Document processing for adaptive page segmentation using order statistic filters

Li Ma

Department of Computer Science
McGill University, Montreal

March 2001

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Science

0-612-70460-2

Abstract

Page segmentation is one of the important and basic research subjects of document analysis. Traditionally, there are two major kinds of page segmentation approaches. One is the top-down approach and the other is the bottom-up approach. Though these two approaches are been used till now, they are not effective for processing documents with high geometrical complexity and the process of splitting document needs iterative operations which is time consuming. The Modified Fractal Signature (MFS) approach which was presented in recent years can overcome the above weaknesses, however it needs to calculate modified fractal signature which makes the theory and the algorithm very complex. In this thesis, we present two new page segmentation approaches (one is the Maximum Order Statistic Filter (MaxOSF) approach, the other is the Median Order Statistic Filter (MexOSF) approach) based on the order statistic filter (OSF) which is more direct and much simpler. We use the MedOSF to remove the salt-pepper noise of the document and use the MaxOSF to do the page segmentation. In practice, they not only can adaptively process the documents with high geometrical complexity, but also save a lot of computing time.

Thesis Supervisor: M. Newborn

Title: Professor

Résumé

La segmentation de page est l'un des importants sujets de recherche en analyse de documents. Traditionnellement, deux approches diffrentes sont utilises dans la segmentation de page dont l'un est descendante et l'autre ascendante. Toutefois, ces deux approches n'offrent aucune efficacites dans l'analyse de documents d'une grande complexit gometrique et l'aspect iteratif du processus de division du document consumme trop de temps. Le "Modified Fractal Signature (MFS)" est une approche introduite, dans ces dernires annes, dans le but de rsoudre ce problme. Le dfaut de cette approche est la complexit du calcul du "modified fractal signature" ce qui rend la thorie et l'algorithme plus compliqus. Dans cette thse, nous allons presenter deux nouvelles approches de segmentation de page: le "Maximum Order Statistic Filter (MaxOSF)" et le "Median Order Statistic Filter (MedOSF)" bases sur "l'Order Statistic Filter (OSF)" dont l'utilisation est plus directe et simple. On utilise le MedOSF pour enlever le bruit du document et le MaxOSF pour la segmentation de page. En pratique, ils peuvent non seulement adaptivement procder des documents d'une grande complexit geomtrique, mais aussi reduire considrablement le temps de calcul.

Superviseur de theses: M. Newborn

Titre: Professeur

Acknowledgements

I would like to express my deepest gratitude to my supervisor Professor M. Newborn. Without his continuous encouragement, invaluable discussions and unselfish assistance, this thesis could not be presented as it is.

I would like to thank Professor T. Merrit and Professor X. Chang, for their kind encouragement and support. I would like to acknowledge to Professor L. Devroye and Professor C. Crepeau for their wonderful lectures and support through the years.

Many thanks go to the other professors, staff, and fellow students, who helped me in any way in my work. I am especially grateful to Professor H. Ma and Professor J. Zhou for discussions on the aspects of the work and providing me several publications related to this thesis.

I am also very grateful to the Department of Computer Science at McGill for providing me with an excellent environment for my studies and thesis work.

Very special thanks go to Z. Zhang, H. Zhou and H. Pan for giving me their warmest friendship during my studies. Their support encouraged me to pass through difficult times.

Finally, I want to express my deepest appreciation to my family, my parents and my dear brother for their encouragement, patience, understanding and sacrifices they made during my studies. Their infinite love and support have always been a constant source of my enthusiasm of study.

Contents

List of Figures

Chapter 1

Introduction

1.1 Data and knowledge engineering

1.1.1 Introduction

According to Webster's dictionary [1], [2], "data refer to numerical information suitable for computer processing, while *knowledge* refers to the sums or range of what has been perceived, discovered or learned. Knowledge can be considered *data* at a high level of abstraction and can be processed by a computer when it is represented as data". The distinction between these two concepts in terms of computer processing is usually vague [3], [4]. In general, we can consider knowledge as a compact and sometimes imprecise way of representing a body of data.

The subjects covered in the field of *data and knowledge engineering* are collectively refering to the algorithms, methods and systems for the design, utilization and maintenance of data and knowledge [2]. The following four areas are involved: design, modeling, control, access and evaluation of data and knowledge engineering systems; methods for automated acquisition and learning of new data and knowledge; representation, architectural and language supports; and deployment, evolution, maintenance and standardization of data and knowledge engineering systems with existing and emerging technologies. We can summarize them in one sentence that data and knowledge engineering can be considered as studies related to computer-aided management of information. As new applications arise and emerging technologies become more mature, the problems involved in this area are continuously evolving and the scope covered by these subjects is flexible and will change in time. In this thesis, we shall focus on algorithms for document processing and page segmentation.

Different degrees of abstraction of a given problem in data and knowledge engineering may be required, depending on the complexity of the problem involved[2]. Simpler problems required less abstraction and can therefore be implemented easily in hardware/software. An example of this type of problem is the design of a hardware selection unit for a relational database. More complex problems require a hierarchy of abstraction to simpler problems before they can be solved. These simpler problems by themselves may not represent realistic, efficient or realizable implementations. An example is a distributed database that can handle concurrent accesses and up-dates. The concurrency control problem is extremely complex if all characteristics of the physical distributed system and user behavior have to be considered. To solve this problem, a simplified model of the communication network, processors and user behavior has to be assumed. A concurrency control algorithm is then developed for the simplified mode and the algorithm is mapped onto the physical distributed system. Since simplifying assumptions have been made in developing the algorithm, the algorithm actually implemented on the physical system may not be totally efficient.

A conceptual view of problems in knowledge and data engineering and the relationship to applications and technologies with varying degrees of abstraction is illustrated in Figure 1 [2].

1.1.2 History of data and knowledge engineering

First let us briefly review the history of this field. The early years of computing research were dominated by information processing [2]. "Loosely speaking, information refers to bits that are stored in computer memories. These bits include knowledge represented in data and software. In Von Neumann computers, for example, data and programs are stored in the same memory area. Therefore, data can sometimes be interpreted as programs, while programs are sometimes considered as data. Since knowledge can be treated as a general class of data, its characterization may be uncertain or imprecise. Knowledge can be represented in computers as data or software, or as a mixture of both."

As applications grew more and more complex, it was discovered that techniques for processing data were quite different from techniques for designing and processing software, and that techniques for acquiring and managing knowledge were different from techniques for data processing and information processing. This discovery led to the development of structure programming, database processing and artificial intelligence [5], [6], [7] in the 1960's and 1970's. Early data and knowledge processing systems were small in scale: modeling, design and management could be handled by one or a few experts. The collective effort of a large number of experts is required with the ever increasing complexity of applications and information processing systems in the 1980's and the increasing need to capture and manage abstract knowledge. Data and knowledge can no longer be handled by individuals alone, and its management must be treated as an engineering discipline.

Due to recent advances in technology and computer architecture, data and knowledge engineering systems become feasible. Research on multiprocessing and distributed processing allow faster and parallel computers to be utilized efficiently. Existing technologies, such as VLSI, VHSIC, and fiber optics and emerging technologies, such as three-dimensional VLSI, lightwave technologies, sea of gates and superconductivity, promise faster computers, more memory, and higher networking bandwidth.

Studies on knowledge and data engineering, therefore, span a broad spectrum of areas and encompass data, information, knowledge, technology, and products. The subject area is truly dynamic and its emphasis may change as new applications evolve and new technologies become mature [2].

1.2 Document processing and page segmentation

1.2.1 Introduction

Document processing and page segmentation, as one of the major fields of knowledge and data engineering, is the main goal of this thesis. The document processing bottleneck has become the major impediment to the development and application of effective information systems [8]. In order to remove this bottleneck, new document processing techniques must be introduced to automatically acquire knowledge from various types of documents.

Generally speaking, a document is considered to have two structures: *geometric structure* and *logical structure*. These two structures play a key role in the process of knowledge acquisition. The latter is divided into phases which involves: *document analysis* (extracting the geometric structure from a document) and *document understanding* (mapping the geometric structure into logical structure). The basic concept of document structure (geometric structure and logical structure) will be introduced and also two traditional approaches: the top-down and the bottom-up approaches will be briefly discussed in later sections.

1.2.2 Page segmentation

Page segmentation is one of the important and basic research subjects of document analysis. The traditional approaches (the top-down and the bottom-up approaches) are used up to now and they are not effective for processing documents with high geometrical complexity and the process of splitting document need iterative operations which is time consuming. Although there are several new approaches such as the MFS approach [9] which need to calculate modified fractal signature, no matter the simplicity of the theory and the efficiency of the algorithm, it needs a further improvement. In this thesis, we shall present two new approaches which are more direct and simpler in Chapter 3. In practice, they not only can adaptively process documents with high geometrical complexity, but also save a lot of computing time.

1.3 Outline of the thesis

In Chapter 1, we introduced the engineering background of this thesis and also presented the reason why and how document processing and page segmentation was brought to this research field.

By presenting a survey on the techniques and problems involved, Chapter 2 aims at serving as a catalyst to simulate research in automatic knowledge acquisition through document processing. In this chapter, we provide a brief overview of the current research and development directions in automatic knowledge acquisition and document processing. We classify research problems and approaches in this area, and discuss future trends.

In Chapter 3 we provide a new approach for page segmentation which is based on the order statistic filter (OSF). We shall introduce the statistic background related to this approach and give the details of the proof. The method was tested by simulations in Chapter 4. The thesis concludes with a discussion of the directions for further work which is presented in Chapter 5.

Chapter 2

Automatic knowledge acquisition and document processing

2.1 Introduction

As mentioned in the issue of knowledge data engineering in Chapter 1, one of the key technologies related to data and knowledge engineering is the acquisition of data and knowledge in the development and utilization of information systems [10]. One of the most challenging topics is document processing and page segmentation.

In fact, many attempts based on artificial intelligence have been made to settle this problem in various ways [11], [12], [13], [14]. We can see that in order to provide a promising solution to this problem, software engineering and artificial intelligence will have to join hands [15]. Their approaches can be divided into two categories: *manual techniques* and *computer-based techniques* [12].

The manual techniques are further divided into many categories such as interviewing methods [16], [17], [18], [19], [20], [21], psychology-based methods [22]-[32], active knowledge engineer roles methods [33], [34], [35],[36] etc., which are listed in the following. Similarly, computer-based techniques can be further divided into ***learning-based*** and ***interactive*** techniques such as multiple experts methods [37], [38], apprenticeship learning methods [39], etc. [40]-[54].

- manual
  - interviewing
    - structured interview [16], [17]
    - unstructured interview [18], [19]
    - focused interview [20], [21]
  - psychology – based
    - protocol analysis [22], [26]
    - psychological scaling [27], [30]
    - card sorting [31], [32]
  - active knowledge engineering roles
    - tutorial interview [33]
    - participant observation [35]

- computer – based
  - interactive
    - multiple experts [37]
    - psychology – based and interviewing methods [38]
  - learning – based
    - apprenticeship learning [42]
    - similarity – based learning [47]

In practice, there is a lot of knowledge that must be acquired from documents including technical reports, letters, newspapers, books, journals, magazines, government files and bank checks etc. and the acquisition of knowledge from such documents can involve an extensive amount of handcrafting on the part of the engineer which is time consuming and limits the application of the information systems. Thus, automatic knowledge acquisition from documents has become an important subject. Using a new technique, *document analysis* and *understanding*, which belongs to a branch of artificial intelligence, is a hopeful solution to this problem and it makes sense to acquire knowledge directly by analyzing and understanding documents. Typical document analysis and understanding system is briefly outlined in the following.

- document analysis
  - geometric comlexity
    - documents without graphics
    - forms
    - documents with graphics
    - documents with graphics as the main elements
  - two traditional approaches
    - top – down approach
    - buttom – up approach
  - techniques
    - hough transform
    - run – length smoothing algorithm
    - neighbourhood line density

- document understanding
  - logical model
  - tree transformation
  - formatting knowledge
  - form definition language

From the survey of the history of document processing we can see that a lot of research on document processing has been made based on optical character recognition (OCR) [20], [21] since 1960's. About two decades ago, the study of automatic text segmentation and discrimination started. Automatic text segmentation and discrimination, or document analysis and understanding, have been widely studied since the early 1980's [22], [23], [24], [25], with advance development of modern computers and the increasing need to acquire large volumes of data. Up to today, a lot of methods have been proposed, and many document processing systems have been described. More details can be found in [26], [27], [28], [29], [30], [31], [32], [33].

Basically, document processing involves extraction of the geometric structure of a document and mapping it into a logical structure from which knowledge can be acquired. Practically, it requires both analysis and understanding of general documents such as books, newspapers, magazines, letters etc. [34], [35], [36] which refer to locating the areas of title, abstract, text, graphic, and half-tone images on the document, finding their relations and finally identifying them, and special documents such as electronic circuit diagrams, music notations, envelopes, checks, tax return forms etc. [37], [38], [39], [40], [41] which refer to separation of characters, texts and symbols from graphics and recognizing them.

According to different requirements and applications, a physical document may be a color image, a binary image or a monochrome grey scale image. Hence, the analysis and understanding of color images [42], binary images or monochrome grey scale images [43] is required. However, in this thesis, we only concern with binary document images, that is because in most cases, the document is black/white and can be represented by a binary image. Also, color images and grey scale images can be converted into binary images, and the analysis and understanding of binary images can be mapped back to their original forms.

First of all, let us consider the terminologies used in this field. Since different terminologies have been used in different studies [44], [45], [46], [47], [50], [55], to establish unified definitions of these terminologies is our first task. We propose that document processing be 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 document refers to document analysis; mapping the geometric structure into logical structure is defined as document understanding. However, in some cases, there is no clear boundary between the two phases just described. For example, we can find the document logical structure by knowledge rules during an analysis of envelops or bank checks. Once we capture the logical structure, we can acquire knowledge from the document. A block diagram of this method is shown in Figure A.2 [8]. The relationship among the geometric structure, logical structure, document analysis, and document understanding can also be found in this figure.

In order to generate knowledge from a document, it is very important to detect its structure. The goal of this thesis is to capture the geometric structure of a document through page segmentation, which is the first phase of document analysis, see Figure A.2. As for the logical structure of a document we leave it for furture work. In the following, we shall present the brief concepts of geometric structure and logical structure.

2.2 Geometric structure of the document

According to [8], we define the geometric or layout structure as follows:

Definition 1 *Geometric (layout) structure is the result of dividing and subdividing the* *content of a document into increasingly smaller parts, on the basis of the presentation.*

An element of the specific geometric structure is called *geometric (layout) object*. We define the following types of geometric objects, according to [8].

- *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 containing 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 frames or one or more blocks.
- *Page set* is a set of one or more pages.

In Figure A.3 (a), a document is composed of several articles, each of which consists of a title, an abstract, subtitles, and paragraphs. The title dominates abstract, chapters, and sections, in which subtitles, dominate paragraphs. Figure A.3 (b) is an example of a geometric structure that is represented by a tree generated from a document shown in Figure A.3 (a). The root node in this example stands for a page of document. Each child node of the tree stands for a set of adjacent blocks located in the same column.

2.3 Logical structure of the document

According to [8], we can define the logical structure as follows:

Definition 2 *Logical structure is the result of dividing and subdividing the content of a document into increasingly smaller parts on the basis of the human-perceptible meaning of the content—for example, into chapters, sections, subsections and paragraphs.*

An element of the specific logical structure of a document which is called *logical object*. For logical object, there are only three types that are defined which are *basic logical object*, *composite logical object*, and *document logical root*. Logical object categories such as *chapter*, *section* and *paragraph* are application-dependent and can be defined using the *object class* mechanism [8].

An example of logical structure that can be represented by a tree form is shown in Figure A.3 (c) corresponding to Figure A.3 (a).

2.4 Relations between geometric structure and logical structure

The geometric structure and the logical structure provide alternative but complementary views of the same document. For example, a block can be regarded as consisting of chapters containing figures and paragraphs or, alternatively, as consisting of pages that contain text blocks and/or graphic blocks. We can consider relations between geometric structure and logical structure in the following manner.

In general, the geometric structure and the logical structure are independent of each other. The geometric structure of a document is determined by a formatting process. The logical structure is usually determined by the author and is embedded in the document during the process of editing. There are attributes called *geometric directives* associated with the logical structure which control the formatting process of geometric structure. For example, the requirement that a chapter starts on a new page, the title of a section and the first two lines of its first paragraph are presented on the same page. Geometric directives may be collected into layout styles, each of which may be referred to by one or more logical objects. There may exist correspondence between geometric objects and logical objects. In principle, however, there is no one-to-one correspondence, since a logical structure corresponds to a variety of geometric structures. There is a transformation of a geometric structure into a logical structure, but the reverse dose not always exist.

Chapter 3

Robustness of statistics and the OSF page segmentation approach

3.1 Introduction

In spite of the use of electronic documents, the volume of paper-based documents continues to grow at a rapid rate. Although paper-based documents which have existed for several centuries, because of the human preference for reading and archiving paper documents, it will continue to play an important role in our lives. However, to store and retrieve the ever increasing number of paper documents has become more and more difficult and cumbersome. On the other hand, electronic documents have several advantages in storage, retrieval and updating. As a result, transformation of a paper document to its electronic version and subsequent document image understanding have become an important and challenging application domain for artificial intelligence and pattern recognition researchers.

