. Japan Industrial /robot Association (JIRA), 3-5-8, Shiba Koen Minato-ku, Tokyo, 1985.

[138] L. Mero and T. Vamos, "Realtime Edge-detection Using Local Operators," *Proc. Third International Joint Conference on Pattern Recognition*, (Colorado CA), November 1976, pp. 31-36

[139] M. Minoh, T. Munetsugu, and K. Ikeda, "Extraction and Classification of Graphical Symbol Candidates Based on Perceptual Organization," *Proc. 11th Int. Cour. on Pattern Recognition*, 1992, Vol. B, pp. 234-237.

[140] M. Minsky, A Framework for Representing Knowledge, in P. Winston, editor, The Phychology of Computer Vision, McGraw-Hill, 1975.

[141] J. Modestino and Fries, "Edge Detection in Noisy Images Using Recursive Digital Filtering," *Computer Graphics Image Processing*, vol. 6, pp. 404-433, 1977.

[142] J. V. Moreau, B. Plessis, O. Bougeois and J. L. Plagnaud, "A Postal Check Reading System," *Proc. of the First International Conference on Document Analysis and Recognition (ICDAR'95)*, Saint-Malo, France, pp. 756-766.

[143] G. Nagy. "A Preliminary Investigation of Techniques for the Automated Reading of Unformatted Text," ACM, Vol. 11, No. 7, pp. 480-487, 1968.

[144] G. Nagy. "What Does a Machine Need to know to Read a Document?," *Proc. Symposium on Document Analylsis and Information Retrieval,* March 16-18, 1992, pp. 1-10.

[145] G. Nagy, "Towards a Structural Document Image Utility," *Structured Document Image Analysis.* H. S. Baird, H. Bunke, and K. Yamamoto, eds., Springer Verlag, Berlin, 1992, pp. 54-69.

[146] G. Nagy, S. Seth, and M. Viswanatha, "A Prototype Document Image Analysis System for Technical Journals," *IEEE Computer,* 25(7):10-22, 1992.

[147] G. Nagy, Kanai, and M. Krishnamoorthy, "Complementary Techniques For Digital Document Analysis," *Proc. ACM Conf. on Document Processing Systems,* 1988, pp. 169-176.

[148] G. Nagy and S. Seth, "Hierarchical Representation of Optically Scanned Documents," *Proc. 7th Int'l Conf. Pattern Recognition (ICPR),* IEEE CS Press, Los Alamitos, Calf., 1984, pp. 347-349.

[149] N. Nahi, "Role of Recursive Estimation in Statistical Image Enhancement," *Proc. IEEE,* vol 60, pp. 872-877, July 1972.

[150] N. Nahi and Lopez-Mors, "Estimation-detection of Object Boundaries in Noisy Images," *IEEE Trans. Automatic Contr,* vol. AC-23, pp. 834-846, 1978.

[151] N. Nahi and M. H. Jahanshahi, "Image Boundary Estimation," *IEEE Trans. Computers,* vol. C-26, pp. 772-781, August 1977.

[152] N. Nahi and T. Assefi, "Bayesian Recursive Image Estimation," *IEEE Trans. Computers,* vol. C-21, pp. 734-738, July 1972.

[153] Y. Nakano, H. Fujisawa, and O. Kunisaki, "A Document Understanding System Incorporating Character Recognition," *Proc. 6th Int. Conference Pattern Recognition* (Paris, France) 1986, pp. 801-803.

[154] Y. Nakano, H. Fujisawa, O. Kunisaki, K. Okada, and T. Hananoi, "A Document Understanding System Incorporating with Character Recognition," *Proc.8th ICPR*, Paris, pp. 801-803, 1986.

[155] Y. Nakano and H. Fujisawa, "A Method of Document Understanding for Automatic Filing," *Trans. Japanese Inst. of Electron. and Comm. Eng.,* Vol. J71-D, No. 10, pp. 2050-2058, 1988.

[156] D. Niyogi and S. N. Srihari, "A Ruled-based System for Document Understanding," *Proc. AAAI'86,* 1986, pp. 789-793.

[157] L. O'Gorman and M. B. Clowes, "Finding Picture Edges Through Collinearity of Feature Points," *IEEE Trans. Computers* vol. C-25, pp. 449-456, April 1976.

[158] L. O'Gorman, "The Document Spectrum for Structural Page Layout Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence,* Vol. 15, No. 11, 1993, pp. 1162-1173.

[159] L. O'Gorman and R. Kasturi, Document Image Analysis, IEEE Computer Society Press, Los Alamitos, California, 1995, Chapter 3.

