1993

Optical Character Recognition Using Morphological Attributes.

Gili Mendel
Louisiana State University and Agricultural & Mechanical College

Order Number 9401552

Optical character recognition using morphological attributes

Mendel, Gili, Ph.D.

The Louisiana State University and Agricultural and Mechanical Col., 1993

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

OPTICAL CHARACTER RECOGNITION USING MORPHOLOGICAL ATTRIBUTES

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for degree of
Doctor of Philosophy

in

The Department of Computer Science

by
Gili Mendel
B.S., Nicholls State University, 1989
May 1993

Acknowledgments

I would like to convey my sincere gratitude to my dissertation advisor, Professor Sitharama Iyengar for enthusiastically supporting my work at the Robotics Research Laboratory at Louisiana State University. His devotion to research will always be an inspiration to my future research. I will always be indebted for his generosity in enlightening me to the challenge of research and academic Computer Science.

Much appreciation is extended toward my advisory committee Professors Donald Kraft, Bush Jones, Tyler John, Alexander Skavantzos, and Peter Wolenski, whose encouragement, intellectual support, invaluable advice, insight, and patience have made my research work possible.

It has been an enlightening experience, both scientific and personal, to collaborate and learn from Dr. Sukhamay Kundu. His problem solving approach and his thorough knowledge in artificial intelligence are truly inspiring. In fact, the approach taken in this research was largely influenced by his incremental work in artificial intelligence and neural networks.

I thank Professor Iyengar and the computer science department at Louisiana State University for continually supporting me as a teaching assistant and providing exceptional research facilities, both at the Louisiana State University Robotics Research Laboratory and the Physics department. I also thank Dr. Jacob Barhen, for the opportunity to conduct research at NASA's Jet Propulsion Laboratory, Mrs. Judy Mackenzie, for her support at the test architecture area at IBM, and Professor Bert

Boyce Bert for his continued encouragement in the School of Library and Information Science.

Lastly, but most important, I would like to express my gratitude to my parents, Abraham and Helena Mendel, for the love and attention they have given to me in my life which have been my inspiration throughout. Of course, I cannot forget my brothers, Udi and Tuvi, and my fiancee, Michelle Richoux, for their continued patience, wisdom and enthusiastic support, without which I could have never reached this stage in my life.

Table of Contents

List of Tables

List of Figures

Abstract

This dissertation addresses a fundamental computational strategy in image processing hand written English characters using traditional parallel computers. Image acquisition and processing is becoming a thriving industry because of the frequent availability of fax machines, video digitizers, flat-bed scanners, hand scanners, color scanners, and other image input devices that are now accessible to everyone. Optical Character Recognition (OCR) research increased as the technology for a robust OCR system became realistic.

There is no commercial effective recognition system that is able to translate raw digital images of hand written text into pure ASCII. The reason is that a digital image comprises of a vast number of pixels. The traditional approach of processing the huge collection of pixel information is quite slow and cumbersome. In this dissertation we developed an approach and theory for a fast robust OCR system for images of hand written characters using morphological attribute features that are expected by the alphabet character set. By extracting specific morphological attributes from the scanned image, the dynamic OCR system is able to generalize and approximate similar images. This generalization is achieved with the usage of fuzzy logic and neural network.

Since the main requirement for a commercially effective OCR is a fast and a high recognition rate system, the approach taken in this research is to shift the recognition computation into the system's architecture and its learning phase. The recognition process constituted mainly simple integer computation, a preferred computation on digital computers. In essence, the system maintains the attribute envelope boundary upon which each English character could fall under. This boundary

is based on extreme attributes extracted from images introduced to the system beforehand.

The theory was implemented both on a SIMD-MC^2 and a SISD machines. The resultant system proved to be a fast robust dynamic system, given that a suitable learning had taken place. The principle contributions of this dissertation are:

- Improving existing thinning algorithms for image preprocessing.
- Development of an on-line cluster partitioning procedure for region oriented segmentation.
- Expansion of a fuzzy knowledge base theory to maintain morphological attributes on digital computers.
- Dynamic Fuzzy learning/recognition technique.

1. Introduction

At the onset of the seventies, the state of the art for optical character recognition (OCR) technology and applications had progressed to a point where reasons for using OCR became readily apparent[70]. Over the years as OCR technology evolved, the manufacturer of optical scanning equipment and systems consistently concentrated their efforts on providing system solutions which enable users to rescue and control data entry costs while simultaneously improving information quality. Image acquisition and processing is becoming a thriving industry because of the frequent availability of fax machines, video digitizers, flat-bed scanners, hand scanners, color scanners, and other image input devices that are now accessible by everyone. Optical Character Recognition research increased as the technology for a robust OCR system became realistic. The term 'morphological' in this dissertation refers to the fact that the methods we will use and the algorithms we will generate are based on *mathematical morphology,* which is a set-theoretic approach to image processing. The hallmark of the morphological approach is to *manage* the selective loss of information, so that the desired objects can be extracted[76]. This enables relatively fast analysis; an advantageous approach for image processing since image patterns are comprised of an overwhelming amount of information. This technique is being employed by the human vision system as well. Although humans receive vast amount of information from their eye sensors, not all information is processed. For example, if one looks at a bare wall, in general, one does not examine the wall reflection rigorously 'pixel by pixel' in order to come to the conclusion that the wall is clear. If, however, there would be a dark spot on the wall, one focuses his attention on that dark spot, although it constitutes only a small portion of the wall. It seems that humans focus their vision-processing time on sub-images which are perceived

significant at the moment. Understanding the manner upon which the human's vision system functions could solve a fundamental problem: generating semantics to sensored image patterns.