The field of document image understanding covers a variety of documents such as technical articles, business letters and faxes, forms, postal mail pieces, musical scores, and maps and drawings. A geometric page layout of a document is a specification of the geometry of the maximal homogeneous regions and their classification (text, table, image drawing, etc). Logical page layout analysis involves determining the type of page, assigning functional labels (title, logo, footnote, caption, etc.) to each block of the page, determining the relationships of these blocks, and ordering the text blocks according to their reading order [56]. *Page segmentation*, as a bottleneck of document understanding, has become the major impediment to the development and application of effective information systems. In order to remove this bottleneck, new techniques must be introduced to page segmentation of various types of documents

From [9] we know there are two traditional approaches, the top-down approach and the bottom-up approach, which can extract the geometric structure from a document refers to document analysis [2]. However these two approaches have two major disadvantages. First, the top-down approach needs iterative operations to break the document into several blocks, while the bottom-up approach needs to merge small components into larger ones iteratively, which means both approaches are time consuming. Second, they are not effective for processing documents with high geometrical complexity. Especially, the top-down approach can process only simple documents with specific format or contain some a priori information.

The paper [9] presented an approach based on the *modified fractal signature* (MFS) for document analysis. The modified fractal signature approach considers the document as a binary image with a fractal feature, using 2-D Minkowski dimension theory to compute the modified fractal signature of document binary image, then according to the modified fractal signature to process page segmentation. It does not need iterative breaking or merging, however, it needs to compute the modified fractal signature, that is why the algorithm and theory are so complex. In the following sections we shall introduce two new page segmentation approaches based on the *order statistic filter* (OSF). For these approaches we only deal with the statistical feature of a document image instead of the fractal feature which are much simpler and effiective. In practice, the OSF approaches not only can adaptively process the document with high geometrical complexity, but also dramatically decrease the algorithm complexity. Notice that here the adaptiveness means that we apply the OSF approaches to not only a regular document image but also a irregular document image.

3.1.1 The document areas and the distributions

First let us see a page of document D (here we only consider D as a black and white document), which consists of many regions that might be texts, tables, graphics and margin blank areas. In general, we can classify those areas into two categories: *text area* (including tables, graphics, texts, and the blank areas between the words, lines and paragraphs) and *blank area* (the four-margin blank areas). See Figure 3.1. In this thesis, we use 0's present the white pixels and 1's present the black pixels.

$$D = \{D_{\mathrm{T}}, D_{\mathrm{B}}\} \tag{3.1}$$

where D_{T} represents a text area, D_{B} denotes a blank area.

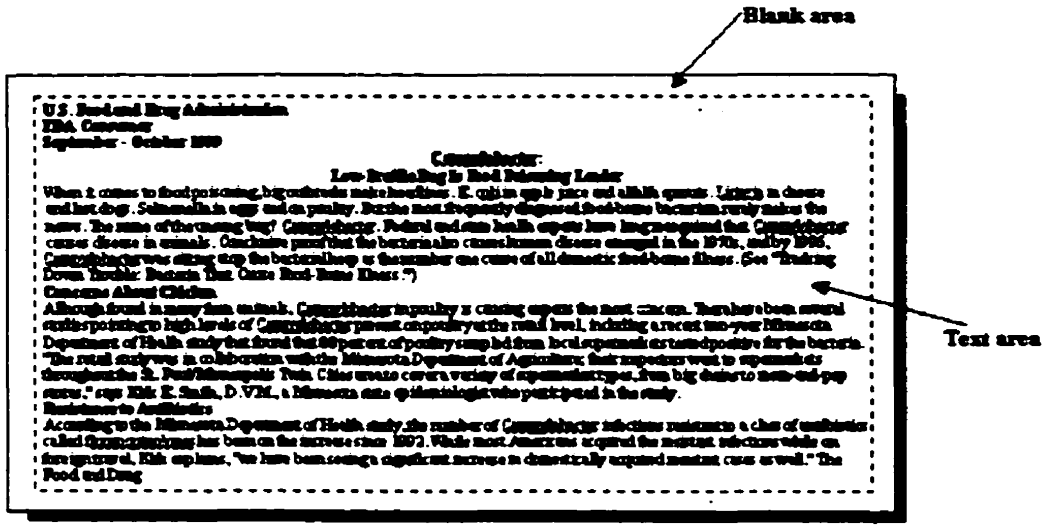


Figure 3.1: Two areas of a document image.

distribution.

x	P(X=x)
0	1

From the probability theory we know the distribution function of a random variable X is $F(x) = P(X < x)$, hence the distribution function of the single point distribution is

$$F(x) = U(x) = \begin{cases} 0 & x < 0, \\ 1 & x \geq 0. \end{cases}$$

where $U(x)$ is called a standard 0-1 distribution function. See Figure 3.2.

Similarly, for an arbitrary pixel $x_{i,j} \in D_{\mathrm{T}}$, it is either black or white, i.e. either $x_{i,j} = 1$ or $x_{i,j} = 0$ and the probability $x_{i,j} = 1$ ($x_{i,j} = 0$) is $\frac{1}{2}$. We can say for any pixel in D_{T}, the distribution of the pixel is an equal probability 0-1 distribution.

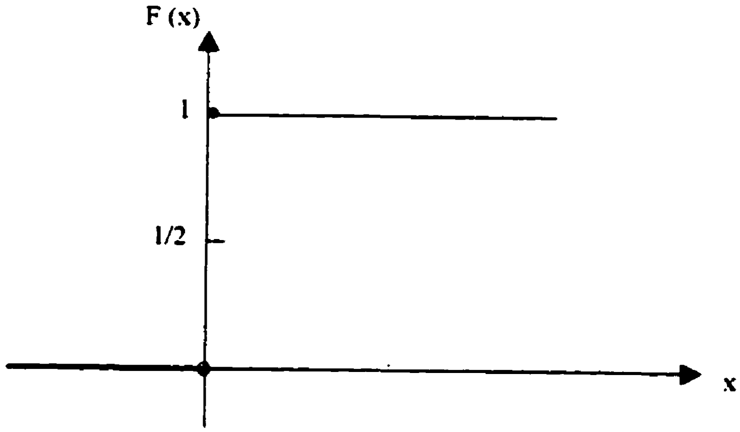


Figure 3.2: A illustration of the single point distribution.

x	P(X=x)
0	$\frac{1}{2}$
1	$\frac{1}{2}$

Hence the distribution function of the equal probability 0-1 distribution is

$$F(x) = \frac{1}{2}(U(x) + U(x-1)) = \begin{cases} 0 & x < 0, \\ \frac{1}{2} & 0 \leq x < 1, \\ 1 & x \geq 1, \end{cases}$$

where

$$U(x-1) = \begin{cases} 0 & x < 1, \\ 1 & x \geq 1. \end{cases}$$

See Figure 3.3.

In this thesis, we shall present two approaches which use the maximum order statistic filter (MaxOSF) and the median order statistic filter (MedOSF) to process the page segmentation, see Section 3.2.3.

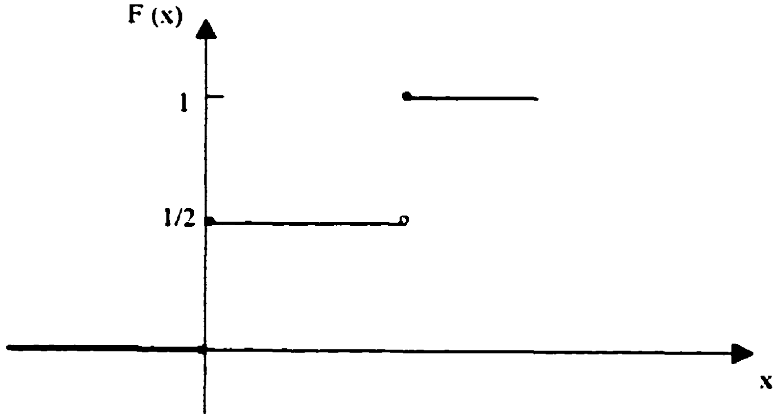


Figure 3.3: A illustration of the equal probability 0-1 distribution.

It is natural to ask how to determine whether a statistic filter is robust, or how to evaluate the sensitivity a statistic filter. In Section 3.1.2, we shall introduce the concept of influence function which can be used to evaluate the sensitivity of a statistic filter. We apply these theories to the MaxOSF and the MedOSF respectively in Section 3.2. Here before we start to discuss the robustness of a statistic filter, we need to introduce some fundamental concepts — functional, order statistics and influence function.

3.1.2 Robustness of statistic

In the statistic parameter estimation theory, a *functional* usually can be considered as a *function* with a domain of a distribution distance space. The difference between functional and function is that the domain of a function is a set of numbers, while the domain of a functional is a set of functions (distributions).

$$\text{function} \qquad F: \; R \longrightarrow R, \;\; F(x) = y, \;\; x \in R, \;\; y \in R,$$

where R is a real number set.

$$\text{functional} \qquad T: \ m \longrightarrow R, \ T(F) = y, \ F \in m, \ y \in R,$$

where m is a set of distributions (functions).

The concept of influence function was first presented by Hample [] and here we use it to evaluate the sensitivity of a statistic filter.

Definition 3 *Let $T(\cdot)$ be a functional which is defined on a set of distributions , for any $x \in R$, $F_\varepsilon(t) = (1-\varepsilon)F(t) + \varepsilon U(t-x), (0 \leq \varepsilon \leq 1)$. If the following limit exists*

$$\begin{aligned} IC(x,F,T) &= \lim_{\varepsilon \to 0} \frac{1}{\varepsilon}\{T(F_\varepsilon) - T(F)\} \\ &= \frac{d}{d\varepsilon} T(F_\varepsilon)|_{\varepsilon=0}, \end{aligned} \tag{3.2}$$

then $IC(x,F,T)$ is said to be an influence function of functional $T(\cdot)$ about overall distribution F.

Notice that for influence function $IC(x,F,T)$, x is the outlier and it can be any real number. However, in this thesis we only discuss the binary document image, hence the value of the outliers are 1.

3.2 Robustness of two important order statistics

In the fields of image processing and pattern recognition, the non-linear stochastic filters which based on order statistics have many very important applications. The theories of these applications are based on the robustness of order statistics. In the following sections, we shall discuss the robustness of two important order statistics — maximum

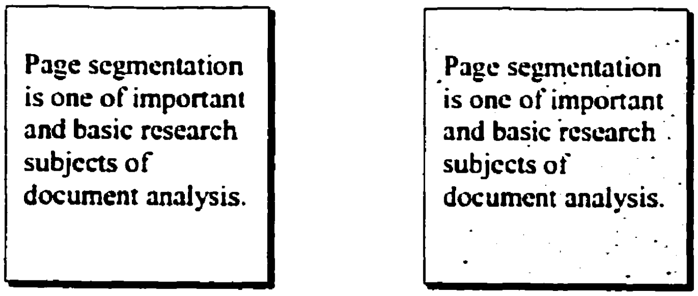


Figure 3.4: A original document image and an image corrupted by noise.

Left: A clear document image; Right: A document image corrupted slightly by sale-pepper-noise.

order statistic (sample maximum) and median order statistic (sample median).

First of all, we shall give the definitions of sample maximum and sample median.

Definition 4 *The order statistics of a sample* $(X_1, \cdots, X_n)$ *are the sample values placed in ascending order. They are denoted by* $(X_{(1)}, \cdots, X_{(n)})$ *which satisfy*

$$X_{(1)} \leq X_{(2)} \leq \cdots \leq X_{(n)}.$$

$X_{(n)}$ *is the sample maximum of* $(X_{(1)}, \cdots, X_{(n)})$ *and we denote it as*

$$X_{(n)} = \max(X_1, \cdots, X_n).$$

The sample median, which we denote by X_{med}, *is a number such that one-half of the observations are less than* X_{med} *and one-half are greater. In terms of the order statistics,*

X_{med} is defined by

$$X_{\text{med}} = \begin{cases} X_{(\frac{n+1}{2})} & \textit{if } n \textit{ is odd}, \\ (X_{(\frac{n}{2})} + X_{(\frac{n}{2}+1)})/2 & \textit{if } n \textit{ is even}. \end{cases}$$

3.2.1 Robustness of the sample median

Definition 5 *Let $T_{\text{med}}(\cdot)$ a functional which is defined on a set of distributions, if*

$$T_{\text{med}}(F) = \inf\{x : x \in R, \text{ s.t. } F(x) \geq \frac{1}{2}\}, \tag{3.3}$$

then T_{med} is said to be a median functional.

For example, according to this definition, when F is an equal probability 0-1 distribution, we have

$$T_{\text{med}}(F) = \inf\{x : x \in R, \text{ s.t. } F(x) = \frac{1}{2}\} = \inf\{x : x \in [0,1)\} = 0. \tag{3.4}$$

At the beginning of this chapter we know that usually a document image is a binary image, when we process the page segmentation using order statistic filter, we always deal with two kinds of distributions: the two-point distribution (document image area) and the single point distribution (four-margin blank area). We will see the effects of the median functional about these two kinds of distributions are different.

Example 1 Influence function of the single point distribution.

We know the distribution function of the single point distribution is

$$F(x) = U(x) = \begin{cases} 0 & x < 0, \\ 1 & x \geq 0. \end{cases} \tag{3.5}$$

Here $F_\varepsilon(t) = (1-\varepsilon)F(t) + \varepsilon U(t-x)$. When $x = 1$, the graph of $F_\varepsilon(t)$ is like Figure 3.5

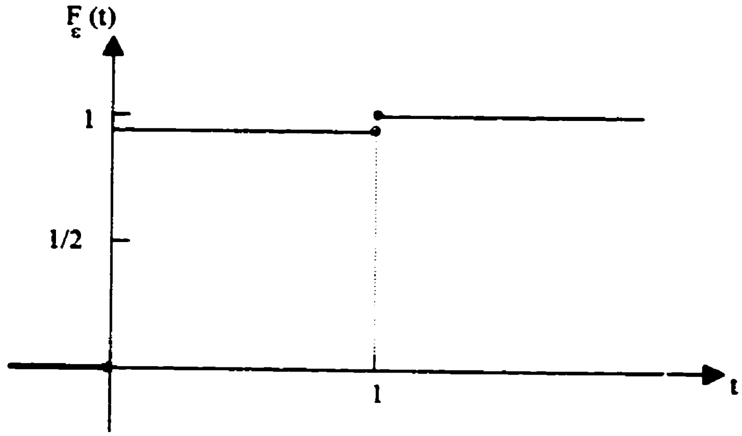


Figure 3.5: Distribution function of the single point distribution.

Suppose $0 < \varepsilon < \frac{1}{2}$, then we have

$$T_{\text{med}}(F) = 0, \qquad T_{\text{med}}(F_\varepsilon) = 0,$$

hence,

$$IC(1, F, T_{\text{med}}) = \lim_{\varepsilon \to 0} \frac{1}{\varepsilon}(T_{\text{med}}(F_\varepsilon) - T_{\text{med}}(F)) = 0, \tag{3.6}$$

which means the sample median is robust to the outliers on the blank area.

Example 2 Influence function of the equal probability 0-1 two-point distribution.

We know the distribution function of the equal probability 0-1 distribution is

$$F(x) = \frac{1}{2}[U(x) + U(x-1)]. \tag{3.7}$$

Here $F_\varepsilon(t) = (1-\varepsilon)F(t) + \varepsilon U(t-x)$. When $x = 1$,

$$F_\varepsilon(t) = \frac{1-\varepsilon}{2}U(t) + \frac{1+\varepsilon}{2}U(t-1).$$

Suppose $0 < \varepsilon < \frac{1}{2}$, then the graph of $F_\varepsilon(t)$ is like the following Figure 3.6

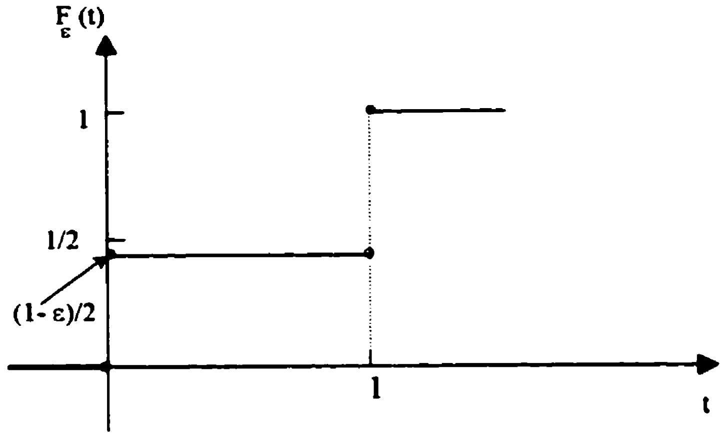


Figure 3.6: Distribution function of the 0-1 distribution.

Here, we have

$$T_{\text{med}}(F) = 0, \qquad T_{\text{med}}(F_\varepsilon) = 1,$$

hence,

$$IC(1, F, T_{\text{med}}) = \lim_{\varepsilon \to 0} \frac{1}{\varepsilon}(T_{\text{med}}(F_\varepsilon) - T_{\text{med}}(F)) = \lim_{\varepsilon \to 0} \frac{1}{\varepsilon} = \infty, \tag{3.8}$$

which means the sample median is very sensitive to the outliers on the text area.

In summary, if F is a single point distribution, sample median is not sensitive to the outlier $x = 1$, however, if F is a equal probability 0-1 distribution, sample median is very sensitive to the outlier $x = 1$, see Table 3.2.1.

Areas	Robustness	Distribution	$\mathrm{IC}(1, \mathrm{F}, \mathrm{T}_{\text{med}})$
text area	not robust	equal probability 0-1 distribution	∞
blank area	robust	single point distribution	0

Table 3.1: Robustness of the sample median.

3.2.2 Robustness of the sample maximum

Definition 6 *Let $T_{\max}(\cdot)$ be a functional which is defined on a set of distributions, if*

$$T_{\max}(F) = \sup\{x : x \in (\text{Supp } p(x)) \cup (\text{Supp } p_i)\}, \tag{3.9}$$

where $p(x)$ is the density function of continuous random variable, $\{p_i\}$ is the distribution law of discrete random variable, Supp $p(x)$ *and* Supp p_i *are support sets of $p(x)$ and p_i respectively, then $T_{\max}$ is said to be a maximum functional.*

Notice that the support set of $p(x)$ (or p_i) is a set of t, such that $p(x) \neq 0$ (or $p_i \neq 0$).