[160] M. Okamoto and A. Miyazawa, "An Experimental Implementation of Document Recognition Systems for Paper Containing Mathematical Expressions," *Pre-Proc. 1990 Syntactic and Structural Pattern Recognition* (Murray Hill, NJ), 1990, pp. 335-351.

[161] N. Otsu, "A Threshold Selection Method from Gray-level Histograms," IEEE Trans. Systems Man Cybernet. SMC-8, 1978, 62-66.

[162] N. R. Pal and S. Pal, "A Review on Image Segmentation Techniques," *Pattern Recognition*, Vol. 26, No. 9, pp. 1277-1294, 1993.

[163] P. W. Palumbo, "Postal Address Block Location in Real Time," *Computer*, Vol. 25, No. 7, July 1992, pp. 34-42.

[164] D.C.W. Pao, H. F. Li and R. Jayakumar, "Shape Recognition Using the Straight Line Hough Transform: Theory and Generation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* 14, No. 11 (1992) pp. 1076-1089.

[165] A. Perry and D. G. Lowe, "Segmentation of Textured Images," *Proc. IEEE Computer Vision and Pattern Recognition,* San Diego, CA, 1989, pp. 326-332.

[166] I. Pitas and C. Kotropoulos, "A texture-based Approach to the Segmentation of Sesmic Image," *Pattern Recognition,* Vol. 25, No. 9, pp. 929-945, 1992.

[167] S. L. Horowitz and T. Pavlidis, "Picture Segmentation by a Tree Traversal Algorithm," *J. Assoc. Comput. Mach.,* vol. 23, pp. 368-388, Apr. 1976.

[168] G. B. Porter and E. V. Rainero, Document Reconstruction: A System for Recovering Document Structure From Layout, In C. Vanoirbeek and G. Coray, editors, *Proceedings of Electronic Publishing, 1992*, pages 127-141. Cambridge University Press, 1992.

[169] J. Pecht, "Speeding-up Successive Minkowski Operations with Bitplane Computers," *Pattern Recognition Letters*, vol. 3, pp. 113-117, 1985.

[170] I. Pitas, Digital Image Processing Algorithms, Prentice Hall International (UK) Ltd, 1993, Chapter 5, pp. 223-253.

[171] *Proc. First Int'l Conf. Document Analysis and Recognition*, Saint-Malo, France, 1991.

[172] *Proc. Second Int'l Conf. Document Analysis and Recognition*, Tsukuba Science City, Japan, 1993.

[173] *Proc. Third Int'l Conf. Document Analysis and Recognition*, Montreal, Canada, 1995.

[174] V. Quint, Systems for the manipulation of structured documents. In J. Andre, R. Futura, and V. Quint, editors, *Structured Documents*, pages 39-74. Cambridge University Press, 1989.

[175] S. Reddi, S. F. Rudin, and H. R. Keshavan, "An Optimal Multiple Threshold Scheme for Image Segmentation," *IEEE Trans. Systems Man Cybernet.* vol. 14 (4), pp. 661-665, 1984.

[176] S. V. Rice, J. Kanai, and T. A. Nertker, "An Evaluation of OCR Accuracy," *UNLV Information Science Research Inst. 1993 Report*, University of Nevada, Las Vegas, Nov., 1993.

[177] J. W. Roach and J. E. Tatem, "Using Domain Knowledge in Low-level Visual Processing to Interpret Handwritten Music: An Experiment," *Pattern Recognition*, 21(1):33-44, 1988.

[178] L. G. Roberts, "Machine Perception of Three-dimensional Solids," *Optical and Electrooptical Information Processing*, (J.T. Tippett, *et al., Eds.*), pp. 159-197, MIT Press, Cambridge, Mass., 1965.

[179] A. Rosenfeld and A. Kak, "Digital Picture Processing," Academic Press, New York, 1976.

[180] J. N. Said, K. Khorasani, and C. Y. Suen, "A New Back-Propagation Learning Algorithm with Application to Unconstrained Handwritten Character Recognition," *World Congress on Neural Networks WCNN'95*, Washington DC July 17-21, 1995, vol. 2, pp. 217-221.

[181] J. N. Said, M. Cheriet, and C. Y. Suen, "An Extended Formal Module for Visual Data Processing of Business Forms and Bank Cheques," *Vision Interface'96*, Toronto, Ontario, May 21-24 1996 pp. 84-92.