Although some OCR systems have been developed thus far, their usage requires the use of a specialized notepad upon which one has to write on. These systems, then, record the essence of the lettering construction. This information is then used at that time to interpret and translate the hand written text into pure ASCII characters. The requirement of a specialized notepad, however, significantly restricts the usage of these systems to specific applications. The additional information gathered by the notepad, however, reduces the complexity of the recognition problem significantly. This is because the system does not have to examine the raw digital image pixel by pixel. A digital image is often noisy and distorted. Also, inconsequential pixels (white pixels) which constituted more than 70% of the digital image are not processed, a significant advantage. In addition, the stroke information of the character construction is extracted relatively fast compared to the time it would take to obtain this information from a raw digital image.

1.1. Preliminaries

The definitions and common terminology of image processing and optical character recognition are presented here, since their usage is employed throughout the dissertation:

- **Pixel**

  An image is represented within a digital computer as a sequence of tiny dots called pixels. The color/gray shade of a pixel is recorded as an integer magnitude usually between 0-256 depending on the application and digitizer used. In optical character recognition, however, the reference to a pixel is to a magnitude of zero or one, white or black respectively. The spatial location of a pixel P is often denoted by its offset from the top left corner of the binary image. $P_{i,j}$, then, denotes the magnitude of the pixel on the i^{th} row and the j^{th} column.

- **Binary Image**

  A digital image where the aggregate pixel's gray shade is either zero or one (black or white) is referred to as binary image.

- **Skeletonization**

  Reducing the binary image (composed of white/black pixels) into a graph such that no pixel could be removed without partitioning the graph or merging two background localities is called skeletonization and is obtained by a thinning algorithm. (see section 2.2).

- **8-neighbor consulting**

  A system which examines all eight neighbors (North, North East, East, ..., West, North West) of a pixel P to arrive at a conclusion.

- **4-neighbor consulting**

  A system which examines only four neighbors of a pixel P: North, East, South, and West.

- **Edge pixel**

  A skeleton's pixel $P_{i,j}$ is considered an edge-pixel if it has only one neighbor with 8-neighbor consulting ($P_{i-1,j}$, $P_{i-1,j+1}$, $P_{i,j+1}$, $P_{i+1,j+1}$, ... , $P_{i-1,j-1}$).

- **DPI**

  The resolution of a scanned digital image is often measured by the number of dots (pixels) per inch.

- **Contour extraction**

  Extracting a sequence of spatial coordinates of pixels which constitute the digital pattern's contour referred to as contour extraction phase.

- **Direction vector features**

  By dividing the rectangular frame enclosing the normalized contour into 4x4 rectangular zones and obtaining the local histogram of the chain codes in each zone, one can construct direction vectors with possible directions of 0^o,45^o, 90^o, and 135^o. Thus, the feature vector has 64 components when all the 16 zones (4x4 localities) are included.

- **Profile features**

  In a structural classifier all of the features associated with the characters are derived from profiles of their external contours. These are character widths, ratios, location of extremes, and discontinuities in character profiles.

- **Horizontal/Vertical Symmetry**

  A character is considered to be Horizontal/Vertical symmetric, if the distribution of pixels is approximately the same on the left and right as well as top and bottom.

- **Cross Junction**

  The spatial location in which two digital strokes cross is considered to be a junction coordinate. If the pattern was reduced to a skeleton, a pixel $P_{i,j}$ is referred to as a junction pixel if $P_{i,j}$ has three or more direct neighbors (with 8-neighbor consulting).

- **Convolution**

  The convolution operation of two 2-dimensional functions f(x,y) and g(x,y) is defined as the following:

$$f(x,y)*g(x,y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}f(\alpha,\beta)g(x-\alpha,y-\beta)d\alpha d\beta \qquad (1.1)$$

where α and β are dummy variables of integration.

Since the digital images are a discrete collection of pixels, the convolution is formulated by letting f(x,y) and g(x,y) be discrete arrays (with some period M and N) in the x and y directions, respectively. Thus, the two-dimensional convolution function is defined as following:

$$f(x,y)*g(x,y)=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}f(m,n)g(x-m,y-n) \qquad (1.2)$$

The convolution function is generally used in image processing to apply a certain function whose discrete pre-computation is stored in an array (usually referred as a mask) onto the discrete gray shades function of the digital image. Convolution is used often, since the discrete function to be used is pre computed, and its utilization involved simple summation and multiplication operation. Also, since the convolution operation maintains no serial dependencies, it could be efficiently executed on a SIMD machines.

1.2. Background

There is no commercially effective recognition system capable of translating a raw digital images of hand written alphanumeric text into pure ASCII. The reason is that a digital image consists of a vast number of pixels. The traditional approach of processing the huge collection of pixel's information with von Neuman architecture is quite slow and cumbersome. An OCR system, however, would be commercially effective if it could improve or compete with manual translation by a skilled typist. A common approach for representing an image's structural shape is to reduce it into a graph[34]. This graph is obtained by approximating the shape's skeleton via a thinning algorithm. The skeleton is an important preprocessing step for many image analysis operations such as document processing[13], fingerprint recognition[58], and biomedical systems[25]. By reducing a digital image into a unitary skeleton, one reduces the number of pixels to be consulted during the recognition phase. Iyengar and I have proposed efficient, robust parallel methods for approximating a skeleton to a given digital pattern. These algorithms where implemented on a MasPar computer, a SIMD-MC^2 machine. An overview of these algorithms is rendered in sections 2.2.3.1 and 2.2.3.2.