Theorem 1 *Let $T_{\max}(\cdot)$ be the maximum functional which is defined on a set of distributions of one dimension bounded continuous random variables and discrete random variables, then the influence functional of the maximum functional has the following analytic expressions:*

$$IC(x, F, T_{\max}) = \infty, \qquad \text{if } x > \sup\{x :\ x \in (\text{Supp } f(x)) \cup (\text{Supp } p_i)\},$$

$$IC(x, F, T_{\max}) = 0, \qquad \text{if } x \leq \sup\{x :\ x \in (\text{Supp } f(x)) \cup (\text{Supp } p_i)\}. \tag{3.10}$$

Proof. For an arbitrary F, from Lebesgue decomposition theorem we have

$$F(x) = a_1 F_c(x) + a_2 F_d(x), \tag{3.11}$$

where $F_c(x)$, $F_d(x)$ denote continuous distribution and discrete distribution respectively. $F_c(x)$ has a density function $f(x)$ and $F_d(x)$ has a distribution law $\{p_i\}$.

Clearly

$$T_{\max}(F) = \sup\{x :\ x \in (\text{Supp } f(x)) \cup (\text{Supp } p_i)\},$$

$$T_{\max}(F_\varepsilon) = \sup\{x : x \in (\text{Supp } (1-\varepsilon)f(x) \cup (\text{Supp } (1-\varepsilon)p_i) \cup \{x\}\}$$

$$= \sup\{x : x \in (\text{Supp } f(x)) \cup (\text{Supp } p_i) \cup \{x\}\}.$$

We have

(a) if $x > \sup\{x : \ x \in (\text{Supp } f(x)) \cup (\text{Supp } p_i)\}$, then $T_{\max}(F_\varepsilon) = x$, hence

$$T_{\max}(F_\varepsilon) - T_{\max}(F) > 0,$$

therefore

$$\begin{aligned} IC(x, F, T_{\max}) &= \lim_{\varepsilon \to 0} \{T_{\max}(F_\varepsilon) - T_{\max}(F)\}/\varepsilon \\ &= \infty. \end{aligned}$$

(b) if $x \leq \sup\{x : \ x \in (\text{Supp } f(x)) \cup (\text{Supp } p_i)\}$, then $T_{\max}(F_\varepsilon) = T_{\max}(F)$, hence

$$T_{\max}(F_\varepsilon) - T_{\max}(F) = 0,$$

therefore

$$\begin{aligned} IC(x, F, T_{\max}) &= \lim_{\varepsilon \to 0} \{T_{\max}(F_\varepsilon) - T_{\max}(F)\}/\varepsilon \\ &= 0. \end{aligned}$$

■

From Theorem 1 we can conclude that if $x > T_{\max}(F)$, even there are only small number of outliers, statistic $X_{(n)}$ will be greatly affected; but if $x \leq T_{\max}(F)$, $X_{(n)}$ is very robust.

Example 1 Influence function of the single point distribution.

We know the single point distribution is

x	$p_i = P(X = x)$
0	1

The support set Supp p_i (a set of x such that $p_i \neq 0$) is $\{0\}$, therefore

$$\sup\{x : x \in \text{Supp } p_i\} = 0,$$

for the outlier $x = 1$, we always have $x > \sup\{t : t \in \text{Supp } p_i\}$, therefore from Theorem 1 we have $IC(1, F, T_{\max}) = \infty$ which means sample maximum is very sensitive to the outliers on the blank area.

Example 2 Influence function of the equal probability 0-1 two-point distribution.

We know the equal probability 0-1 distribution is

x	$p_i = P(X = x)$
0	$\frac{1}{2}$
1	$\frac{1}{2}$

The support set Supp p_i (a set of x such that $p_i \neq 0$) is $\{0,\ 1\}$, therefore

$$\sup\{x :\ x \in \text{Supp } p_i\} = 1,$$

for the outlier $x = 1$, we always have $x \leq \sup\{x : x \in \text{Supp } p_i\}$, therefore from Theorem 1 we have $IC(1, F, T_{\max}) = 0$ which means sample maximum is robust for the outliers on the text area.

In summary, if F is a single point distribution (the blank area), sample maximum is sensitive to the outlier $x = 1$, however, if F is a equal probability 0-1 distribution (the text area), sample median is robust for the outlier $x = 1$, see Table 3.2.2.

Areas	Robustness	Distribution	$\mathrm{IC}(1,\mathrm{F},\mathrm{T}_{\max})$
text area	robust	equal probability 0-1 distribution	0
blank area	not robust	single point distribution	∞

Table 3.2: Robustness of sample maximum.

3.2.3 The OSF page segmentation approaches

From previous sections we know a page of document D consists of text area D_T and blank area D_B

$$D = \{D_{\mathrm{T}}, D_{\mathrm{B}}\}. \tag{3.12}$$

For an arbitrary pixel $X_{i,j} \in D$, we choose its connected (say, four-connected) neighborhood,

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

Figure A.4 (a) Four-connected neighborhood

Let $(x_{i-1,j},\ x_{i,j-1},\ x_{i,j},\ x_{i,j+1},\ x_{i+1,j})$ be observations of sample

$$(X_{i-1,j},\ X_{i,j-1},\ X_{i,j},\ X_{i,j+1},\ X_{i+1,j}).$$

For document binary image, if $X_{i,j}$ is black pixel, then $x_{i,j} = 1$; if $X_{i,j}$ is white pixel, then $x_{i,j} = 0$. Hence we can construct OSFs with filter window length (threshold) n=5,

$$x_{i,j} = \mathrm{MaxOSF}(x_{i-1,j},\cdots,x_{i,j+1}) = \max(x_{i-1,j},\cdots,x_{i,j+1}) = x^{ij}_{(n)} = x^{ij}_{(5)}, \tag{3.13}$$

$$x_{i,j} = \text{MedOSF}(x_{i-1,j}, \cdots, x_{i,j+1}) = \text{med}(x_{i-1,j}, \cdots, x_{i,j+1}) = x^{ij}_{\text{med}} = x^{ij}_{(3)}, \qquad (3.14)$$

then we can easily process the page segmentation of document binary image.

Chapter 4

Validation of the OSF approaches

4.1 Introduction

When we apply the OSF approaches, the OSF has the effect of linking together neighborhood black areas that are separated by less than the threshold n (i.e. the filter window length, the number of the connected neighborhood +1). With an appropriate choice of n, the linked areas will be regions of a common color. The degree of linkage depends on the following factors: a) the threshold value n, b) the distribution of white and black pixels in the document, and c) the scanning resolution.

If the smoothing thresholds are selected correctly, the blocks of different content in the document images will be smeared into images with different features. The choice of the smoothing threshold n is very important. Very small n value simply close individual characters. Slightly larger values of n smooth together individual characters in a word but are not large enough to bridge the space between two words. Too large values of n often cause sentences to join to non-text regions, or to connect to adjacent columns.

In general, threshold n is set according to the character height, gap between words and interline spacing.

4.2 Introduction to the programs and the implementations

We implement these approaches using Matlab under Windows environment, and we also use Matlab to plot the graphs. The following are the two approaches.

4.2.1 The MaxOSF approach

We use the MaxOSF approach, which is based on the maximum order statistic filter, to process the page segmentation. The details of the algorithm are presented in the following:

Algorithm

Input: a page of document image.

Output: the geometric structure of the document.

Step 1 Scan a document image D_0 and get the binary image file (BMP file).

Step 2 Read the file bit by bit, then we can get a binary matrix M_0. The position of each matrix component stands for the location of the pixel in the image. 0's represent the white pixels and 1's represent the black pixels.

Step 3 Use the OSF to process the binary matrix M_0, we can get another matrix M_1 which includs the information of the geometric structure of the document:

Substep 1 Let $x_{i,j}$ be a component of M_0 and $x_{i-1,j}$, $x_{i,j-1}$, $x_{i,j+1}$, $x_{i+1,j}$ be the four-connected neighborhood of $x_{i,j}$. Sort $x_{i-1,j}$, $x_{i,j-1}$, $x_{i,j}$, $x_{i,j+1}$, $x_{i+1,j}$ in ascending order.

Substep 2 Let $x'_{i,j}$ be a component of M_1, then

$$x'_{i,j} = \max\{x_{i-1,j},\ x_{i,j-1},\ x_{i,j},\ x_{i,j+1},\ x_{i+1,j}\} = x^{ij}_{(5)}.$$

Step 4 From matrix M_1, we can get the image D_1 which represents the geometric structure of the document.

For Step 3, let us see an example.

Suppose we have a binary matrix M_0 (7×10 matrix),

$$M_0 = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

We can expand it to a 9×12 matrix M'_0 with components $y_{i,j}$, $i = 1, \cdots, 9$, $j = 1, \cdots, 12$, such that

$$y_{i,j} = \begin{cases} x_{i-1,j-1} & i = 2, \cdots, 8;\ j = 2, \cdots, 11, \\ 0 & i = 1, 9;\ j = 1, 12, \end{cases}$$

i.e.

$$M_0' = \begin{pmatrix} 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&1&1&1&0&0&0&0&0&0&0 \\ 0&1&1&1&1&0&0&0&0&0&0&0 \\ 0&1&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&1&0&0&0 \\ 0&0&0&0&0&0&0&1&1&1&1&0 \\ 0&0&0&0&0&0&0&1&1&1&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \end{pmatrix}$$

Using the MaxOSF, we can get another 9×12 matrix M_1'.

$$M_1' = \begin{pmatrix} 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&1&1&1&1&0&0&0&0&0&0 \\ 0&1&1&1&1&1&0&0&0&0&0&0 \\ 0&1&1&1&1&0&0&0&0&0&0&0 \\ 0&1&1&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&1&1&1&1&1&1&0 \\ 0&0&0&0&0&0&1&1&1&1&1&0 \\ 0&0&0&0&0&0&1&1&1&1&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \end{pmatrix}$$

The components of matrix M_1', $y'_{i,j}$, $i = 1, \cdots, 9$, $j = 1, \cdots, 12$, satisfy that,

$$y'_{i,j} = \begin{cases} x_{(5)}^{i-1\ j-1} & i = 2, \cdots, 8;\ j = 2, \cdots, 11, \\ 0 & i = 1, 9;\ j = 1, 12. \end{cases}$$

Hence, we can get a matrix M_1 which is

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

4.2.2 The MedOSF approach

Sometimes the document image is corruped by salt-pepper noise, we need to remove the noise first before we do the page segmentation. Here we use the MedOSF to remove the noise.

Algorithm

Input: a page of document image with salt-pepper-noise.

Output: a page of document without the noise.

Step 1 Scan a document image D_0 and get the binary image file (BMP file).

Step 2 Read the file bit by bit, then we can get a binary matrix M_0. The position of each matrix component stands for the location of the pixel in the image. 0's represent white pixels and 1's represent black pixels.

Step 3 Use OSF to process the binary matrix M_0, we can get another matrix M_1 which includs the information of the geometric structure of the document:

Substep 1 Let $x_{i,j}$ be a component of M_0 and $x_{i-1,j}$, $x_{i,j-1}$, $x_{i,j+1}$, $x_{i+1,j}$ be the four-connected neighborhood of $x_{i,j}$. Sort $x_{i-1,j}$, $x_{i,j-1}$, $x_{i,j}$, $x_{i,j+1}$, $x_{i+1,j}$ in ascending order,

Substep 2 Let $x'_{i,j}$ be a component of M_1, then

$$x'_{i,j} = \text{median}\{x_{i-1,j},\ x_{i,j-1},\ x_{i,j},\ x_{i,j+1},\ x_{i+1,j}\} = x^{ij}_{(3)},$$

Step 4 From matrix M_1, we can get an image D_1 which represents the document without the noise.

For Step 3, let us see an example.

Suppose we have a binary matrix M_0 (7×10 matrix),

$$M_0 = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

After We adding some noise to the orignal image, the binary image M_0 becomes

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

We can expand it to a 9×12 matrix M_0' with components $y_{i,j},\ i = 1, \cdots, 9,\ j = 1, \cdots, 12,$ such that

$$y_{i,j} = \begin{cases} x_{i-1,j-1} & i = 2, \cdots, 8;\ j = 2, \cdots, 11, \\ 0 & i = 1, 9;\ j = 1, 12, \end{cases}$$

i.e.

$$M_0' = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Using the MedOSF, we can get another 9×12 matrix.

$$M'_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The components of matrix M'_1, $y'_{i,j}$, $i = 1, \cdots, 9$, $j = 1, \cdots, 12$, satisfy that,

$$y'_{i,j} = \begin{cases} x^{i-1\ j-1}_{(5)} & i = 2, \cdots, 8;\ j = 2, \cdots, 11, \\ 0 & i = 1, 9;\ j = 1, 12. \end{cases}$$

Hence, we can get the matrix M_1 which is

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

4.3 Comparision

From the previous section we know the efficiency of the algorithms are decided by: a) the size of the document image ($M \times N$ matrix), b) the threshold n. The computing time of algorithm is $O(nMN)$. In practice, once the threshold is decided, we consider n as a constant (say 5, 9, etc.), therefore we can conclude that the computing time is $O(MN)$ (if the input matrix is a $N \times N$ square matix, we say the computing time is $O(N^2)$). Comparing to the top-down and the bottom-up methods, the OSF approaches do not iterative operations and we just need to go through each element of the input image matrix one by one, then we get the segmentation immediately. That is why the OSF appraches are so fast and effective.

We made three simulations to test the effects of the MaxOSF and the MedOSF approaches. In Simulation 1 and 2, we applied the MaxOSF approach to two different document images with two different geometrical complexities respectively, see Figure A.5, A.8. The geometric structure of the first document image is not complicated where the document blocks have regular shapes and the geometric structure of the second document image is compicated where the document blocks have irregular shapes and also it includes a picuture. The results are shown in Figure A.6, A.9. In Simulation 3, we applied the MedOSF approach to a document which is corrupted by salt-pepper-noise, see Figure A.11. Here we first use the MedOSF to remove the noise, then we use the MaxOSF to do page segmentation, the result is shown in Figure A.12.

4.3.1 Effect of the MaxOSF approach

We applied the MaxOSF approach to two document images (Simulation 1 and 2) and for each image, we change the threshold n from 5, 9, 13 to 25. In Figure A.6, A.9 we see that when the threshold n is small, it can only smooth together individual characters in a word but are not large enough to bridge the space between the words. Too large values of n often cause sentences to join to nontext regions, or to connect to adjacent columns (or adjacent paragraphs). In Simulation 1 and 2, we see that when $n = 25$, the results are best, if $n > 25$ then it causes sentences to connect to adjacent paragraphs.

Also we see that when $n = 5,\ 13$, the edges of segmentations are zigzag, but when $n = 9,\ 25$, the edges of segmentations are smooth. This is because we used two different shapes of filter windows. From Figure A.4, we see that when $n = 5,\ 13$, the filter windows are rhombuses which cause the zigzags of the edges. However, when $n = 9,\ 25$, the filter windows are squares, that is why the edges of segmentations are smooth.

4.3.2 Effect of the MedOSF approach

In Simulation 3, we applied the MedOSF to a document image which is corrupted by noise. We first use the MedOSF to remove the noise (see Section 4.2.2), then we use the MaxOSF to process the segmentations (see Section 4.2.1). If the original document is not corrupted very seriously, then MedOSF is very effective. However, from Chapter 3 we know that sample median is robust to outliers (noise) on the blank areas, but sensitive to the outliers on the text areas, which means if the text areas of the original document are corrupted seriously by the noise, then the effeciency of the MedOSF in not ideal. The result is shown in Figure A.12.

Chapter 5

Conclusions and future work

5.1 Conclusions

In this thesis, we presented two approaches (one is the MaxOSF approach, another is the MedOSF approach) to process page segmentation. We applied these two approaches to three documents and for each document we change the threshold from $n = 5, 9, 13$ to 25. All the results were generated by using Matlab. Extensive use of Matlab was a feature of this thesis. Results of the simulations were crucial to the conclusion of the thesis. The details about the Matlab code can be found in the Appendix.

As described in Chapter 3, we know that a binary document image generally consists of text areas and blank areas. The MaxOSF is robust for the text area, we can use it to process the page segmentation and the MedOSF is robust for the blank area, hence we use it to remove the noise. From the experimental results we notice that when choosing a different threshold will result in a different segmentation. If n is small, it can not bridge the space between the words (between the sentences), if n is too big, it causes sentences

OSF	Areas	Robustness	Influence function
MaxOSF	Text area	robust	$IC(1, F, T_{\max}) = 0$
MaxOSF	Blank area	not robust	$IC(1, F, T_{\max}) = \infty$
MedOSF	Text area	not robust	$IC(1, F, T_{\text{med}}) = \infty$
MedOSF	Blank area	robust	$IC(1, F, T_{\text{med}}) = 0$

Table 5.1: Robustness of the MaxOSF and the MedOSF.

(graphs) to connect to the adjacent paragraph. Usually, for different document images, the thresholds are different. This is decided by the distribution of white and black pixels in the document and the scanning resolution. From our experiments we see that when $n = 25$, the segmentations are best. The results are shown in Figure A.7, A.10, A.12.

5.2 Further work

Experimental results show that the new page segmentation approaches: the MaxOSF and the MedOSF are very effective. Like the MFS approach in [9], the OSF approaches can adaptively process the page segmentation to all kinds of document binary image with high geometrical complexity, and the approaches are very simple and no need of iteration. However, the OSF approach has its own limitation – it is not applicable to the general grey level images or the images with background color (such as the paper money). The page segmentation of those images is very difficult. Somehow, we know that most of the document images such as books, newspapers, magazines, paper etc. are binary images, therefore the OSF approaches are good enough. However, there are several ways we may extend the approaches to deal with grey level images, such as setting a window filter to remove the background color, or providing a new algorithm which combines the MaxOSF approach with the MFS approach etc.. A rigorous theoretical development needs to be done and this will be left for future work.