[182] J. N. Said, M. Cheriet, and C. Y. Suen, "Dynamic Morphological Processing: A Fast Method For Base Line Extraction," *Proc. 13th International Conference on Pattern Recognition, ICPR'96*, Aug. 25-30, 1996 Vienna, Austria. Vol. B, pp. 8-12.

[183] P. K. Sahoo, S. Soltani, and A. K. C. Wong, "SURVEY: A Survey of Thresholding Techniques," *Compt. Vision Graphics Images Process*, vol. 41, pp. 233-260, 1988.

[184] J. C. Salome, M. Leroux and H. Oiry, "Retrieval of Script Information Appearing in Bank Cheques for Automatic Purposes," *Proc. 5th Visual Communications and Image Processing'90*, Switzerland, Oct 1990.

[185] G. Semi and E. Cohen, "External Word Segmentation of Off-line Handwritten Text Lines," *Pattern Recog.*, **27**, 41-52 (1994).

[186] J. Serra, "Image Analysis and Mathematical Morphology," Academic Press, London, 1982.

[187] K. Shanmugam, F. Dickey, and R. Dubes, "An Optical Frequency Domain Filter for Edge Detection in Digital Pictures," *IEEE, Trans. Pattern Anal. Machine Intelligence*, vol. PAMI-1, pp. 37-49, 1979.

[188] C. E. Shannon, "A Mathematical Theory of Communication," Bell Syst. Tech. J., Vol. 27, pp. 379-423, 1948.

[189] C. E. Shannon, "A Mathematical Theory of Communication," Bell Syst. Tech. J., Vol. 27, pp. 623-656, 1948.

[190] H. C. Shen and C. Y. C. Bie. "Feature Frequency Matrices as Texture Image Representation," *Pattern Recognition Letters*, Vol. 13, No. 3, pp. 195-205, 1992.

[191] H. C. Shen and C. Y. C. Bie, "Visual Textural Properties Based on Feature Frequency Matrices," *Proc. Vision Interface VI'91*, 1991, pp. 170-175.

[192] J. C. Simon and K. Zerhouni. "Robust description of a line image," *Proc. First Int. Conf. on Document Analysis and Recognition*, Saint-Malo, France, Sept. 30 - Oct. 2, 1991, pp. 3-14.

[193] M. Spann and R. Wilson, "A Quad-tree Approach to Image Segmentation Which Combines Statistical and Spatial Information," *Pattern Recognition*, vol. 18, pp. 257-269, 1985.

[194] A. L. Spitz. "Style Directed Document Recognition," *Proc. First Int. Conf. on Document Analysis and Recognition*, Saint-Malo, France, Sept. 30 - Oct. 2, 1991, pp. 611-619.

[195] S. N. Srihari and G. W. Zack, "Document Image Analysis," *Proc. 8th Int. Conf. Pattern Recognition*, (Paris, France), 1986. pp. 434 - 436.

[196] G. Stoy et al., "The RightPages Image-Based Electronic Library for Alerting and Browsing," *Computer*, Vol. 25, No. 9, Sept. 1992, pp. 17-26.

[197] N. W. Strathy, F. Said, and C. Y. Suen, "A Sequential Method of Extracting Contour Chains from an Image," *in Proceedings of the* 12^{th} *International Conference on Pattern Recognition*, Jerusalem, Israel, October 1994, pp. 580-582.

[198] N. W. Strathy, C. Y. Suen, and A. Krzyzak, "Segmentation of Handwritten Digits Using Contour Features," *in Proceedings of the* 2^{nd} *International Conference on Document Analysis and Recognition*, Ibaraki, Japan, October 1993, pp. 577-580.

[199] C. Y. Suen, L. Lam, D. Guillevic, N. W. Strathy, M. Cheriet, J. N. Said, and R. Fan, "Bank Cheque Processing," *International Journal of Imaging Systems and Technology, Special Issue on Document Analysis and Recognition*. Vol. 7, 1996. pp. 392-403.

[200] C. Y. Suen, "The Research and Development of the Automatic Payment Recognition System," Technical Report, CENPARMI, Concordia University, Montreal, Canada, December, 1995.

[201] C. Y. Suen and Y. Y. Tang. "A Statistical Approach to Document Classification," Technical Report, Centre for Pattern Recognition and Machine Intelligence (CENPARMI), Concordia University, 1992.

[202] C. Y. Suen and Y. Y. Tang. "A Structural Approach to Document Classification," Technical Report, Centre for Pattern Recognition and Machine Intelligence (CENPARMI), Concordia University, 1992.