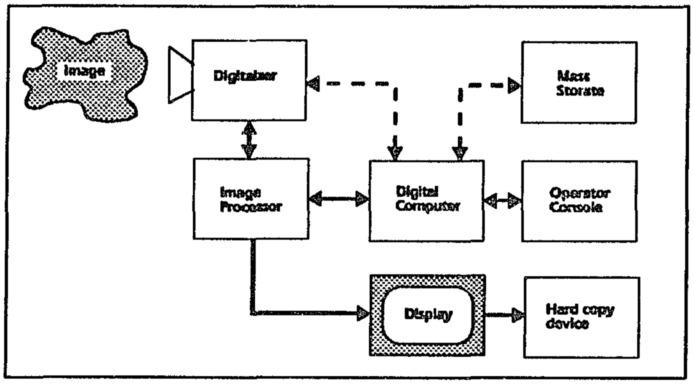


Fig. 1.1 Elements of an image processing system from [34].

Many methods have thus far been proposed to translate scanned documents. Still, most techniques employ rigorous computational methods, and this leads to cumbersome and slow translation procedures. These systems are based on the following hardware components for a general-purpose digital image processing system as suggested by Rafael[34] and is illustrated in Fig. 1.1. A digitizer converts the image's reflection into digitized representation. An image processor consists of hardware modules responsible for image acquisition, low-level processing (noise reduction and such), storage, and display. The digital computer is responsible for generating the semantics pertinent to the image acquired by the image processor. This system also includes a Mass storage (the extent of a raw 200 DPI image of a letter sized paper is approximately 1,000,000 bytes). Likewise the system includes a high resolution printer/display devices and an operator console. Many feature-oriented detection techniques were proposed to recognize digital patterns of hand/printed characters. Blesser et al[14] have argued that the failures of these feature-detection techniques lie in the ad hoc nature by which the features were chosen. Thus to build a machine that will recognize characters as accurately as humans, specific knowledge about human classification for characters must first be obtained.

This dissertation employs Blesser's guidelines to achieve a recognition system of the English hand/printed character set, independently of their size and font shape. The OCR system developed in this dissertation was implemented using the elements of a

digital image processing system, similar to the system suggested by Refael. It seems that the human vision system, in general, uses the same general idea. Still, the major difference between the human vision system and the elements of a computerized image processing system is the internal structure and operation schema of both the image processor (compared to the human eye) and the digital computer (compared to the human brain). In essence, the traditional von Neuman architecture employed by our common digital computers introduces inefficiencies in dealing with vast numbers of pixel magnitude and their combined image semantic. This problem could be overcome with specialized digital computers designed specifically for image acquisition and processing. The approach taken in this research is to shift the recognition computation into the system's architecture and its learning phase. That is, the design of the system is based on characteristics of the expected English character's digital pattern. The recognition is then automated by extracting unique attributes from the digital image (which requires a preprocessing phase) and does not require thorough computation.

Extracting specialized information by preprocessing the raw sensored image is employed by the human's vision system as well. The human's vision sensor, the eye, serves more then just the mapping of light reflection information into mosaic data to be processed by the brain. The eye performs extensive preprocessing of information which is transmited to the brain, that is, specific information regarding the image composition rather than a simple digital representation of the image's reflection on the eye's retina.

1.2.1. Structure of the Human Eye

The human's vision system employs the eye as a preprocessor to the image that is reflected on the retina. The eye sensor submits conceptual attributes to the brain, rather than a mosaic representation of the reflected image. Fig. 1.2 is a cross section diagram which depicts the anatomy structure of the human eye, as shown by Kent and Graff [49].

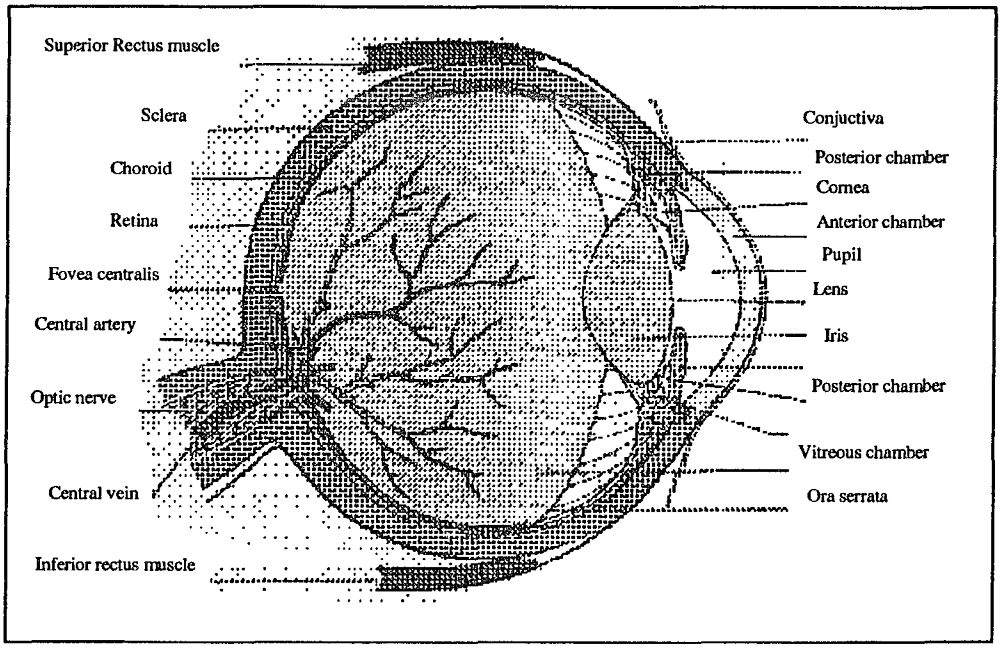


Fig. 1.2 The internal anatomy of the eye ball from[49].