Bibliography

(1) Substantial references which are sorted by the order they are mentioned in the thesis.

[1] H. Webster, *Webster's II New Riverside University Dictionary*, Boston, MA :Riverside, 1984.

[2] C. V. Ramamoorthy, and Benjamin W. Wah, Knowledge and data engineering, in *IEEE Trans. Knowledge Data Eng.*, vol. 1, no. 1, pp 9-15, 1989.

[3] Special Issue on Data Engineering, *IEEE Computer Mag.*, vol. 19, no. 1, Jan. 1986.

[4] G. Wiederhold, Knowledge and database management, *IEEE Software Mag.*, vol. 1, no. 1, pp. 63-73, Jan. 1984.

[5] J. D. Ullma, *Principles of Database Systems*, New York: Computer Science Press, 1984, pp. 211-267.

[6] A. Barr and E. A. Feigenbaum, *The Handbook of Artificial Intelligence*, vols 1, 2, and 3, Los Altos, CA: Kaufmann, 1981, 1982.

[7] J. McCarthy, Program with common sense, in *Mechanization of Thought Processes*, London: Her Majesty's Stationery Office, 1959, pp. 75-84.

[8] Y. Y. Tang, Chang D. Yan, and Ching Y. Suen, Document processing for automatic knowledge acquisition, in *IEEE Trans. Knowledge Data Eng.*, vol. 6, no. 1, pp. 3-21, 1994.

[9] Y. Y. Tang, Hong Ma, and Dihua Xi, Modified fractal signature (MFS): A new approach to document analysis for automatic knowledge acquistion, in *IEEE Trans. Knowledge Data Eng.*, vol. 9, no. 5, pp. 747-762, 1997.

[10] P. V. C. Hough, Methods and means for recognizing complex patterns, U. S. Patent 3, 069, 654, 1962.

[11] W. P. Birmingham and D. P. Siewiorek, Automated knowledge acquisition for a computer hardware systhesis system, *Knowledge Acquisition*, vol. 1, no. 4. pp. 321-340, 1989.

[12] J. H. Boose, A survey of knowledge acquisition techniques and tools, *Knowledge Acquisition*, vol. 1, no. 1, pp. 3-37, 1989.

[13] B. R. Gaines and J. H. Boose, Eds., *Knowledge Acquisition for Knowledge-Based Systems*, New York: Academic, 1988.

[14] R. G. Reynolds, J. I. Maletic, and S. E. Porvin, PM: A system to support the automatic and acquisition of programming knowledge, *IEEE Trans. Knowledge Data Eng.*, vol. 2, no. 3, pp. 273-282, 1990.

[15] H. A. Simon, Whether software engineering needs to be artificially intelligence, *IEEE Trans. Software Eng.*, vol. SE-12, no. 7, pp. 726-732, 1986.

[16] L. Abele, F. Wahl, and W. Scheri, Procedures for an automatic segmentation of text graphic and halftone regions in document, in *Proc. 2nd Scandinavian Conf. Image Analysis*, 1981, pp. 177-182.

[17] T. Aklyma and I. Masuda, A segmentation method for document images without the knowledge of document formats, *Trans. Japan. Inst. Electron. Commun. Engineers*, vol. J66-D, no. 2, pp. 111-118, 1982 (in Japanese).

[18] T. Aklyama and N. Hagita, Automated entry system for printed documents, *Pattern Recogn.*, vol. 23, no. 11, pp. 1141-1153, 1990.

[19] 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. Comput.*, vol. C-20, no. 12, pp. 1527-1543, 1971.

[20] N. Bartneck, Knowledge based address block finding using hybird knowledge representation schemes, in *Proc. 3rd USPS Advanced Technology Conf.*, pp. 249-263, 1988.

[21] N. J. Belkin, H. M. Brooks, and P. J. Daniels, Knowledge elicitation using discourse analysis, *Int. J. Man-Machine Studies*, vol. 27, pp. 127-144, 1987.

[22] A. Bergman, E. Bracha, P. G. Mulgaonkar, and T. Shaham, Advanced research in address block location, in *Proc. 3rd USPS Advanced Technology Conf.*, pp. 218-232, 1988.

[23] R. Beyth-Maron and S. Dekel, *An Elementary Approach to Thinking Under Uncertainty*, translated and adapted by S. Lichtenstein, B. Maron, and R. Beyth-Maron. Hillsdale, NJ: Lawrence Erlbaum, 1985.

[24] J. P. Bixler, Tracking text in mixted-mode document, in *Proc. ACM Conf. Document Processing Systems*, 1988, pp. 177-185.

[25] J. H. Boose and J. M. Bradshaw, Expertise transfer and complex problems: Using AQUINAS as a knowledge acquisition workbench for expert systems, *Int. J. Man-Machine Studies*, vol. 26, pp. 3-28, 1987.

[26] J. H. Boose and J. M. Bradshaw, AQUINAS: A knowledge acquisition workbench for building knowledge-based system, in *Proc. 1st European Workshop on Knowledge Acquisition for Knowledge-Based Systems*, Reading Univ., Sept. 1987, pp. A6, 1-6.

[27] J. M. Bradshaw, Strategies for selecting and interviewing experts, *Boeing Computer Services Tech. Rep.*.

[28] J. Breuker and B. Wielinga, Use of models in the interpretation of verbal data. in A. Kidd, *Knowledge Elicitation for Expert Systems: A Pratical Handbook*, New York: Plenum Press, 1987.

[29] A. M. Burton, N. R. Shadbolt, A. P. Hedgecock and G. Rugg, A formal evaluation of knowledge elicitation techniques for expert systems: Domain 1, in *Proc. 1st European Workshop on Knowledge Acquisition for Knowledge-Based Systems*, Reading Univ., Sept. 1987, pp. D3, 1-21.

[30] R. G. Casey and D. R. Ferguson, Intelligent forms processing, *IBM Syst. J.*, vol 29, no. 3, pp. 435-450, 1990.

[31] G. Ciardiello, M. P. Poccotelli, G. Seafuro, and M. R. Spada, An experimental system for office document handing and text recognition, in *Proc. 9th Int. Conf. Pattern Recognition*, 1988, pp. 739-743.

[32] D. A. Cleaves, Cognitive biased and corrective techniques: proposals for improving elicitation procedures for knowledge-based systems, *Int. J. Man-Machine Studies*, vol 27, pp. 155-166, 1987.

[33] V. Demjanenko, Y. C. Shin, R. Sridhar, P. Palumbo, and S. Srihari, Real-time connected component analysis for address block location, in *Proc. 4th USPS Advanced Technology Conf.*, 1990, pp. 1059-1071.

[34] A. Dengel and G. Barth, Document description and analysis by cuts, in *Proc. RIAO*, Massachusetts Inst. of Technology, 1988.

[35] A. Dengel and G. Barth, High level document analysis guided by geometric aspects, *Int. J. Pattern Recogn. and Artificial Intell.*, vol. 2, no. 4, 641-655, 1988.

[36] A. Dengel, Document image analysis–expectation-driven text recognition, in *Proc. Syntactic and Structural Pattern Recogn. (SSPR90)*, 1990, pp. 78-87.

[37] W. Doster, Different states of a document's content on its way from the gutenbergian world to the electronic world, in *Proc. 7th Int. Conf. Pattern Recogn.*, 1984, pp. 872-874.

[38] A. C. Downton and C. G. Leedham, Preprocessing and presorting of envelope images for automatic sorting using OCR, *Pattern Recogn.*, vol. 23, no. 3/4, pp. 347-362, 1990.

[39] D. G. Elliman and I. T. Lancaster, A review of segmentation and contextual analysis techniques for text recognition, *Pattern Recogn.*, vol. 23, no. 3/4, pp. 337-346, 1990.

[40] F. Esposito, D. Malerba, G. Semeraro, E. Annese, and G. Scafuro, An experimental page layout recognition system for office document automatic classification: An intergrated approach for inductive generalization, in *Proc. 10th Int. Conf. Pattern Recogn.*, 1990, pp. 567-572.

[41] J. L. Fisher, S. C. Hinds, and D. P. D'Amato, A rule-based system for document image segmentation, in *Proc. 10th Int. Conf. Pattern Recogn.*, 1990, pp. 567-572.

[42] L. A. Fletcher and R. Kasturi, A robust algorithm for text string separation from mixted text/graphics images, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, no. 6, pp. 910-918, 1988.

[43] Freiling, J. Alexander, S. Messick, S. Rehfuss, and S. Shulman, Stating a knowledge engineering project: A step-by-step approach, *AI Mag.*, vol. 6, no. 3, pp. 150-164, Fall 1985.

[44] H. Fujisawa *et al.*, Document analysis and decomposition method for multimedia contents retrieval, in *Proc. 2nd Int. Symp. Interoperable Inform. Syst.*, 1988, pp. 231-238.

[45] H. Fujisawa and Y. Nakano, A top-down approach for analysis of document images, in *Proc. SSPR90*, 1990, pp. 113-122.

[46] B. R. Gaines, An overview of knowledge acquisition and transfer, *Int. J. Man-Machine Studies*, vol. 26, pp. 453-472, 1987.

[47] J. G. Gammack and R. M. Young, Psychological techniques for eliciting expert knowledge, in R and D in expert system, in *Proc. 4th Expert System Conf.*, Warwick, England, 1984. Max Bramer, Ed. Cambridge: Cambrige University Press.

[48] U. Gappa, Classical: A knowledge acquisition system facilitating the formalization of advanced aspects in heuristic classification, in *Proc. 2nd European Knowledge Acquisition Workshop (EKAW-88)*, Bonn, Germany, June 1988, pp. 19, 1-16.

[49] R. C. Gonzalez and P. Wintz, *Digital Image Processing*, Reading, MA: Addison-Wesley, 1987.

[50] T. R. Gruber, Acquiring strategic knowledge from experts, *Int. Man-Machine Studies*, vol. 29, pp. 579-597, 1988.

[51] S. Guiasu, *Information Theory with Applications*, New York: McGraw-Hill, 1977.

[52] J. Higashino, H. Fujisawa, Y. Nakano and M. Ejiri, A knowledge-based segmentation method for document understanding, in *Proc. 8th Int. Conf. Pattern Recogn*, 1986, pp. 745-748.

[53] S. C. Hinds, J. L. Fisher, and D. P. D'Amato, A document skew detection method using run-length encoding and the Hough transform, in *Proc. 10th Int. Conf. Pattern Recogn.*, 1990, pp. 464-468.

[54] R. F. Hink and D. L. Woods, How humans process uncertain knowledge: An introduction for knowledge engineers, *AI Mag.*, vol. 8, no. 3, pp. 41-53, Fall 1987.

[55] W. Horak, Office document architecture and office document interchange formats: Current status of international standardization, *IEEE Comput.*, Oct. 1985, pp. 50-60.

[56] A. Jain and B. Yu, Document representation and its application to page segmentation, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 3, pp. 294-308, 1998.

[57] X. Chen, Non-parameter statistic estimation, 1997.

‹2› Additional references which are alphabetically sorted. For the people who are interested in this topic can find more detailed information in these references.

[58] M. F. Barnsley and A. D. Sloan, A better way to compress images, *Byte*, pp. 215-223, Jan. 1988.

[59] B. Couanon and J. Camillerapp, A way to separate knowledge from program in structured document analysis: Application to optical music recognition, *Proc. 3rd Int. Conf. Document Analysis and Recogn.*, pp. 1092-1097, Montreal, 1995.

[60] A. Dengel, Initial Learning of Document Structure, in *Proc. 2nd Conf. Document Analysis and Recogn.*, pp. 86-90, Tsukuba, Japan, 1993.

[61] K. L. Falconer, *Fractal Geometry: Mathematical Foundation and Applications*, New York: Wiley, 1990.

[62] K. L. Falconer, *The Geometry of Fractal Sets*, Cambridge, England: Cambridge Univ. Press, 1985.

[63] M. Hose and Y. Hoshino, Segmentation method of document images by two-dimensional Fourier transformation, *Syst. Comput. in Japan.*, vol. 16, no. 3, pp. 38-47, 1985.

[64] H. S. Hou, *Digital Document Processing*, New York: Wiley, 1983.

[65] K. Inagaki, T. Kato, T. Hiroshima, and T. Sakai, MACSYM: A hierachical parallel image processing system for even-driven pattern understanding of documents, *Pattern Recogn.*, vol. 17, no. 1, pp. 85-108, 1984.

[66] *ISO 8613: Information Processing-Text and Office Systems-Office Document Architecture (ODA) and Interchange Format*, International Organization for Standardization, 1989.

[67] O. Iwaki, H. Kida, and H. Arakawa, A character/graphic segmentation method using neighborhood line density, *Trans. Inst. Electron. Commun. Engineers of Japan*, Part D, vol. J68D, no. 4, pp. 821-828, 1985.

[68] O. Iwaki, H. Kida, and H. Arakawa, A segmentation method based in office document hierarchical structure, in *Proc. IEEE Int. Conf. Syst. Man. Cybernetics*, Alexandria, VA, Oct. 1987, pp. 759-763.

[69] C. Jacobson and M. J. Freiling, ASTEK: A multi-paradigm knowledge acquisition tool for complex strutured knowledge, *Int. J. Man-Machine Studies*, vol. 29, pp. 311-327, 1988.

[70] V. Jagannathan and A. S. Elmaghraby, MEDKAT: Multiple expert delphi-based knowledge acqisition tool, in *Proc. ACM NE Regional Conf.*, Boston, Oct. 1985, pp. 103-110.

[71] L. Johnson and N. E. Johnson, Knowledge elicitation involving teachback interviewing, in A. Kidd, *Knowledge Eliciatation for Expert Systems: A Pratical Handbook*, New York: Plenum, 1987.

[72] E. G. Johnson, Short note: Printed text discrimination, *Comput. Graphics Image Processing*, vol. 3, no. 1, pp. 83-89, 1974.

[73] J. Kanai, M. S. Krishnamoorthy, and T. Spencer, Algorithms for manipulating nested block represented images, in *Advance Printing of Paper Summaries. SPSE's 26th Fall Symp.*, Arlington, VA, Oct. 1986, pp. 190-193.

[74] H. Kato and S. Inokuchi, The recognition system for printed piano music using musical knowledge and constraints, in *Proc. SSPR90*, 1990, pp. 231-248.

[75] H. Kida, O. Iwaki, and K. Kawada, Document recognition system for office automation, in *Proc. 8th Int. Conf. Pattern Recogn.*, 1986, pp. 446-448.

[76] A. L. Kidd and M. B. Cooper, Man-Machine interface issues in the construction and use of an expert system, *Int. J. Man-Machine Studies.*, no. 22, pp. 91-102, 1985.

[77] Y. Kodratoff and G. Tecuci, Learning at different level of knowledge, in *Proc. 2nd European Knowledge Acquisition Workshop (EKAW-88)*, Bonn, Germany, June 1988, pp. 3.1-17.

[78] J. Kreich, A. Luhn, G. Maderlechner, Knowledge based interpretation of scanned business letters, in *Proc. IAPR Workshop on CV*, 1988, pp. 417-420.

[79] K. Kubota, O. Iwaki, and H. Arakawa, Image segmentation techniques for document processing, in *Proc. 1983 Int. Conf. Text Processing with a Large Character Set*, 1983, pp. 73-78.

[80] K. Kubota, O. Iwaki, and H. Arakawa, Document understanding system, in *Proc. 7th Int. Conf. Pattern Recogn*, 1984, pp. 612-614.

[81] D. Lenat, M. Prakish, and M. Shepard, CYC: Using common sense knowledge to overcome brittleness and knowledge acquisition bottle-necks, *AI Mag.*, vol 6, no. 4, pp. 65-85, Winter 1985.

[82] M. Linster, Kriton: A knowledge elicitation tool for expert systems, in *Proc. 2nd European Knowledge Acquisition Workshop (EKAW-88)*, Bonn, Germany, June 1988, pp. 4.1-9.

[83] H. Makino, Representation and segmentation of document images, in *Proc. IEEE Comput. Soc. Conf. Pattern Recogn. and Image Processing*, 1983, pp. 291-296.

[84] B. B. Mandelbrot, *The Fractal Geometry of Nature*, New York: Freeman, 1982/1983.

[85] P. Maragos and F. K. Sun, Measuring the Fractal Dimension of Signals: Morphological Covers and Iterative Optimization, *IEEE Trans. Signal Processing*, vol. 41, no. 1, pp. 108-121, Jan. 1993.

[86] I. Masuda, N. Hagita, T. Aklyama, T. Takahashi, and S. Naito, Approach to smart document reader system, in *Proc. CVPR'85*, 1985, pp. 550-557.

[87] J. R. Munkres, *Topology, A First Course*, Englewood Cliffs: N.J.: Prentice Hall, Inc., 1975.

[88] R. S. Michalski, Theory and methodology of inductive learning, *Machine Learning, An Artificial Intelligence Approach*, Palo Alto, CA: Tioga, 1983.

[89] K. Morik, Acquiring domain models, *Int. J. Man-Machine Studies*, vol. 26, pp. 93-104, 1987.

[90] G. Nagy, A preliminary investigation of techniques for the automated reading of unformatted text, Comm. ACM, vol. 11, no. 7, pp. 480–487, 1968.

[91] G. Nagy, At the frontiers of OCR, in *Proc. IEEE*, vol. 80, pp. 1093-1100, 1992.

[92] G. Nagy, S. Seth, and M. Viswanathan, A Prototype Document Image Analysis System for Technical Journals, *Computer*, vol. 25, pp. 10-22, 1992.

[93] G. Nagy and S. Seth, Hierachical representation of optically scanned documents, in *Proc. 7th Int. Conf. Pattern Recogn.*, 1984, pp. 247-349.