[203] T. N. Tan and A. G. Constantinities. "Texture Analysis Based on a Human Visual Model," *Proc. IEEE Int. Conf. on Acoust., Speech, Signal Proc.*, Albuquerque, New Mexico, 1990, pp. 2091-2110.

[204] Y. Y. Tang and C. Y. Suen. "Concrete Document and Conceptual Document," Technical Report, Centre for Pattern Recognition and Machine Intelligence (CENPARMI), Concordia University, 1992.

[205] Y. Y. Tang, C. Y. Suen, and C. D. Yan. "Document Processing for Automatic Knowledge Acquisition," to appear in *IEEE Trans. on Knowledge and Data Engineeriny, 1994.*

[206] Y. Y. Tang, C.D. Yan, M. Cheriet, C.Y. Suen, Automatic Analysis and Understanding of Documents. C. H. Chen, L.F. Pau, and P.S.P. Wang, editors, *Handbook of Pattern Recognition and Computer Vision,* World Scientific. Singapore, 1993.

[207] Y. Y. Tang, C. Y. Suen, C. D. Yan, and M. Cheriet, "Document Analysis and Understanding: A Brief Survey," *Proceedings of the International Conference on Document Analysis and Recognition,* pages 17-31, 1991.

[208] Y. Y. Tang, C. Y. Suen, C. D. Yan, M. Cheriet, "Financial Document Processing Based on Staff Line and Description Language," *IEEE Trans. Systems, Man. and Cybernetics,* vol. 25, no. 5, pp. 738-754, May 1995.

[209] Y. Y. Tang, C. D. Yan, and C. Y. Suen, "Form Description Language and its Mapping onto Form Structure," *Technical Report,* Centre for Pattern Recognition and Machine Intelligence (CENPARMI), Concordia University, 1990.

[210] Y. Y. Tang, C. D. Yan, M Cheriet, and C. Y. Suen, "Financial Document Analysis and Understanding," Technical Report, Centre for Pattern Recognition and Machine Intelligence (CENPARMI), Concordia University, 1990.

[211] Y. Y. Tang, C. Y. Suen, and C. D. Yan, "Chinese Form Pre-processing for Automatic Data Entry," *Proc. Int. Conf. on Computer Processing of Chinese and Oriental Languages,* August 13-16, 1991, Taipei, Taiwan.

[212] S. L. Taylor, R. Fritzson, and J. A. Pastor, "Extraction of Data from Preprinted Forms," *Machine Vision and Applications* Vol. 5, No. 3, 1992, pp. 211-222.

[213] J. T. Tou, "Computer Recognition of Design Concepts," *Proc. 11th Int. Conf. on Pattern Recognition,* 1992, Vol. B, pp. 639-642.

[214] J. Toyoda, Y. Noguchi, and Y. Nishimura, "Study of Extracting Japanese Newspaper Articles," *Proc. 6th Int. Conf. Pattern Recognition* (Munich, Germany), 1982, pp. 1113-1115.

[215] O. D. Trier and A. K. Jain, "Goal-directed Evaluation of Binarization Methods," *IEEE Trans. Pattern Analysis and Machine Intelligence,* Vol. 17 (12), pp. 1191-1201, December 1995.

[216] O. D. Trier and T. Taxt, "Evaluation of Binarization Methods for Document Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* Vol. 17 (3), pp. 312-315, March 1995.

[217] Y. Tsuji, "Document Image Analysis for Generation Syntactic Structure Description," *Proceedings of the International Conference on Pattern Recognition,* pages 744-747, 1988.

[218] Y. Tsuji *et al.*, "Document Recognition System With Layout Structure Generator," *Proc. IAPR Workshop on Machine Vision Applications,* (Tokyo, Japan), 1990, pp.479-482.

[219] S. Tsujimoto and H. Asada, "Understanding Multi-Articled Documents," *Proceedings 10th Int. Conf. Pattern Recognition* (Atlantic City, NJ), 1990, pp. 551-556.

[220] S. Tsujimoto and H. Asada, "Resolving Ambiguity in Segmenting Touching Characters," *Proceedings 1st Inter. Conf. Document Analysis and Recognition* (Saint Malo, France), 1991, pp. 701-709.

[221] S. Tsujimoto and H. Asada, "Document Image Processing for Accurate and High Speed Text Reading," in *Pre-Proc. 1990 Syntactic and Structural Pattern Recognition* (Murray Hill, NJ), 1990, p. 501.

[222] M. R. Turner. "Texture Discrimination by Gabor Functions," *Biol Cybern.*, Vol. 55, pp. 71-82, 1986.