The eye receptors are sensitive to a small portion of the electromagnetic spectrum. Still, within this range, the eye is able to distinguish an immense number of different wavelengths of frequencies that are interpreted by the brain as different colors. When light rays strike the cornea, they are bent (refracted) in much the same way as light rays striking optical lenses [28]. Actually, the principal difference between the lens of the eye and an ordinary optical lens is that the former is flexible. Additional bending of the light rays occurs as they pass from the cornea to the anterior cavity and through the lens. The lens carries out the focusing adjustments required to ensure that the images

are sharply focused on the retina. The existence of two eye sensors enable humans to have binocular vision and depth perception.

Although the eyes view portions of the external world that overlap considerably with one another, they do not form exactly identical images of an object because they occupy slightly different locations[28]. The nervous layer of the retina is composed of three principal layers of neurons: rod and cone cells, bipolar neurons, and ganglion neurons that are connected to the fibers of the optic nerve as illustrated in Fig. 1.3[49]. There are over 100 million rods in each eye. They are positioned on the peripheral parts of the retina and respond to dim light for black and white vision. In addition, they are sensitive to movement.

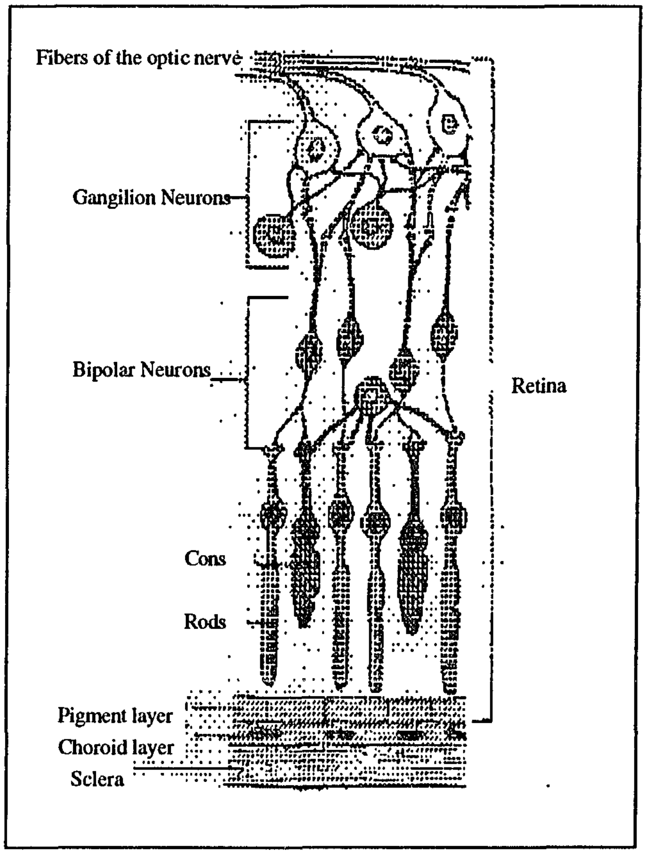


Fig. 1.3 The layers of the retina from [49].

Still, they provide poor visual acuity. In contrast, there are about seven million cones per eye. They provide daylight color vision and are responsible for visual acuity. They are concentrated near the center of the retina called the fovea centralis, which is the area of keenest vision[49]. Thus, although our peripheral view is quite large, we can only concentrate our detail vision on a small portion of a vista. Still, the information gathered by the rods is important, since an object movement or a drastic gradient change of color will attract our attention to focus this information on the fovea centralis. At that time, a more specific information is passed to the bipolar neuron layer. The acuteness of the information that is generated by the cones is further accented by the neural pathways to the bipolar neurons and their connection to ganglion neurons compared to the pathways from the rods. In the peripheral regions of the retina, many rods synapse with a single bipolar cell and many bipolar cells synapse with a single ganglion cell. In contrast, within the fovea there are approximately equal numbers of cones, bipolar cells, and ganglion cells, and relatively little convergence occurs[28]. Thus, the reflection of light onto the retina is fed via the rods and cones as an input to a neural network which have two main levels, the Bipolar neurons and the gangion neurons. The output of this neural net is then passed via the fibers of the optic nerve to the brain. The eye is then preprocesses the "raw light reflection" into frequency of nerve impulses. This mapping function is quite complex and by no means a one to one correspondence to the electromagnetic frequencies sensed by the cones and rods. The receptive field of a cell is the area within the retina that influence the activity of that cell as a result of light exposure. Actually, it is the photoreceptor area. However, this input may not be a direct reflection of light on the retina. It could be provided indirectly by other cells. Moreover, this input, fed by other cells, could be either inhibitory or stimulatory. This type of connection and reaction is similar to the way the human brain processes information as well. Because of such occurrences in addition to convergence and lateral interactions, the processing of visual signals begins

in the retina. The ganglion cells do not transmit to the brain a simple mosaic pattern of the image as it was reflected on the retina by the lens[49].

1.2.2. Visual Signals in the Human Brain

The human brain is enclosed by the skull and meninges and is bathed in cerebrospinal fluid. It has a tremendous metabolic rate and is susceptible to oxygen deprivation and certain toxins[49], indeed a *super computing system.* The two cerebral hemispheres carry out different functions. Generally, analytical and verbal skills are controlled by the left hemisphere and spatial and artistic intelligence are the source of the right hemisphere[28]. The vision system is located in the back bottom portion of the brain. The vision information received by the brain is fed by the optic nerve which consists of axons of aggregated ganglion neurons that emerge through the posterior aspect of the eyeball. The two optic nerves (from each eye ball) converge at the optic chiasma (see Fig. 1.4 from Mason[49]), located just anterior to the pituitary gland.