[94] G. Nagy, Towards a structured-document-image utility, in *Proc. SSPR90*, 1990, pp. 293-309.

[95] G. Nagy, J. Kanal, and M. Krishnamoorthy, Two complementary techniques for digitized document analysis, in *Proc. ACM Conf. on Document Processing Systems*, 1988, pp. 169-176.

[96] G. Nagy, S. C. Seth, and S. D. Stoddard, Document analysis with an expert system, *Pattern Recogn. Pratice II*, New York: Elsevier, 1986, pp. 149-159.

[97] Y. Nakano, H. Fujisawa, O. Kunisaki, K. Okada, and T. Hananoi, A document understanding system incorporating with character recognition, in *Proc. 8th Int. Conf. Pattern Recogn.*, 1986, pp. 801-803.

[98] D. Niyogi and S. N. Srihari, A rule-based system for document understanding, in *Proc. AAAI'86*, 1986, pp. 801-803.

[99] P. T. Nguyen and J. Quinqueton, Space Filling Curves and Texture Analysis, *Proc. Int. Conf. Pattern Recognition*, Munich, Germany, pp. 282-285, 1982.

[100] K. Nygaard and O. J. Dahl, The development of the SIMULA languages, in *History of Programming Languages*, R. L. Wexelblat, Ed. New York: Academic, 1981, pp. 439-480.

[101] E. A. Parrish, Jr, A foreword to knowledge and data engineering, *IEEE Trans. Knowledge Data Eng.*, vol. 1, no. 1, pp. 9-15, 1989.

[102] T. Pavlidis, *Algorithm for Graphics and Image Processing*, Rockville, MD: Computer Science Press, 1982.

[103] S. Peleg, J. Naor, R. Hartley, and D. Avnir, Multiple resolution texture analysis and classification, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, no. 4, pp. 518-523, July 1984.

[104] A. Pentland, Frantal-based description of natural scences, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, no. 11, pp. 661-674, Nov. 1984.

[105] C. V. Ramamoorthy and B. W. Wah, Knowledge and data engineering, *IEEE Trans. Knowledge Data Eng.*, vol. 1, no. 1, pp. 9-15, 1989.

[106] A. Rastogi and S. N. Srihari, Recognizing textual blocks in document images using the Hough transform, Dept. of Computer Science, State Univ. of New York, Buffalo, Tech, Rep. 86-01, 1986.

[107] R. A. Rusk and R. M. Krone, The Crawford slip method (CSM) as a tool for extraction of expert knowledge, in *Human-Computer Interaction*, G. Salvendy, Ed. New York: Elsevier, pp. 279-282.

[108] W. Scherl, F. Wahl, and H. Fuchsberger, Automatic separation of text, graphic and picture segments in printed material, *Pattern Recogn. in Pratice*, 1980, pp. 213-221.

[109] C. E. Shannon, A mathematical theory of communication, *Bell Syst. Tech. J.*, vol. 27, pp. 379-423, 1948.

[110] M. L. G. Shaw, Problems of validation in a knowledge acquisition system using multiple experts, in *Proc. 2nd European Knowledge Acquisition Workshop (EKAW-88)*, Bonn, Germany, June 1988, pp. 5.1-15.

[111] Y. Shima, T. Murakami, and M. Koga, A high speed algorithm for propagation-type labelling based on block sorting of runs in binary images, in *Proc. 10th Int. Conf. Pattern Recogn.*, 1990, pp. 655-658.

[112] S. N. Srihari, C. H. Wang, P. W. Palumbo, and J. J. Hull, Recognizing address blocks on mail pieces: specialized tools and problem-solving architecture, *AI Mag.*, vol. 8, no. 4, pp. 25-40, 1987.

[113] S. N. Srihari and V. Govindaraju, Analysis of textual images using the Hough transform, *Machine Vision Appl.*, vol. 2, pp. 141-153, 1989.

[114] H. Stephanou, Perpectives on imperfect information processing, *IEEE Trans. Syst. Man Cybern.*, vol. 17, pp. 780-798, 1987.

[115] C. Y. Suen, Y. Y. Tang, and C. D. Yan, Document layout and logical model: A general analysis for document processing, Center for Pattern Recognition and Machine Intelligence (CENPARM), Concordia Univ., Tech. Rep., 1989.

[116] C. Y. Suen, C. D. Yan, and Y. Y. Tang, Document analysis and understanding: A method for automated acquisition of data and knowledge, Center for Pattern Recognition and Machine Intelligence (CENPARM), Concordia Univ., Tech. Rep., 1990.

[117] Y. Y. Tang, C. Y. Suen, and C. D. Yan, Chinese form pre-processing for automatic entry, in *Proc. Int. Conf. Computer Processing of Chinese and Oriental Languages*, Taipei, Taiwan, Aug. 13-16, 1991, pp. 313-318.

[118] Y. Tsuji, Document image analysis for generating synatic structure description, in *Proc. 9th Int. Conf. Pattern Recogn.*, 1988, pp. 744-747.

[119] S. Tsujimoto and H. Asada, Understanding multi-articled documents, in *Proc. 10th Int. Conf. Pattern Recogn.*, 1990, pp. 551-556.

[120] M. Viswanathan, Analysis of scanned documents–a syntactic approach, in *Proc. SSPR90*, 1990, pp. 450-459.

[121] F. Wahl, L. Abele, and W. Scheri, Merkmale fuer die segmentation von dokument zur automatischen textverabeitung, in *Proc. 4th DAGM Symp.*, 1981.

[122] C. H. Wang, P. W. Palumbo, and N. Srihari, Object recognition in visually complex documents: An architecture for locating address blocks on mail pieces, in *Proc. 9th Int. Conf. Pattern Recogn.*, 1988, pp. 365-367.

[123] D. Wang and S. N. Srihari, Classification of newspaper image blocks using texture analysis, *CVGIP*, vol. 47, pp. 327-352, 1989.

[124] S. Watanabe, and T. Kaminuma, Rccent developments of the minimum entropy algorithm, in *Proc. 9th Int. Conf. Pattern Recogn.*, Rome, Nov. 1988, pp. 536-540.

[125] M. Welbank, Knowledge acquisition update, *Insight Study*, no. 5, System Designers, 1987.

[126] K. Y. Wong, R. G. Casey, and F. M. Wahl, Document analysis system, *IBM J. Res. Develop.*, vol. 26, no. 6, pp. 647-656, 1982.

[127] M. Yanada and K. Hasuike, Document image processing based on enchanced border following algorithm, in *Proc. 10th Int. Conf. Pattern Recogn.*, 1990, pp. 551-556.

[128] C. D. Yan, Y. Y. Tang, and C. Y. Suen, Form inderstanding system based on form description language, in *Proc. 1st Int. Conf. Document Analysis and Recogn.*, Saint-Malo, France, Sept. 30-Oct. 2, 1991, pp. 283-293.

[129] P. S. Yeh, S. Antoy, A. Litcher, and A. Rosenfeld, Address location on envelopes, *Pattern Recogn.*, vol. 20, no. 2, pp. 213-227, 1987.

[130] R. Young and J. Gammack, Role of psychological techniques and intermediate representation in knowledge elicitation, in *Proc. 1st European Workshop on Knowledge-Based Systems*, Reading Univ., Sept. 1987, pp. D7.1-5.

[131] T. Y. Young and K. S. Fu, *Handbook of Pattern Recogn. and Image Processing*, New York: Academic, 1986.

[132] B. Yu, Automatic understanding of symbol-connected diagrams, *Proc. 3rd Int. Conf. Document Analysis and Recogn.*, pp. 803-806, Montreal, 1995.

[133] B. Yu and A. Jain, A Generic System for Form Dropout, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, pp. 1127-1134, 1996.

[134] B. Yu, A. Jain, and M. Mohiuddin, Address Block Location on Complex Mail Pieces, *Proc. 4th Conf. Document Analysis and Recogn*, Ulm, Germany, 1997.

Appendix A

Figures

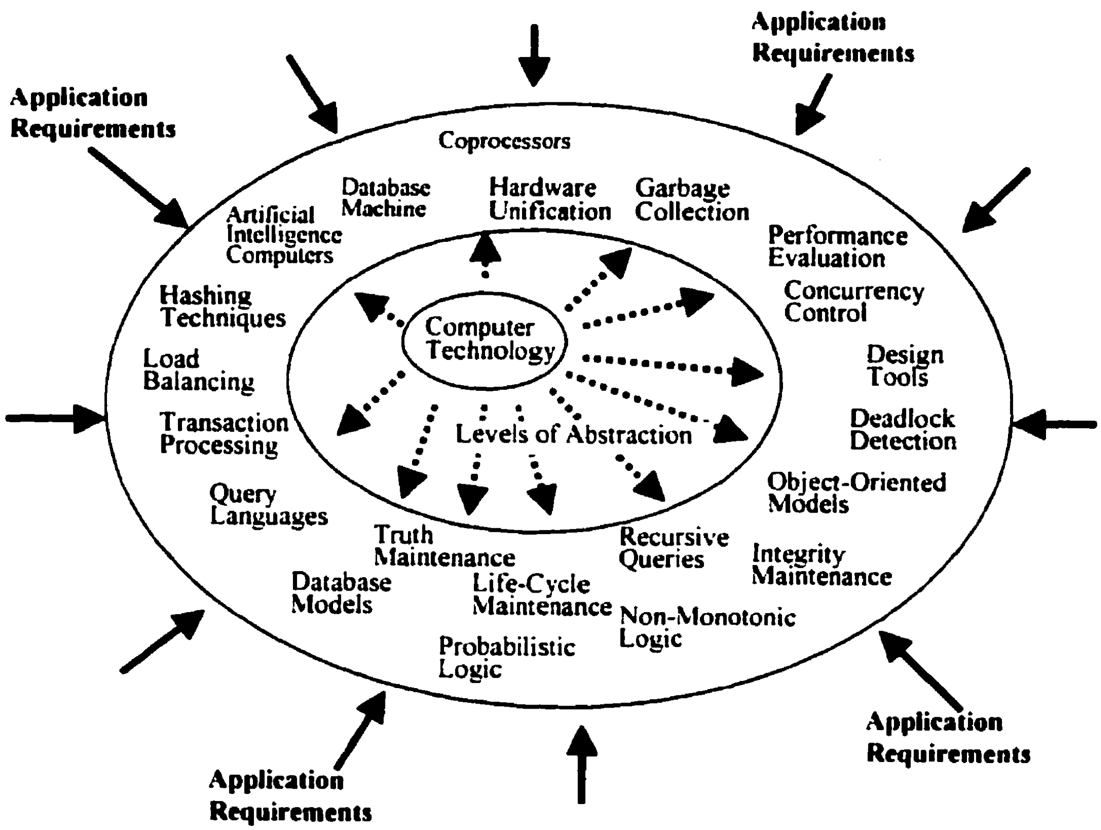


Figure A.1: Problems with various levels of abstractions in data and knowledge engineering.

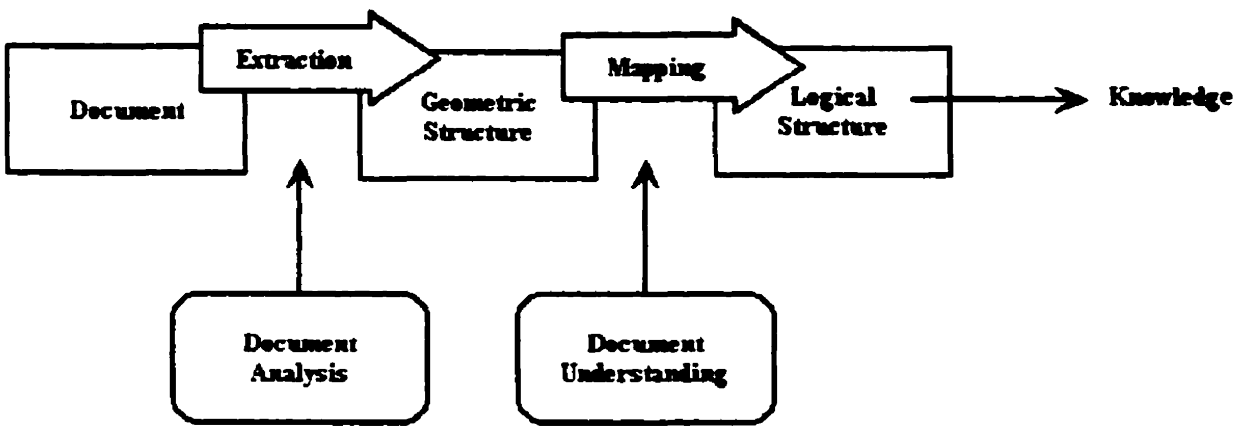


Figure A.2: Document processing and knowledge acquisition.

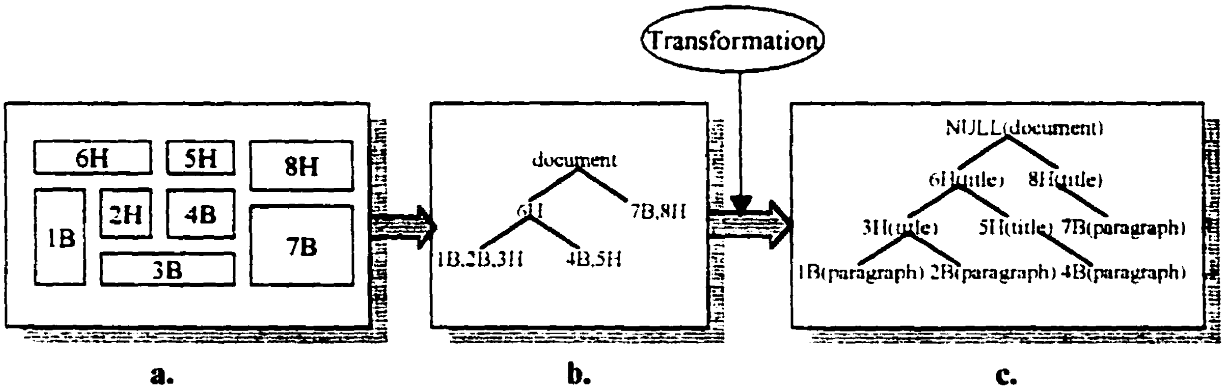
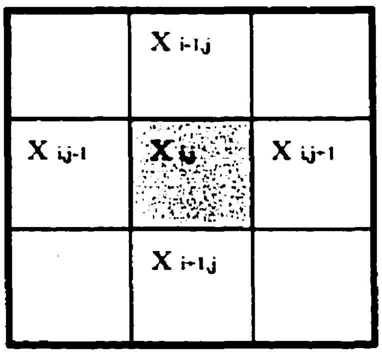
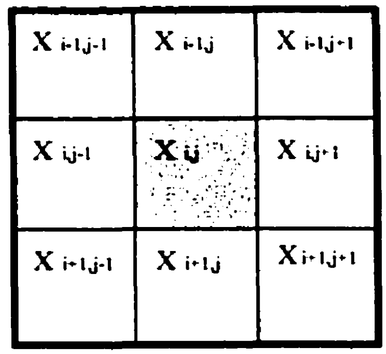


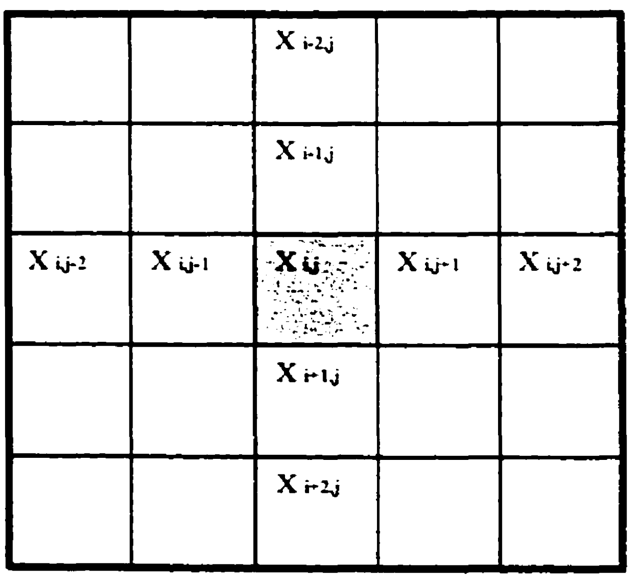
Figure A.3: Geometric structure and logical structure of a document.