[223] *Proc. 4th United States Postal Service Advanced Technology Conf.*, Washington, D.C., 1990.

[224] S. Venkatesh and L. J. Kitchen, "Edge Evaluation Using Necessary Components," *CVGIP-Graphic Models Image Processing* 54 (1992) pp. 23-30.

[225] H. Yashiro, T. Murakami, Y. Nakano, and H. Fujisawa, "A New Method of Document Structure Extraction Using Generic Layout Knowledge," *Proc. Int'l Workshop Industrial Applications of Machine Intelligence and Vision*, MIV-89, Tokyo, pp. 282-287, April 1989.

[226] F. M. Walh, K. Y. Wong, and R. G. Casey, "Block Segmentation and Text Extraction in Mixed Text/Image Documents," *Comput. Graphics, and Image Processing,* vol. 20, pp. 375-390.

[227] C. H. Wang, P. W. Palumbo, and N. Srihari, "Object Recognition in Visually Complex Documents: A Method for Locating Address Blocks on Mail Pieces," *Proc. 9th ICPR, pp. 365-367,* 1986.

[228] S. Wang and R. M. Haralick, "Automatic Multithreshold Selection," *Compt. Vision Graphics Image Process,* vol. 25, pp. 46-67, 1984.

[229] D. Wang and S. N. Srihari, "Analysis of Form Images," *International Journal of Pattern Recognition and Artificial Intelligence,* Vol. 8 No. 5, pp. 1031-1052, 1994.

[230] T. Watanabe and Q. Luo, "A framework of Layout Recognition for Document Understanding," *Proc. Symposium on Document Analysis and Information Retrieval,* March 16-18, 1992, pp. 77-95.

[231] T. Watanabe, H. Naruse, Q. Luo, and N. Sugie. "Structure Analysis of Table-form Documents on the Basis of the Recognition of Vertical and Horizontal Line Segments," *Froc. First Int. Conf. on Document Analysis and Recognition,* Saint-Malo, France, Sept. 30 - Oct. 2, 1991, pp. 638-646.

[232] J. Weszka, C. Dyer, and A. Rosenfeld, "A Comparative Study of Texture Features for Terrain Classification," *IEEE Trans. System Man Cybernet,* vol. SMC-6, pp. 269-285, 1976.

[233] L. Wilcox and Spitz, Automatic Recognition and Representation of Documents. In J. C. van Vliet, editor, Document Manipulation and Typography, pages 47-57. Cambridge University Press, 1988.

[234] Wilkinson *et al.*, "The First Census Optical Character Recognition Systems Conference," Tech Report NIST 4912, US, Bureau of the Census and Nat'l Inst. of Standards, Washington, D.C., 1992.

[235] K. Y. Wong, R. G. Casey, and F M. Wahl. "Document Analysis Systems." *IBM Journal Res. Development,* 26(6):647-656, 1982.

[236] Y. Yakimovsky, "Boundary and Object Detection in Real World Images," *J. Assoc. Comput. Mach,* vol. 23, pp. 599-618, Oct. 1976.

[237] M. Yamada and K. Hasuike, "Document Image Processing Based on Enhanced Border Following Algorithm," *Proc. 10th Int. Conf. on Pattern Recognition.* 1990, pp. 551-556.

[238] A. Yamashita, T. Amano, H. Takahashi, and K. Toyokawa. "A Model Based Layout Understanding Method for Document Recognition System (DRS)." *Proc. First Int. Conf. on Document Analysis and Recognition,* Saint-Malo, France, Sept. 30 - Oct. 2, 1991, pp. 130-138.

[239] J. Y. Yang and K. Liu, "A New Method of Extraction Continuous Curves from Images," *Proceedings of IEEE Asian Electronics Technology Conference,* Hong Kong, 1987, pp. 244-248.

[240] C. D. Yan, Y. Y. Tang, and C. Y. Suen, "Form Understanding System Based on Form Description Language," *Proc. First Int. Conf. on Document Analysis and Recognition,* (Saint-Malo), France, Sept. 30-Oct 2, 1991, pp. 283-293.

[241] P. S. Yeh, S. Antony, A. Litcher, and Rosenfeld, "Address Location on Envelopes," *Pattern Recognition,* Vol. 20, No. 2, pp. 213-227.

[242] X. Zhuang and R. M. Haralick, "Morphological Structuring Element Decomposition," *Computer Vision Graphics Image Processing,* vol. 35, pp. 370-382, 1986.