All fibers which arises form the medial half of each retina cross at that point to the opposite side. The fibers of the optic nerve that arise from the lateral half of the retina do not cross each other. From the optic chiasma, the axons continue as the optic tracts. Thus, each track consists of ganglion cell axons from the lateral half of the original retina and the medial half of the opposite retina. At this point, some of the axons within the optic tracts travel to the mid brain nuclei (the pretectal nuclei) that is responsible for the pupillary light reflex. These signals ultimately lead to an increased stimulation of the circular smooth muscles of the iris by neurons of the parasympathetic division of the autonomic nervous system. Most of the ganglion cell axons within the optic tracks are connected to the lateral geniculate bodies of the thalamus where they

synapse with neurons that form pathways called optic radiations. These connect to the visual cortex of the occipital lobes.

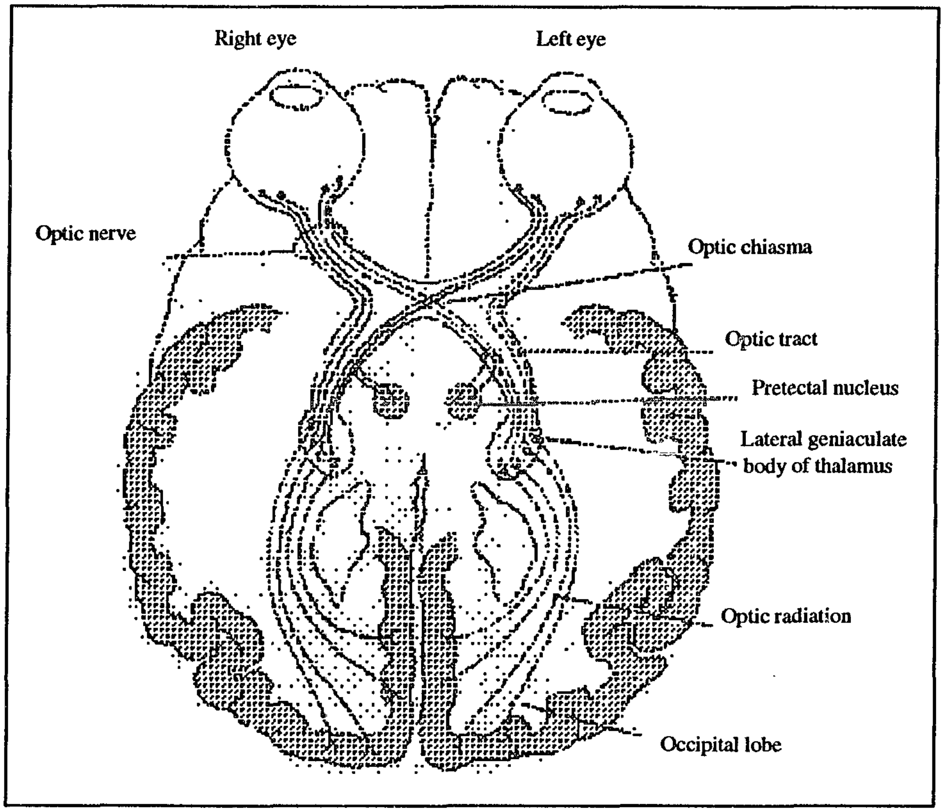


Fig. 1.4 Ventral view of the human brain from [49].

The hierarchy of cells within the visual cortex vary widely in organization. Although some cortical cells have circular symmetrical receptive fields whose responses resemble those of ganglion cells and geniculate neurons, the receptive fields of other cortical cells are organized so that cells respond best to a specific oriented line segments (such as narrow slits of light) rather than spots of light. That is, cells called simple cell cells, respond only when the line is oriented at a precise angle in a particular portion of a cell's receptive field[49]. This information is in response to lines and borders. Other cells, called complex cells, respond when the line is oriented at a precise angle regardless of its position in a cell's receptive field. This enable the recognition of the movement of a visual stimulus. Again, Mason maintain that the

components of the vision pathway transmit coded messages about verity aspects of the image reflected on the retina rather than a mosaic pattern of the image[49]. These messages convey information of contrast, movement, and color in such fashion that the brain utilizes it to develop a visual representation of the examined image.

1.3. Virtues of an OCR System

Computer vision systems are currently being introduced in various environments in order to provide machines with the capability to "see," the intelligence to "understand" the surrounding environment, and the ability to identify various objects[17]. Image processing/recognition systems have proved to be quite effective in the verification of circuitry boards, element categorization and classification, or object measurements. The manipulation of visible images is just one of the many uses of image processing. Digitizers could produce digital patterns to images reflected from or with the combination of invisible light. These systems are already in use for space exploration and various medical applications.

Document processing is becoming a thriving industry as scanners and sizable storage media are accessible by everyone. In many industries, individual records are scanned and stored on-line for later reference. This enterprise enables an easy real time access to records by many people without the need to be close to the archive or the storage area were the original documents are usually stored. Although microfilm serve some of the above purposes, the advantages of an on-line document retrieval is notable. The maintenance of the records that are on-line are done only at the system upon which the records are kept. Access is given to authorized personnel whether directly, a local area network, or *internet* type connection. Still, the advantages of these on-line

document retrieval system could be increased tremendously if used in conjunction of an optical recognition system as well.

A given robust OCR system (that is, one able to translate hand written text into ASCII) should be able to process images of typed text as well. Entering typed text documents into any given word processor will become a trivial fast procedure. The frequent use of this OCR system is guaranteed to be an integral part of an office's daily routine. Memos, or any other material relevant to the proper operation of the business could be put on-line to be referenced by all.