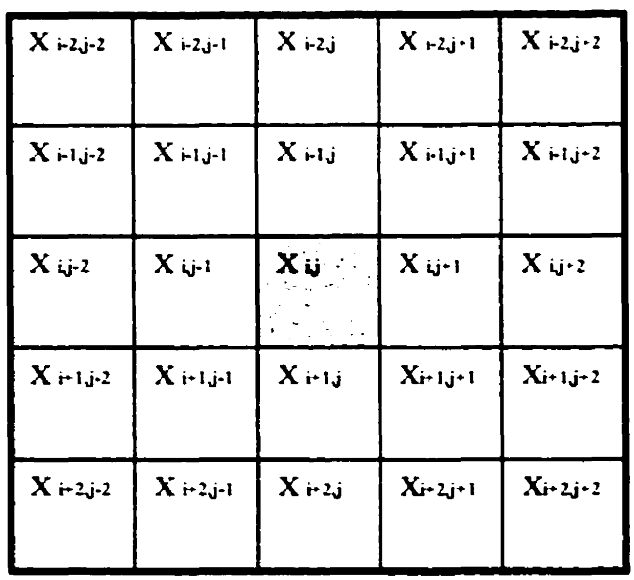
(a) Four-connected neighborhood (n=5)



(b) Eight-connected neighborhood (n=9)

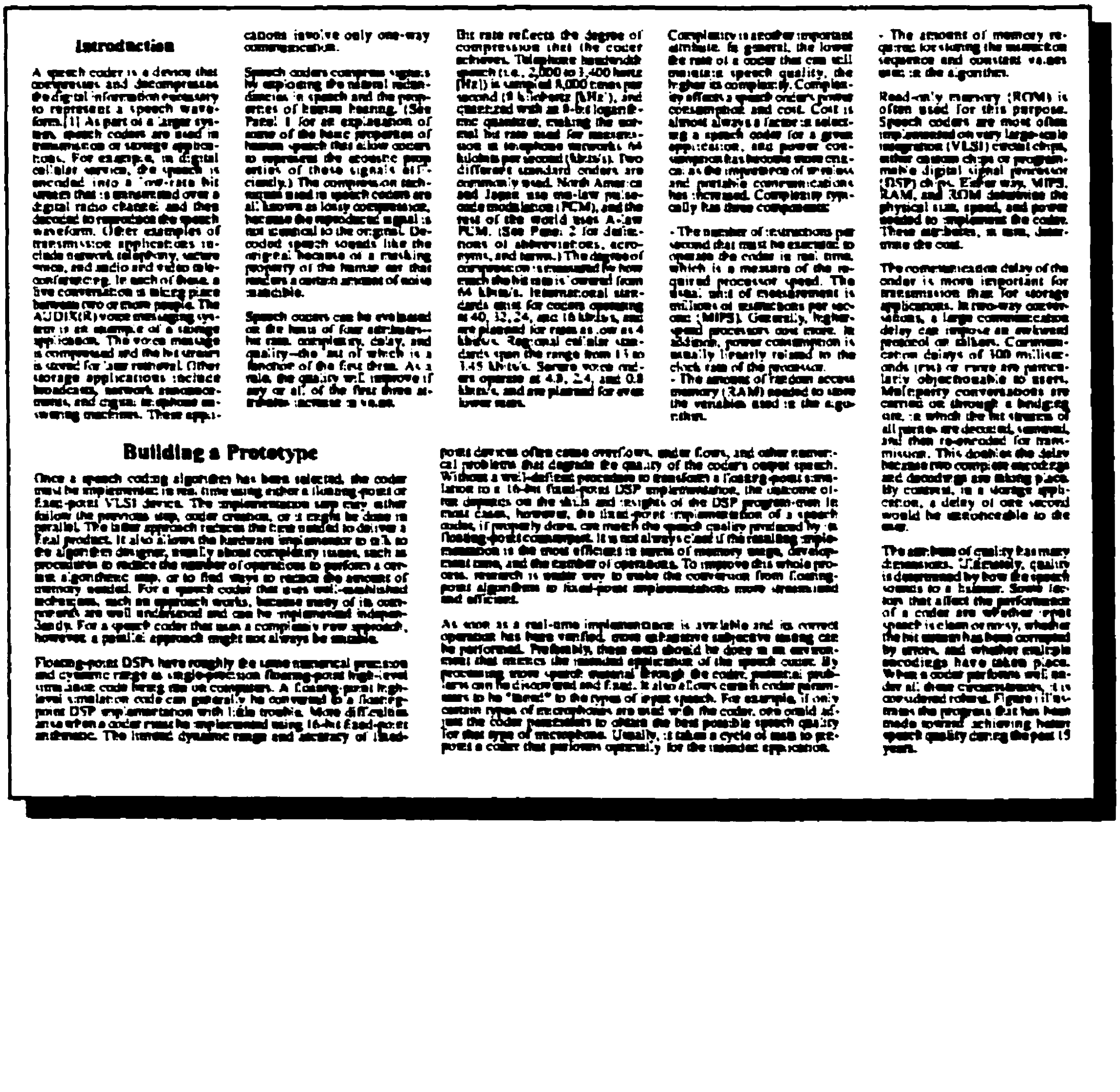


(c) Twelve-connected neighborhood (n=13)



(d) Twenty four-connected neighborhood (n=25)

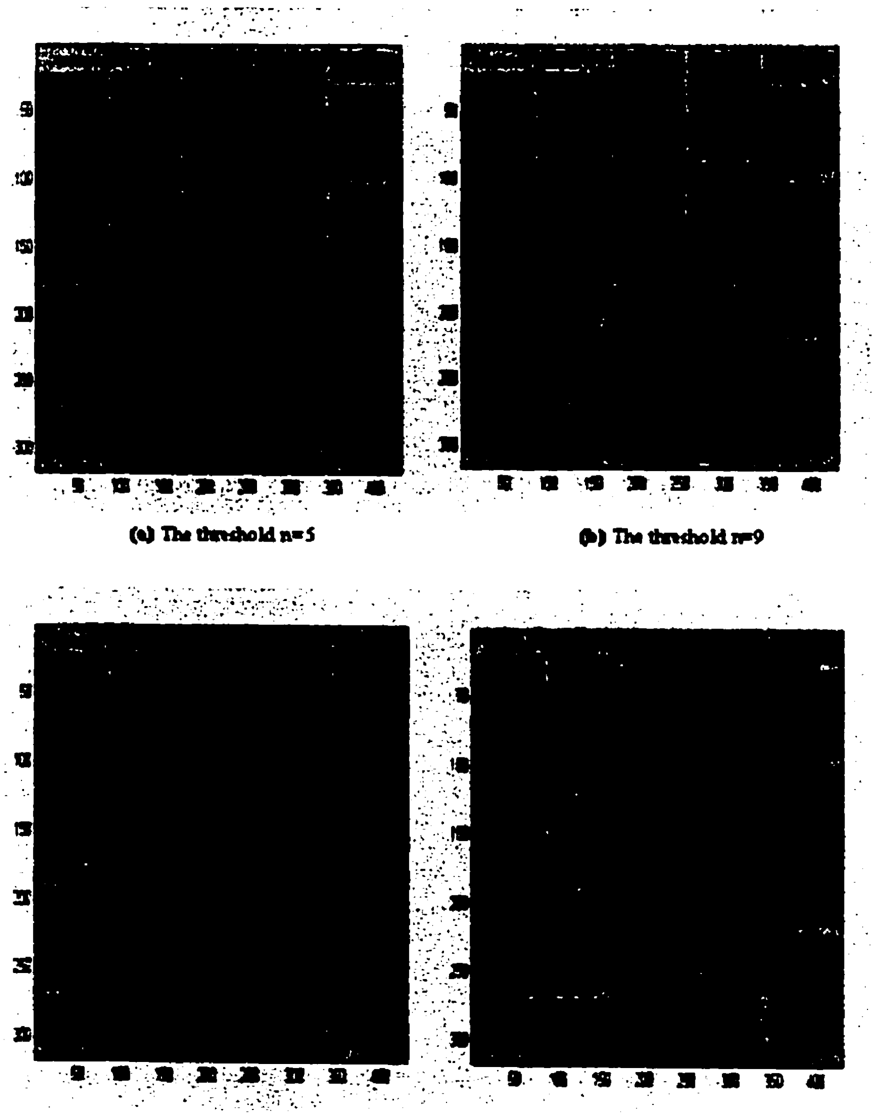
Figure A.4: An illustration of connected neighborhood (filter windows).



Introduction

Building a Prototype

Figure A.5: An illustration of original document image (document 1).



(a) The threshold n=5

(b) The threshold n=9

(c) The threshold n=13

(d) The threshold n=25

Figure A.6: An illustration of page segmentation of document 1 with different thresholds.

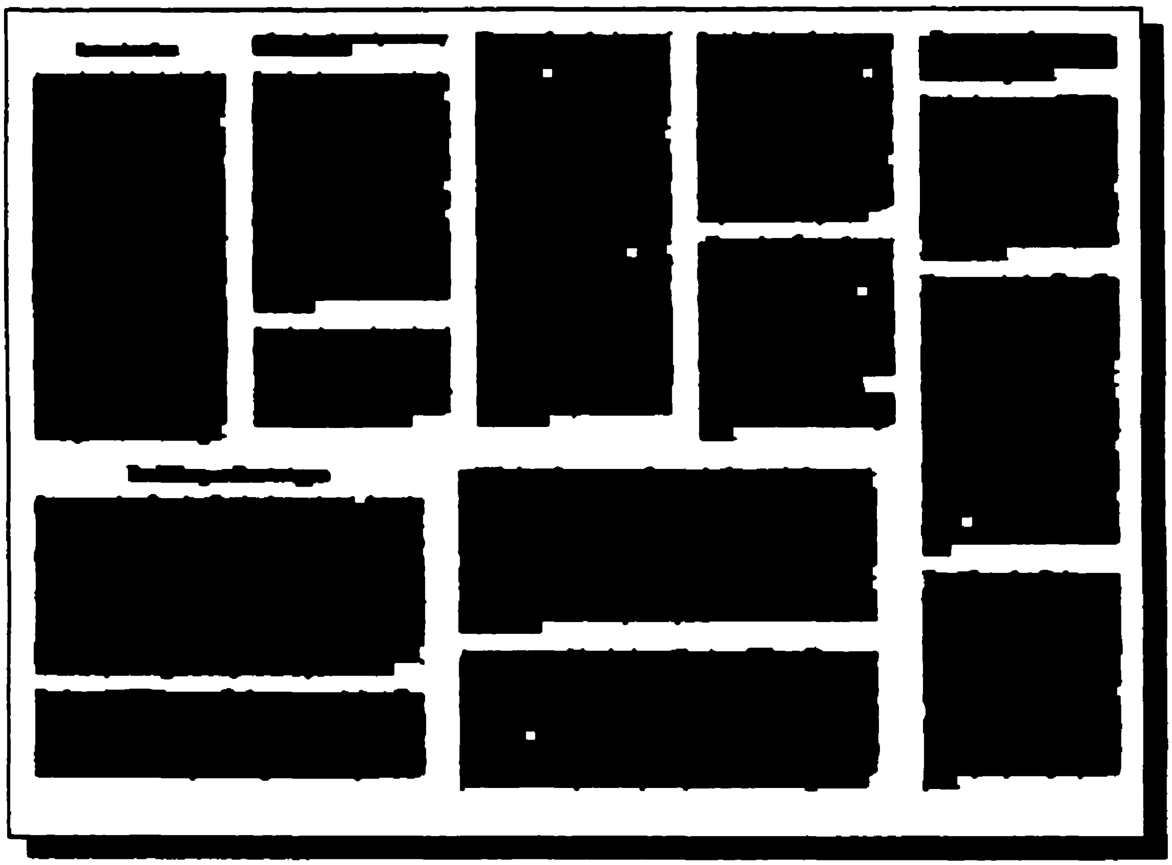
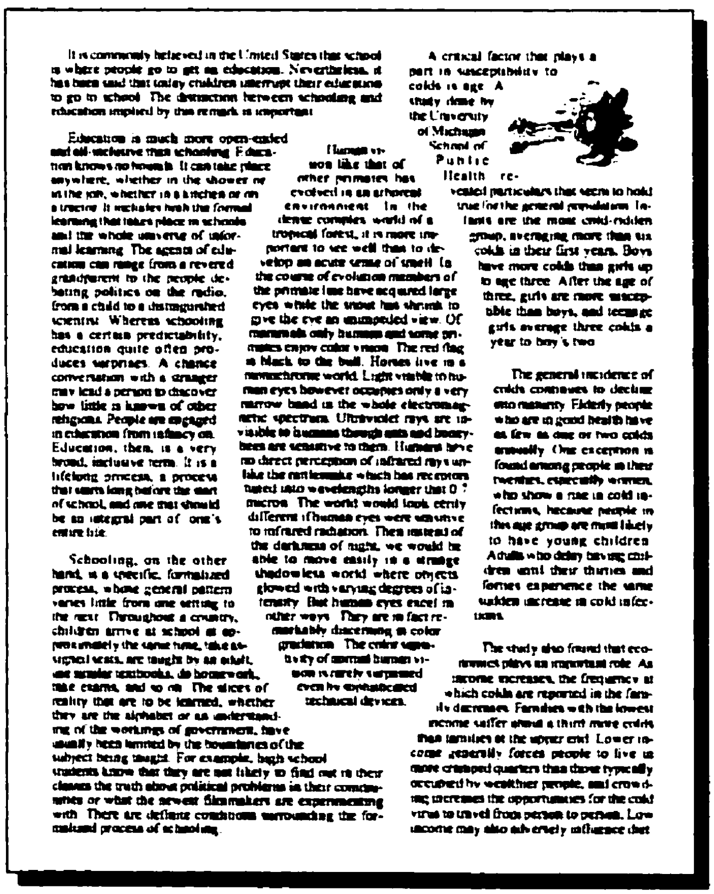


Figure A.7: An illustration of page segmentation of document 1.



It is commonly believed in the United States that school is where people go to get an education. Nevertheless, it has been said that today children interrupt their education to go to school. The distinction between schooling and education implied by this remark is important.

Education is much more open-ended and all-inclusive than schooling. Education knows no bounds. It can take place anywhere, whether in the shower or on the job, whether in a kitchen or on a tractor. It includes both the formal learning that takes place in schools and the whole universe of informal learning. The agents of education can range from a revered grandparent to the people debating politics on the radio, from a child to a distinguished scientist. Whereas schooling has a certain predictability, education quite often produces surprises. A chance conversation with a stranger may lead a person to discover how little is known of other religions. People are engaged in education from infancy on. Education, then, is a very broad, inclusive term. It is a lifelong process, a process that starts long before the start of school, and one that should be an integral part of one's entire life.

Schooling, on the other hand, is a specific, formalized process, whose general pattern varies little from one setting to the next. Throughout a country, children arrive at school at approximately the same time, take assigned seats, are taught by an adult, use similar textbooks, do homework, take exams, and so on. The slices of reality that are to be learned, whether they are the alphabet or an understanding of the workings of government, have usually been limited by the boundaries of the subject being taught. For example, high school students know that they are not likely to find out in their classes the truth about political problems in their communities or what the newest filmmakers are experimenting with. There are definite conditions surrounding the formalized process of schooling.

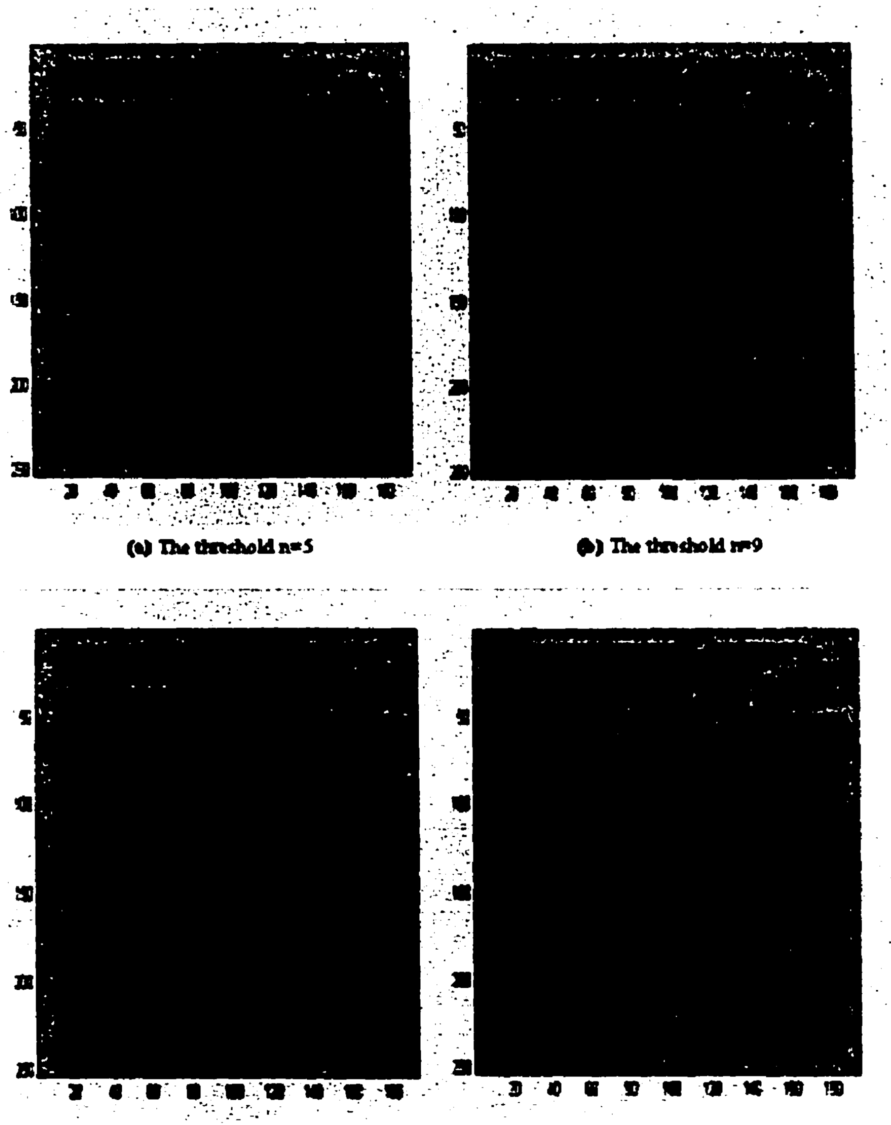
Human vision like that of other primates has evolved in an arboreal environment. In the dense complex world of a tropical forest, it is more important to see well than to develop an acute sense of smell. In the course of evolution members of the primate line have acquired large eyes while the snout has shrunk to give the eye an unimpeded view. Of mammals only humans and some primates enjoy color vision. The red flag is black to the bull. Horses live in a monochrome world. Light visible to human eyes however occupies only a very narrow band in the whole electromagnetic spectrum. Ultraviolet rays are invisible to humans though ants and honeybees are sensitive to them. Humans have no direct perception of infrared rays unlike the rattlesnake which has receptors tuned into wavelengths longer than 0.7 micron. The world would look eerily different if human eyes were sensitive to infrared radiation. Then instead of the darkness of night, we would be able to move easily in a strange shadowless world where objects glowed with varying degrees of intensity. But human eyes excel in other ways. They are in fact remarkably discerning in color gradation. The color sensitivity of normal human vision is rarely surpassed even by sophisticated technical devices.

A critical factor that plays a part in susceptibility to colds is age. A study done by the University of Michigan School of Public Health revealed particulars that seem to hold true for the general population. Infants are the most cold-ridden group, averaging more than six colds in their first years. Boys have more colds than girls up to age three. After the age of three, girls are more susceptible than boys, and teenage girls average three colds a year to boy's two.

The general incidence of colds continues to decline into maturity. Elderly people who are in good health have as few as one or two colds annually. One exception is found among people in their twenties, especially women, who show a rise in cold infections, because people in this age group are most likely to have young children. Adults who delay having children until their thirties and forties experience the same sudden increase in cold infections.

The study also found that economics plays an important role. As income increases, the frequency at which colds are reported in the family decreases. Families with the lowest income suffer about a third more colds than families at the upper end. Lower income generally forces people to live in more cramped quarters than those typically occupied by wealthier people, and crowding increases the opportunities for the cold virus to travel from person to person. Low income may also adversely influence that

Figure A.8: An illustration of original document image (document 2).



(a) The threshold n=5

(b) The threshold n=9

(c) The threshold n=13

(d) The threshold n=25

Figure A.9: An illustration of page segmentation of document 2 with different thresholds.

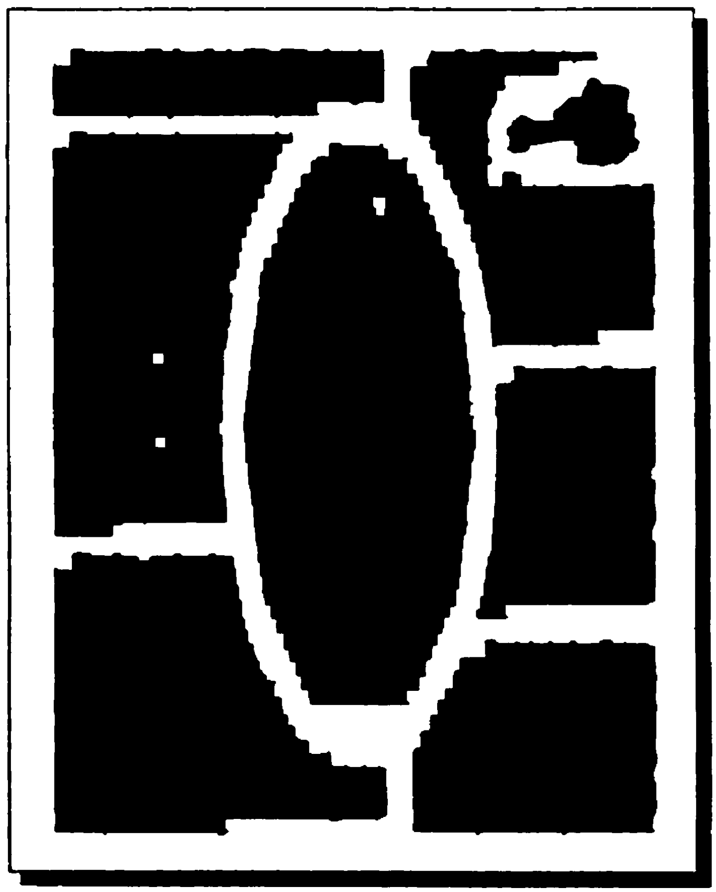
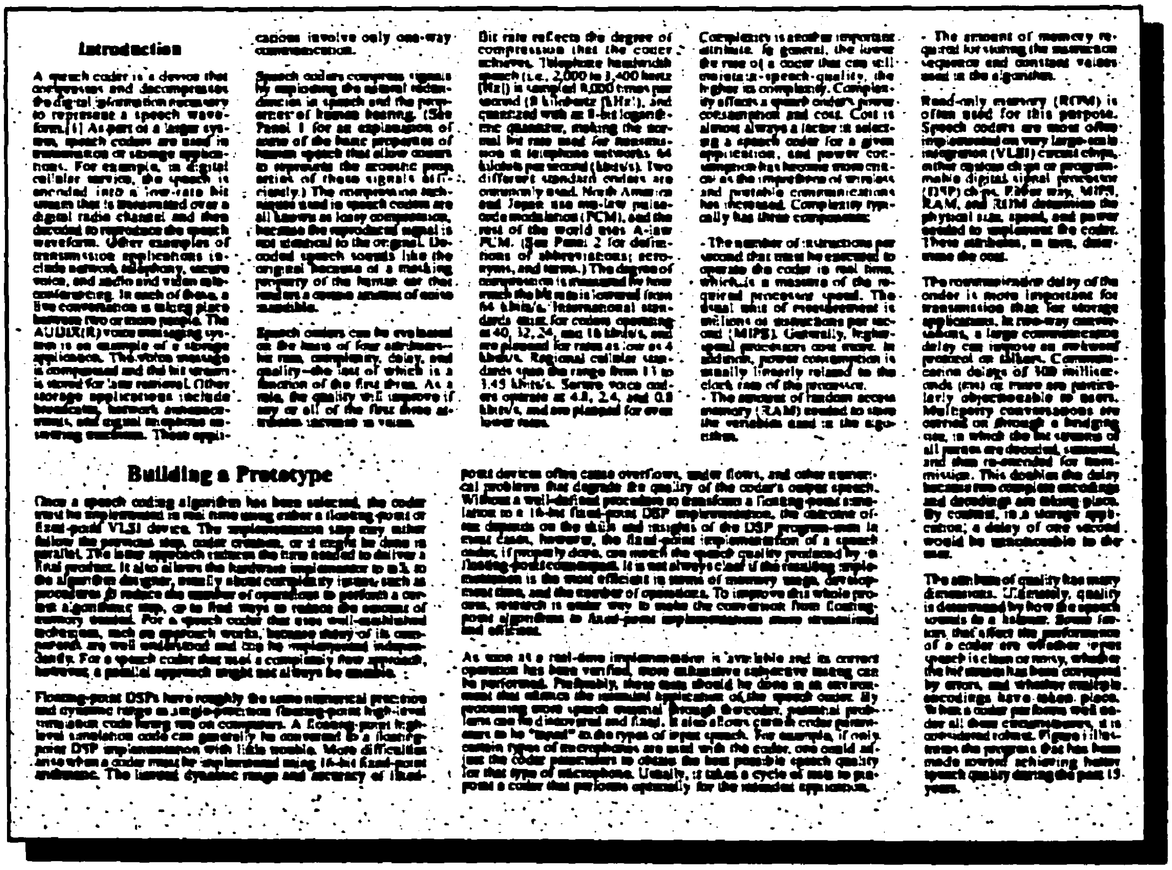


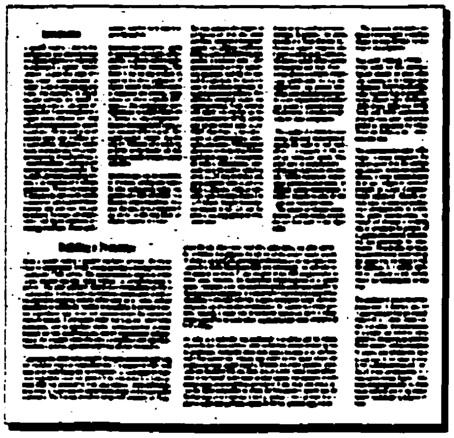
Figure A.10: An illustration of page segmentation of document 2.



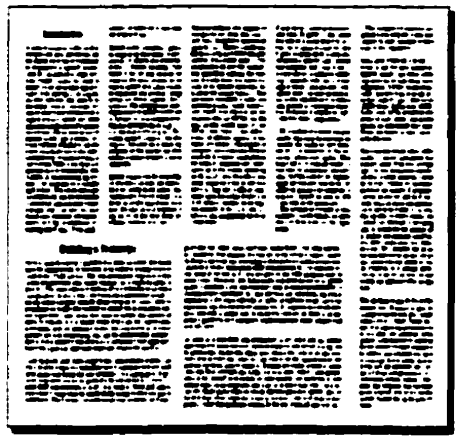
Introduction

Building a Prototype

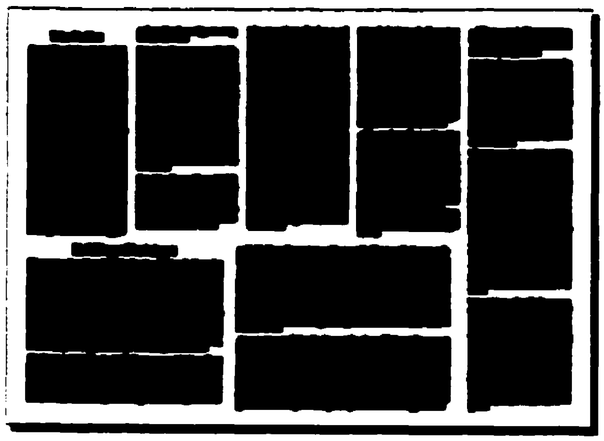
Figure A.11: An illustration of original document image (document 3).



(a)



(b)



(c)

(a) The original document image which is corrupted by noise;
(b) The clean document image after using the MedOSF;
(c) Page segmentation of document 3.

Figure A.12: An illustration of page segmentation of document 3.

Appendix B

Partial Matlab code

B.1 Simulation 1 and 2: the MaxOSF approach with threshold 5, 9, 13, 25.

```
%==================================================%
%      Implementation of the Algorithm             %
%              Li  Ma  (9745151)                   %
%              December 2000                       %
%==================================================%
%  MaxOSF Algorithm: A new approach for page       %
%  segmentation which is based on the max order    %
%  statistic filter (MaxOSF).                      %
%                                                  %
%  INPUT:    A page of document image(Doc01.bmp).  %
```

```
% OUTPUT:   Geometric structure of the document.  %
% COMPLIER: Using Matlab under the Windows        %
%           enviornment.                          %
%=================================================%
%================KEY WORD========================%
%    THRESHOLD n: is the filter windows length    %
%    which is the no. of the connected neighbour  %
%    hood plus 1.                                 %
%=================================================%
clear all;

%====================INPUT STARTS================%
%Read image from a graphic file (BMP file) into A
%whose class is unsigned 8-bit integer (uint8).
A=imread('Doc01','bmp');

%===============THE ALGORITHM STATS==============%
%Double convert A to double precision.
a=double(A);
%Find the length of vector a.
SizeA = size(a);

%-----------------------------------------------%
```

```
% Experiment 1: Using max filter with threshold 5.%
%-----------------------------------------------%
%Add four zero edge to matrix a (m by m), increase
%the dimension of matrix a to m+2 by m+2.
SizeB = SizeA+2;
for i=1:SizeB(1)
   for j=1:SizeB(2)
      b_temp(i,j)=1;
   end
end
%Define a new matrix b_temp (m+2 by m+2), which is
%matrix a with four zero edges.
for i= 2:(SizeB(1)-1)
   for j=2:(SizeB(2)-1)
      b_temp(i,j)=a(i-1,j-1);
   end
end
%We notice that in Matlab, when the pixel is higher,
%the color is lighter. Hence after read image to
%matrix a, the white pixel is 1 and black pixel is
%0. Now we define another matrix b in which the
%white pixel is 0 and black pixel is 1.
for i=1:SizeB(1)
```

```
   for j=1:SizeB(2)
      if b_temp(i,j)==0
         b(i,j)=1;
      else b(i,j)=0;
      end
   end
end
for i=1:SizeB(1)
   for j=1:SizeB(2)
    bb(i,j)=20;
   end
end
%Now we using the max filter to do page segmenation.
%Here, the threshold is 5, choosing the maximum from
%five points (p_1,...,p_5).
for i=2:SizeB(1)-1
   for j=2:SizeB(2)-1
        p(1)=b(i-1,j);
        p(2)=b(i,j-1);
        p(3)=b(i,j);
        p(4)=b(i,j+1);
        p(5)=b(i+1,j);
        max=0;
```

```
      for k=1:5
          if max<p(k)
             max=p(k);
          end
        end
        if max==0
           %If point(i,j) is white pixel,
           %we draw it in lighter color
           bb(i,j)=20;
        else
           %If point(i,j) is black pixel,
           %we draw it in darker color.
           bb(i,j)=10;
        end
   end
end
%Remove the four edges of the matrix bb, we get
%m by m matrix bbb which is the geometric structure
%of document image.
for i= 1:SizeA(1)
   for j= 1:SizeA(2)
      bbb(i,j)=bb(i+1,j+1);
   end
```

```
end


%==============OUTPUT OF EXPERIMENT 1=============%

%Convert matrix bbb to unsigned 8-bit integer matrix.

B=uint8(bbb);

%Draw image B in two color.

image(B)


%------------------------------------------------%

%  Experiment 2: Using max filter with threshold 9.%

%------------------------------------------------%

%Add four zero edge to matrix a (m by m), increase

%the dimension of matrix a to m+2 by m+2.

SizeC = SizeA+2;

for i=1:SizeC(1)

   for j=1:SizeC(2)

      c_temp(i,j)=1;

   end

end

%Define a new matrix c_temp (m+2 by m+2), which is

%matrix a with four zero edges.

for i= 2:(SizeC(1)-1)

 for j=2:(SizeC(2)-1)
```

```
  c_temp(i,j)=a(i-1,j-1);
 end
end
%Now we define another matrix c in which the
%white pixel is 0 and black pixel is 1.
for i=1:SizeC(1)
   for j=1:SizeC(2)
      if c_temp(i,j)==0
         c(i,j)=1;
      else c(i,j)=0;
      end
   end
end
for i=1:SizeC(1)
   for j=1:SizeC(2)
    cc(i,j)=20;
   end
end
%Now we using the max filter to do page segmenation.
%Here, the threshold is 9, choosing the maximum from
%five points (p_1,...,p_9).
for i=2:SizeC(1)-1
   for j=2:SizeC(2)-1
```

```
  p(1)=c(i-1,j);
  p(2)=c(i,j-1);
  p(3)=c(i,j);
  p(4)=c(i,j+1);
  p(5)=c(i+1,j);
  p(6)=c(i-1,j-1);
  p(7)=c(i-1,j+1);
  p(8)=c(i+1,j-1);
  p(9)=c(i+1,j+1);
  max=0;
for k=1:9
    if max<p(k);
       max=p(k);
    end
  end
  if max==0
     %If point(i,j) is white pixel,
     %we draw it in lighter color
     cc(i,j)=20;
  else
     %If point(i,j) is black pixel,
     %we draw it in darker color.
     cc(i,j)=10;
```

```
        end
    end
end
%Remove the four edges of the matrix cc, we get
%m by m matrix ccc which is the geometric structure
%of document image.
for i= 1:SizeA(1)
    for j= 1:SizeA(2)
        ccc(i,j)=cc(i+1,j+1);
    end
end


%===============OUTPUT OF EXPERIMENT 2==============%
%Convert matrix ccc to unsigned 8-bit integer matrix.
C=uint8(ccc);
%Draw image C in two color.
image(C)


%-------------------------------------------------%
%  Experiment 3: Using max filter with threshold 13.%
%-------------------------------------------------%
%Add four zero edge to matrix a (m by m), increase
%the dimension of matrix a to m+2 by m+2.
```

```
SizeD = SizeA+4;

for i=1:SizeD(1)

   for j=1:SizeD(2)

      d_temp(i,j)=1;

   end

end

%Define a new matrix d_temp (m+2 by m+2), which is

%matrix a with four zero edges.

for i= 3:(SizeD(1)-2)

 for j=3:(SizeD(2)-2)

  d_temp(i,j)=a(i-2,j-2);

 end

end

%Now we define another matrix d in which the

%white pixel is 0 and black pixel is 1.

for i=1:SizeD(1)

   for j=1:SizeD(2)

      if d_temp(i,j)==0

         d(i,j)=1;

      else d(i,j)=0;

      end

   end

end
```

```
for i=1:SizeD(1)

   for j=1:SizeD(2)

    dd(i,j)=20;

   end

end

%Now we using the max filter to do page segmenation.

%Here, the threshold is 13, choosing the maximum from

%five points (p_1,...,p_13).

for i=3:SizeD(1)-2

   for j=3:SizeD(2)-2

         p(1)=d(i-1,j);

         p(2)=d(i,j-1);

         p(3)=d(i,j);

         p(4)=d(i,j+1);

         p(5)=d(i+1,j);

         p(6)=d(i-1,j-1);

         p(7)=d(i-1,j+1);

         p(8)=d(i+1,j-1);

         p(9)=d(i+1,j+1);

         p(10)=d(i-2,j);

         p(11)=d(i,j-2);

         p(12)=d(i+2,j);

         p(13)=d(i,j-2);
```

```
      max=0;
     for k=1:13
        if max<p(k);
           max=p(k);
        end
      end
      if max==0
         %If point(i,j) is white pixel,
         %we draw it in lighter color
         dd(i,j)=20;
      else
         %If point(i,j) is black pixel,
         %we draw it in darker color.
         dd(i,j)=10;
      end
   end
end
%Remove the four edges of the matrix dd, we get
%m by m matrix ddd which is the geometric structure
%of document image.
for i= 1:SizeA(1)
   for j= 1:SizeA(2)
      ddd(i,j)=dd(i+2,j+2);
```

```
   end
end

%==============OUTPUT OF EXPERIMENT 3==============%
%Convert matrix ddd to unsigned 8-bit integer matrix.
D=uint8(ddd);
%Draw image D in two color.
image(D)

%--------------------------------------------------%
%  Experiment 4: Using max filter with threshold 25.%
%--------------------------------------------------%
%Add four zero edge to matrix a (m by m), increase
%the dimension of matrix a to m+2 by m+2.
SizeE = SizeA+4;
for i=1:SizeE(1)
   for j=1:SizeE(2)
      e_temp(i,j)=1;
   end
end
%Define a new matrix e_temp (m+2 by m+2), which is
%matrix a with four zero edges.
for i= 3:(SizeE(1)-2)
```

```
 for j=3:(SizeE(2)-2)
  e_temp(i,j)=a(i-2,j-2);
 end
end
%Now we define another matrix e in which the
%white pixel is 0 and black pixel is 1.
for i=1:SizeD(1)
   for j=1:SizeD(2)
      if e_temp(i,j)==0
         e(i,j)=1;
      else e(i,j)=0;
      end
   end
end
for i=1:SizeE(1)
   for j=1:SizeE(2)
    ee(i,j)=20;
   end
end
%Now we using the max filter to do page segmenation.
%Here, the threshold is 25, choosing the maximum from
%five points (p_1,...,p_25).
for i=3:SizeE(1)-2
```

```
for j=3:SizeE(2)-2

    p(1)=e(i-1,j);

    p(2)=e(i,j-1);

    p(3)=e(i,j);

    p(4)=e(i,j+1);

    p(5)=e(i+1,j);

    p(6)=e(i-1,j-1);

    p(7)=e(i-1,j+1);

    p(8)=e(i+1,j-1);

    p(9)=e(i+1,j+1);

    p(10)=e(i-2,j);

    p(11)=e(i,j-2);

    p(12)=e(i+2,j);

    p(13)=e(i,j-2);

    p(14)=e(i-2,j-2);

    p(15)=e(i-2,j-1);

    p(16)=e(i-1,j-2);

    p(17)=e(i+1,j-2);

    p(18)=e(i+2,j-2);

    p(19)=e(i+2,j-1);

    p(20)=e(i-2,j+1);

    p(21)=e(i-2,j-2);

    p(22)=e(i-1,j-2);
```

```
        p(23)=e(i+1,j-2);

        p(24)=e(i+2,j+1);

        p(25)=e(i+2,j-2);

        max=0;

      for k=1:25

          if max<p(k);

             max=p(k);

          end

        end

        if max==0

           %If point(i,j) is white pixel,

           %we draw it in lighter color

           ee(i,j)=20;

        else

           %If point(i,j) is black pixel,

           %we draw it in darker color.

           ee(i,j)=10;

        end

   end

end

%Remove the four edges of the matrix ee, we get

%m by m matrix eee which is the geometric structure

%of document image.
```

```
for i= 1:SizeA(1)
   for j= 1:SizeA(2)
      eee(i,j)=ee(i+2,j+2);
   end
end


%==============OUTPUT OF EXPERIMENT 4=============%
%Convert matrix ddd to unsigned 8-bit integer matrix.
E=uint8(eee);
%Draw image E in two color.
image(E)
```