1.3.1. Information Retrieval

Scanning document images to be stored for later retrieval is quite a simple task. Still, the stored images have no meaningful semantics beside to the referencing schema that is assigned to it. This referencing schema is used to accomplish actual retrieval of the image at a later time. The image itself is treated as no more then a sequence of binary pixels (white or black). The main function of this image system is to enable the display of those pixels at the correct spatial location in order to produce the correct image of the original document. If, however, one augments the referencing schema of the image by the addition of an ASCII translation of the image's text (by an optical character recognition system), the retrieval of the binary image is then reduced to a regular search of a sub-string within the scanned images. Therefore, the retrieval of a document does not require one to remember any specific storage naming convention. Rather, one will be able to search for a particular document's image by its content as well.

Many businesses make their customers and/or employees fill out specialized information forms. Most often, this information is entered into a computer by simply keying in the recorded data. This handling procedure could be replace by the usage of an optical character recognition system. The filled forms could be passed through a page scanner which in turn feeds the digital image to an OCR module. This module, equipped with the knowledge of the form's format, extracts the proper information into the appropriate database. A copy of the digital image could be retained as well. This scanning based procedure could speed up the processing of the information tremendously. An insurance policy maker is able to scan a new insurance application form to be transmitted directly to the base company (where ever this company might be). The base company, in turn, is able to make a real time decision on whether or not to grant the customer the insurance policy he sought.

1.3.2. Compression and Storage

The scanning of images is performed, in general, either for later retrieval or perhaps to further process the digital information onto meaningful insight pertaining the scanned image. The higher the resolution of the scanner, the better the quality of the digital image. Higher resolution entails easier recognition of important features within the digital image. The reproduction of the original image will be unblemished as the resolution increases. The resolution of the reconstructed image will depend, however, on the resolution of the reconstruction device (such as printer and display). Although higher resolution preempts better quality, it also denotes more pixels to store and process. A letter size paper scanned with 200 DPI resolution (200 DPI is considered below medium resolution quality) will result in 3,740,000 pixels for storage and

processing. Thus, employing image processing as part of a daily business operation will necessitate sizable storage media to store the various documents which are to be processed by the office.

Many compression techniques are available for image processing and computer science applications in general. Still, an 80% compression ratio of a document is considered to be exceptionally good. If, however, the document is composed of mainly alphanumeric information, translating the digital image into pure ASCII will reduce a letter size document into about 2,700 alphanumeric characters. A compression of about 99.92%. OCR, then, could serve more than information retrieval goals. It could significantly reduce storage space needs, if the original digital image is not required to reside on-line.

1.3.3. Transmission and Duplication

Telecommunication technology has improved significantly in the past few years. The needs for better communication are fueling a vibrant telecommunication industry. Satellite communication is now used routinely for television and telephone information-transmission. Beepers, cellular phones, and fax machines are accessible by everyone. Still, the price of transmitting information is quite steep. Fax machines, although operating at 9,600 baud (bits per second), transfer facile images through regular phone lines. The resolution of the facile images is kept significantly low. The reason is that low resolution denotes fewer pixels and thus less data to transmit. Frequent use of long distance phone calls by a fax machine could be quite expensive.

To resolve the low resolution images employed by fax machines, an OCR technology could be used. This system could translate the alphanumeric portions of the

document into pure ASCII while retaining some attributes regarding the original image, such as font and size. The receiving fax machine could reconstruct the document with much improved quality. Moreover, by reducing the information of the facsimile image into a sequence of characters and their attributes, the system actually compresses the amount of information to be transferred significantly. This characteristic would be a significant improvment for a fax machine that depends on phone lines as transferal media.

The combination of fax machines and OCR technology is beginning to become commercially available. WinFax Pro 3.0 is a PC software which uses OCR technology in conjunction with a spell checker that can automatically convert faxes into an editable format such as ASCII or RTF (Rich Text Format). The OCR component is potentially a big time-saver. David Andrews tested the beta version of WinFax Pro 3.0 for BYTE magazine. His testing indicated that the OCR performance of WinFax Pro exhibited a degree of recognition accuracy that varied from fax to fax and even from sentence to sentence within the same fax. Character recognition of a fax document is no picnic[22]. When the technology of an OCR becomes a robust system, its usage will have a major effect in various areas of document processing.

1.3.4. Languages Translation

The process of converting information from one language to another with a computer, MT (machine translation which comes under the natural language processing heading), is an increasingly important technology [60]. International economic and political stability and well-being are dependent on shared information. Information, however, may be represented in different formats and languages. The European

community has to overcome nine different official languages. Although most translations are still performed by people, computers are shouldering part of the burden and are expected to take over most of the load as MT technology improves. The IAMT (International Association for Machine Translation) and AMTA (Association for Machine Translation in the Americas) have formed in order to improve and promote MT technology.

MT technology is more than looking up words. The system analyzes the original language's text to be translated into sentences in the target language. The philosopher I. A. Rechards once wrote that translation is "probably the most complex type of event yet produced in the evolution of the cosmos." It is not a wonder, then, that the architecture of MT systems vary in seemingly infinite ways [60]. Perhaps within a decade MT will appear in several ways — translating telephones, multilingual E-mail, and machines that scan and translate letters and articles written in foreign languages[27]. Several technologies will make such scenario's possible: automated speech recognition, speech generation, optical character recognition, and machine translation[27]. Combining optical-scanning and OCR technology with personal computer-based MT, the illumina translates whatever text you place on its face plate[27]. It is clear then, that a robust fast OCR system could be used in order to scan and translate foreign literature and documents from one source language to a target language. As international communication and trade is accelerating, the combination of MT and OCR technologies is a valuable and important way to keep control on the increasing volume of information.

1.4. General System Approach

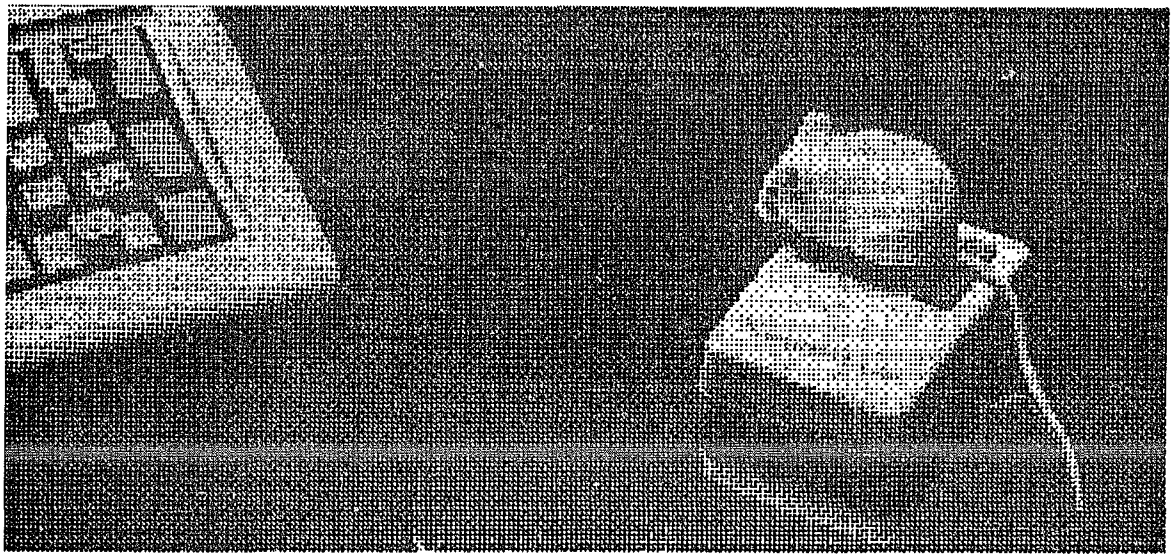


Fig. 1.5 A picture worth a thousand words.

It is a known phrase that *a picture is worth a thousand words.* This phrase gains significant effect within the image processing domain. The picture depicted in Fig. 1.5 is worth exactly 238,000 pixels (700x340). Extracting the semantics of the shade distribution with digital computers is not a trivial task. The picture is a reflection of three objects: a computer keyboard, a computer mouse, and a live *real* mouse. Still, most of the pixels in this picture illustrate the background and paint no significant information regarding any of the three objects. As humans, our vision system will not spend much processing time on the background distribution of pixel. It seems that we will use some degree of image processing and object recognition in addition to some logical inference analysis in order to recognize all three objects. The keyboard and the live mouse are likely to be recognized by using past exposure knowledge to these object. This past knowledge should be enough to recognize the unique feature characteristics of the keyboard and the mammal mouse. Recognizing the computer mouse, however, is probably a combination of both image and logical inference. If the

pixels representing the computer keyboard and live mouse were to be deleted, one will not be able to recognize the computer mouse object, at least with the degree of certainty that one is able to do so with the knowledge that the picture is composed of a computer keyboard and a live mouse. This knowledge in conjunction to the spatial distribution of pixel information which paint the computer mouse, make us conclude that since we have a computer keyboard (which bias our logical inference analysis to computer components); and since we recognized a live mouse, then the pixels representing that small square box (that could fit the description of a computer mouse) is most likely a computer mouse device. We see that a recognition system is more than just an ad hoc recognition of spatial distributions of pixel's shades. The combination of image recognition and logical inferences should enable one to arrive into object recognition where the object to be recognized could not be determined with certainty.

The key obstacle to image processing is the vast amount of pixels that are processed during a recognition phase. Thus, emphasis should be given into some preprocessing stages upon which irrelevant pixels are eliminated so that the recognition system is able to further consult a small sub set of the pixels which constitute the original image. Edges are curves in an image where rapid changes in brightness occur[8]. In a gray-scale image, they could be caused by changes in a surface where two objects overlap, or at a shadow. In a binary image, edges are the border where the object meets a background. To find the edges in an image, the derivative of the image pixel's shade function can be taken. At the areas of high discontinuity, such as edges, the value of the derivative will be relatively large, and small elsewhere. Because an image is a discrete distribution of pixels, a discrete approximation of the derivative is used. One such discrete approximation is the Sobel operator [16]. Taking a Sobel operation on an SIMD machine could be performed in few instruction cycles. The result, however, is a significant reduction to the number of pixels that are to be further processed. Running a Sobel operator the digital image depicted in Fig. 1.5, is

illustrated in Fig. 1.6. It seems, then, that the transformation of a raw digital image into some intermediate representation prior to the recognition course could be an advantageous and inexpensive approach. The human's vision system perform extensive preprocessing analysis within the eye sensor prior to the analysis performed by the brain (as described in section 1.2.1). Moreover, the eye sensor transform the light reflection to a set of indication and attributes pertaining lines, gradient color change and movements, rather than a set of pixel stimuli.