B.2 Simulation 3: the MedOSF and the MaxOSF approaches with threshold 5, 9, 13, 25.

```
%================================================%
%      Implementation of the Algorithm           %
%              Li  Ma  (9745151)                 %
%              December 2000                     %
%================================================%
% MedOSF Algorithm: A new approach for page      %
% segmentation which is based on the med order   %
```

```
% statistic filter (MedOSF).                      %
%                                                 %
% INPUT:    A page of document image(Doc02.bmp).  %
% OUTPUT:   Geometric structure of the document.  %
% COMPLIER: Using Matlab under the Windows        %
%           enviornment.                          %
%=================================================%

%================KEY WORD=========================%
%     THRESHOLD n: is the filter windows length   %
%     which is the no. of the connected neighbour %
%     hood plus 1.                                %
%=================================================%
clear all;
%====================INPUT STARTS=================%
%Read image from a graphic file (BMP file) into A
%whose class is unsigned 8-bit integer (uint8).
A=imread('Doc02','bmp');
%===============THE ALGORITHM STATS===============%
%Double convert A to double precision.
a=double(A);
%Find the length of vector a.
SizeA = size(a);
```

```
%---------------------------------------------------------%
% Experiment 1: Using med and max filter with threshold 5.%
%---------------------------------------------------------%
%Add four zero edge to matrix a (m by m), increase
%the dimension of matrix a to m+2 by m+2.
SizeB = SizeA+2;
for i=1:SizeB(1)
   for j=1:SizeB(2)
      b_temp(i,j)=1;
   end
end
%Define a new matrix b_temp (m+2 by m+2), which is
%matrix a with four zero edges.
for i= 2:(SizeB(1)-1)
   for j=2:(SizeB(2)-1)
      b_temp(i,j)=a(i-1,j-1);
   end
end
%We notice that in Matlab, when the pixel is higher,
%the color is lighter. Hence after read image to
%matrix a, the white pixel is 1 and black pixel is
%0. Now we define another matrix b in which the
%white pixel is 0 and black pixel is 1.
```

```
for i=1:SizeB(1)
   for j=1:SizeB(2)
      if b_temp(i,j)==0
         b(i,j)=1;
      else b(i,j)=0;
      end
   end
end
for i=1:SizeB(1)
   for j=1:SizeB(2)
    bb(i,j)=20;
   end
end
%Now we using the med filter to do page segmenation.
%Here, the threshold is 5, choosing the median from
%five points (p_1,...,p_5).
for i=2:SizeB(1)-1
   for j=2:SizeB(2)-1
        p(1)=b(i-1,j);
        p(2)=b(i,j-1);
        p(3)=b(i,j);
        p(4)=b(i,j+1);
        p(5)=b(i+1,j);
```

```
        med=0;
        med=median(p);
        if med==20
           %If point(i,j) is white pixel,
           %we draw it in lighter color
           bb(i,j)=20;
        else
           %If point(i,j) is black pixel,
           %we draw it in darker color.
           bb(i,j)=10;
        end
   end
end
%Now we using the max filter to do page segmenation.
%Here, the threshold is 5, choosing the maximum from
%five points (p_1,...,p_5).
for i=2:SizeB(1)-1
   for j=2:SizeB(2)-1
        p(1)=bb(i-1,j);
        p(2)=bb(i,j-1);
        p(3)=bb(i,j);
        p(4)=bb(i,j+1);
        p(5)=bb(i+1,j);
```

```
      max=0;
     for k=1:5
       if max<p(k)
          max=p(k);
       end
     end
     if max==0
       %If point(i,j) is white pixel,
       %we draw it in lighter color
       bbb(i,j)=20;
     else
       %If point(i,j) is black pixel,
       %we draw it in darker color.
       bbb(i,j)=10;
     end
   end
end
%Remove the four edges of the matrix bbb, we get
%m by m matrix bbbb which is the geometric structure
%of document image.
for i= 1:SizeA(1)
   for j= 1:SizeA(2)
      bbbb(i,j)=bbb(i+1,j+1);
```

```
   end
end
%==============OUTPUT OF EXPERIMENT 1=============%
%Convert matrix bbb to unsigned 8-bit integer matrix.
B=uint8(bbbb);
%Draw image B in two color.
image(B)
%-------------------------------------------------------%
%  Experiment 2: Using med and max filter with threshold 9.%
%-------------------------------------------------------%
%Add four zero edge to matrix a (m by m), increase
%the dimension of matrix a to m+2 by m+2.
SizeC = SizeA+2;
for i=1:SizeC(1)
   for j=1:SizeC(2)
      c_temp(i,j)=1;
   end
end
%Define a new matrix c_temp (m+2 by m+2), which is
%matrix a with four zero edges.
for i= 2:(SizeC(1)-1)
 for j=2:(SizeC(2)-1)
  c_temp(i,j)=a(i-1,j-1);
```

```
 end
end
%Now we define another matrix c in which the
%white pixel is 0 and black pixel is 1.
for i=1:SizeC(1)
   for j=1:SizeC(2)
      if c_temp(i,j)==0
         c(i,j)=1;
      else c(i,j)=0;
      end
   end
end
for i=1:SizeC(1)
   for j=1:SizeC(2)
    cc(i,j)=20;
   end
end
%Now we using the med filter to do page segmenation.
%Here, the threshold is 5, choosing the median from
%five points (p_1,...,p_9).
for i=2:SizeC(1)-1
   for j=2:SizeC(2)-1
        p(1)=c(i-1,j);
```

```
        p(2)=c(i,j-1);
        p(3)=c(i,j);
        p(4)=c(i,j+1);
        p(5)=c(i+1,j);
        p(6)=c(i-1,j-1);
        p(7)=c(i-1,j+1);
        p(8)=c(i+1,j-1);
        p(9)=c(i+1,j+1);
        med=0;
        med=median(p);
        if med==20
           %If point(i,j) is white pixel,
           %we draw it in lighter color
           cc(i,j)=20;
        else
           %If point(i,j) is black pixel,
           %we draw it in darker color.
           cc(i,j)=10;
        end
   end
end
%Now we using the max filter to do page segmenation.
%Here, the threshold is 9, choosing the maximum from
```

```
%five points (p_1,...,p_9).
for i=2:SizeC(1)-1
   for j=2:SizeC(2)-1
        p(1)=cc(i-1,j);
        p(2)=cc(i,j-1);
        p(3)=cc(i,j);
        p(4)=cc(i,j+1);
        p(5)=cc(i+1,j);
        p(6)=cc(i-1,j-1);
        p(7)=cc(i-1,j+1);
        p(8)=cc(i+1,j-1);
        p(9)=cc(i+1,j+1);
        max=0;
      for k=1:9
          if max<p(k);
             max=p(k);
          end
        end
        if max==0
           %If point(i,j) is white pixel,
           %we draw it in lighter color
           ccc(i,j)=20;
        else
```

```
            %If point(i,j) is black pixel,
            %we draw it in darker color.
            ccc(i,j)=10;
         end
    end
end
%Remove the four edges of the matrix ccc, we get
%m by m matrix cccc which is the geometric structure
%of document image.
for i= 1:SizeA(1)
   for j= 1:SizeA(2)
      cccc(i,j)=ccc(i+1,j+1);
   end
end
%===============OUTPUT OF EXPERIMENT 2==============%
%Convert matrix ccc to unsigned 8-bit integer matrix.
C=uint8(cccc);
%Draw image C in two color.
image(C)
%----------------------------------------------------------%
%  Experiment 3: Using med and max filter with threshold 13.%
%----------------------------------------------------------%
%Add four zero edge to matrix a (m by m), increase
```

```
%the dimension of matrix a to m+2 by m+2.

SizeD = SizeA+4;

for i=1:SizeD(1)

   for j=1:SizeD(2)

      d_temp(i,j)=1;

   end

end

%Define a new matrix d_temp (m+2 by m+2), which is

%matrix a with four zero edges.

for i= 3:(SizeD(1)-2)

 for j=3:(SizeD(2)-2)

  d_temp(i,j)=a(i-2,j-2);

 end

end

%Now we define another matrix d in which the

%white pixel is 0 and black pixel is 1.

for i=1:SizeD(1)

   for j=1:SizeD(2)

      if d_temp(i,j)==0

         d(i,j)=1;

      else d(i,j)=0;

      end

   end
```

```
end

for i=1:SizeD(1)

   for j=1:SizeD(2)

    dd(i,j)=20;

   end

end

%Now we using the max filter to do page segmenation.

%Here, the threshold is 13, choosing the maximum from

%five points (p_1,...,p_13).

for i=3:SizeD(1)-2

   for j=3:SizeD(2)-2

        p(1)=d(i-1,j);

        p(2)=d(i,j-1);

        p(3)=d(i,j);

        p(4)=d(i,j+1);

        p(5)=d(i+1,j);

        p(6)=d(i-1,j-1);

        p(7)=d(i-1,j+1);

        p(8)=d(i+1,j-1);

        p(9)=d(i+1,j+1);

        p(10)=d(i-2,j);

        p(11)=d(i,j-2);

        p(12)=d(i+2,j);
```

```
        p(13)=d(i,j-2);
        med=0;
        med=median(p);
        if med==20
           %If point(i,j) is white pixel,
           %we draw it in lighter color
           dd(i,j)=20;
        else
           %If point(i,j) is black pixel,
           %we draw it in darker color.
           dd(i,j)=10;
        end
   end
end
%Now we using the max filter to do page segmenation.
%Here, the threshold is 13, choosing the maximum from
%five points (p_1,...,p_13).
for i=3:SizeD(1)-2
   for j=3:SizeD(2)-2
        p(1)=dd(i-1,j);
        p(2)=dd(i,j-1);
        p(3)=dd(i,j);
        p(4)=dd(i,j+1);
```

```
  p(5)=dd(i+1,j);
  p(6)=dd(i-1,j-1);
  p(7)=dd(i-1,j+1);
  p(8)=dd(i+1,j-1);
  p(9)=dd(i+1,j+1);
  p(10)=dd(i-2,j);
  p(11)=dd(i,j-2);
  p(12)=dd(i+2,j);
  p(13)=dd(i,j-2);
  max=0;
for k=1:13
    if max<p(k);
       max=p(k);
    end
  end
  if max==0
     %If point(i,j) is white pixel,
     %we draw it in lighter color
     ddd(i,j)=20;
  else
     %If point(i,j) is black pixel,
     %we draw it in darker color.
     ddd(i,j)=10;
```

```
        end
    end
end
%Remove the four edges of the matrix ddd, we get
%m by m matrix dddd which is the geometric structure
%of document image.
for i= 1:SizeA(1)
   for j= 1:SizeA(2)
      dddd(i,j)=ddd(i+2,j+2);
   end
end
%==============OUTPUT OF EXPERIMENT 3=============%
%Convert matrix ddd to unsigned 8-bit integer matrix.
D=uint8(dddd);
%Draw image D in two color.
image(D)
%-----------------------------------------------------------%
% Experiment 4: Using med and max filter with threshold 25.%
%-----------------------------------------------------------%
%Add four zero edge to matrix a (m by m), increase
%the dimension of matrix a to m+2 by m+2.
SizeE = SizeA+4;
for i=1:SizeE(1)
```

```
  for j=1:SizeE(2)
     e_temp(i,j)=1;
  end
end
%Define a new matrix e_temp (m+2 by m+2), which is
%matrix a with four zero edges.
for i= 3:(SizeE(1)-2)
 for j=3:(SizeE(2)-2)
  e_temp(i,j)=a(i-2,j-2);
 end
end
%Now we define another matrix e in which the
%white pixel is 0 and black pixel is 1.
for i=1:SizeD(1)
   for j=1:SizeD(2)
      if e_temp(i,j)==0
         e(i,j)=1;
      else e(i,j)=0;
      end
   end
end
for i=1:SizeE(1)
   for j=1:SizeE(2)
```

```
    ee(i,j)=20;
   end
end
%Now we using the max filter to do page segmenation.
%Here, the threshold is 25, choosing the maximum from
%five points (p_1,...,p_25).
for i=3:SizeE(1)-2
   for j=3:SizeE(2)-2
         p(1)=e(i-1,j);
         p(2)=e(i,j-1);
         p(3)=e(i,j);
         p(4)=e(i,j+1);
         p(5)=e(i+1,j);
         p(6)=e(i-1,j-1);
         p(7)=e(i-1,j+1);
         p(8)=e(i+1,j-1);
         p(9)=e(i+1,j+1);
         p(10)=e(i-2,j);
         p(11)=e(i,j-2);
         p(12)=e(i+2,j);
         p(13)=e(i,j-2);
         p(14)=e(i-2,j-2);
         p(15)=e(i-2,j-1);
```

```
        p(16)=e(i-1,j-2);
        p(17)=e(i+1,j-2);
        p(18)=e(i+2,j-2);
        p(19)=e(i+2,j-1);
        p(20)=e(i-2,j+1);
        p(21)=e(i-2,j-2);
        p(22)=e(i-1,j-2);
        p(23)=e(i+1,j-2);
        p(24)=e(i+2,j+1);
        p(25)=e(i+2,j-2);
        med=0;
        med=median(p);
        if med==20
           %If point(i,j) is white pixel,
           %we draw it in lighter color
           ee(i,j)=20;
        else
           %If point(i,j) is black pixel,
           %we draw it in darker color.
           ee(i,j)=10;
        end
    end
end
```

```
%Now we using the max filter to do page segmenation.
%Here, the threshold is 25, choosing the maximum from
%five points (p_1,...,p_25).
for i=3:SizeE(1)-2
   for j=3:SizeE(2)-2
        p(1)=ee(i-1,j);
        p(2)=ee(i,j-1);
        p(3)=ee(i,j);
        p(4)=ee(i,j+1);
        p(5)=ee(i+1,j);
        p(6)=ee(i-1,j-1);
        p(7)=ee(i-1,j+1);
        p(8)=ee(i+1,j-1);
        p(9)=ee(i+1,j+1);
        p(10)=ee(i-2,j);
        p(11)=ee(i,j-2);
        p(12)=ee(i+2,j);
        p(13)=ee(i,j-2);
        p(14)=ee(i-2,j-2);
        p(15)=ee(i-2,j-1);
        p(16)=ee(i-1,j-2);
        p(17)=ee(i+1,j-2);
        p(18)=ee(i+2,j-2);
```

```
    p(19)=ee(i+2,j-1);
    p(20)=ee(i-2,j+1);
    p(21)=ee(i-2,j-2);
    p(22)=ee(i-1,j-2);
    p(23)=ee(i+1,j-2);
    p(24)=ee(i+2,j+1);
    p(25)=ee(i+2,j-2);
    max=0;
  for k=1:25
      if max<p(k);
         max=p(k);
      end
    end
    if max==0
       %If point(i,j) is white pixel,
       %we draw it in lighter color
       eee(i,j)=20;
    else
       %If point(i,j) is black pixel,
       %we draw it in darker color.
       eee(i,j)=10;
    end
end
```

```
end

%Remove the four edges of the matrix eee, we get

%m by m matrix eeee which is the geometric structure

%of document image.

for i= 1:SizeA(1)

   for j= 1:SizeA(2)

      eeee(i,j)=eee(i+2,j+2);

   end

end

%===============OUTPUT OF EXPERIMENT 4==============%

%Convert matrix ddd to unsigned 8-bit integer matrix.

E=uint8(eeee);

%Draw image E in two color.

image(E)
```