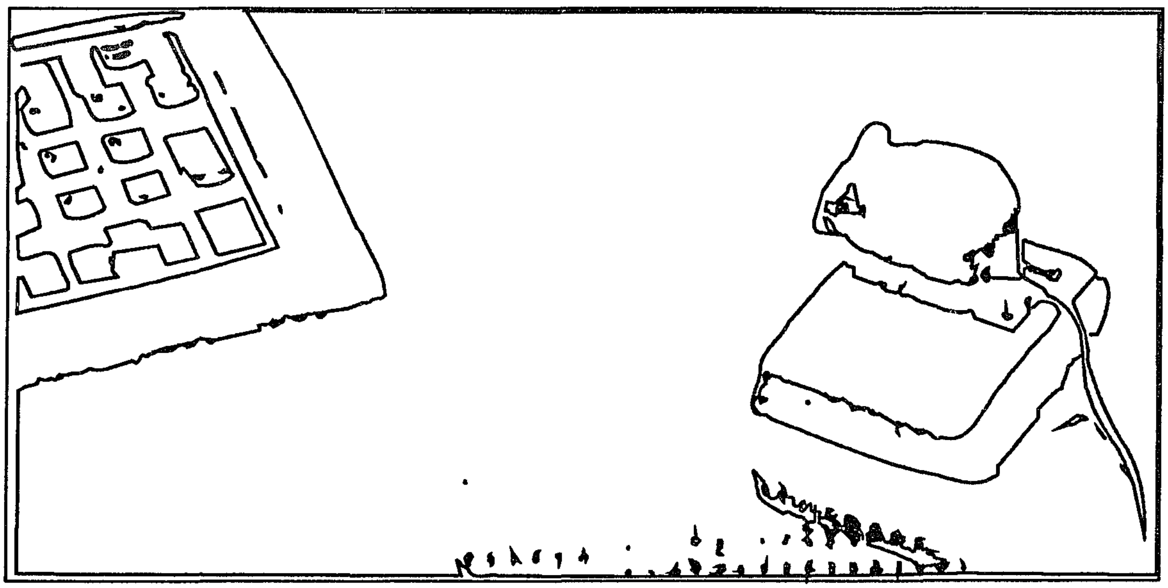


Fig. 1.6 Extracting edge pixels.

The specific objectives and goal of this research is to design an optical character recognition system using morphological attributes and parallelism for hand/printed English characters. The system should conform to the following requirements:

- *Low recognition and rejection error rates.*
- *High speed recognition for commercial applications.*
- *Dynamic learning.*

The complexity of a solution to a problem is greatly determined by the problem's representation. For example, if one is to determine if an integer is divisible by three, then the traditional approach is to first sum up the digits of this number and determine if that sum is devisable by three. That is, the integer 93 is divisible by three since 9+3 = 12. To determine if 12 is devisable by three, we continue to add the digits 1+2=3, and then we can conclude that since the integer 3 is divisible by three then the original integer, 93, is divisible by three as well. Augmenting the division by 3 rule set to one digit integers, one should repetitively subtract the value of three from the integer in question as long as the result is positive. If we have a remainder, then the original integer is not divisible. If we do not have a remainder, then the original integer is divisible by three. More specifically, the divisible by three function, *f*, is illustrated in equations (1.3-1.5).

$$f(d_1 d_2 d_3 \ldots d_n) = \begin{cases} \mathrm{f}(\sum \mathrm{d_i}) & \mathrm{n} > 1 \\ \mathrm{g(d)} & \text{eitherwise} \end{cases} \qquad (1.3)$$

$$g(d) = \begin{cases} g(d-3) & d > 2 \\ h(d) & eitherwise \end{cases} \qquad (1.4)$$

$$h(d) = \begin{cases} 3Divisible & d = 0 \\ Not3Divisible & eitherwise \end{cases} \qquad (1.5)$$

where $(d_1 d_2 d_3 \ldots d_n)$ is a decimal digit representation of the input integer.

The divisible by three function, *f*, is a robust solution. The time complexity of this function is O(n). Additional digits will prolong the time it will take this function to produce an answer. If, however, the processing environment is such that the divisible

by three problem is frequent, one should consider changing the representation schema of integer. That is, if one will work in base 3 rather than base 10, then the solution to the divisible by three problem become trivial and is shown in equation 1.6.

$$f(d_1 d_2 d_3 \dots d_n) = \begin{cases} 3Divisible & d_n = 0 \\ Not3Divisible & eitherwise \end{cases} \tag{1.6}$$

where $(d_1 d_2 d_3 \dots d_n)$ are the base 3 digit representation of an integer.

The guideline principle in developing the OCR system is to design both a recognition system as well as a schema upon which the image's information is stored for further efficient consulting and processing. The representation of the information and its processing manner is designed in close correlation to the expected data (spatial pixual representation of alphanumeric English characters) and its usage schema within the recognition phase. Still, representation alone will not solve the intricate time complexity when processing digital images. Emphasis should be given in designing an automated recognition system that does not require thorough computation and assessment of the scanned image. The design of the system's architecture should solve most of the computation needs. As an example, consider the common Coke machine's architecture. In order to acquire a soft drink from a Coke machine, one enters coins into the coin slot. At that time, each coin slides over a sequence of pre-determined circular openings, in which case, a coin will fall through the first hole that is large enough to accommodate the size of the coin. At this point, the coin will slide onto a spring mechanism with a pre-determined *k* constant to validate that the coin is actually authentic. If the coin is authentic, the coin will then continue to a money collection bin as it passes through a mechanical or electrical switch upon which a count is kept of the amount of money entered so far into the machine. If enough money was entered, and a particular button is pressed, then the appropriate soft drink's bin door will open to

release the appropriate soft drink. One clearly understands that if the soft drink bins were not filled properly, the machine will release a soft drink that is not in accordance to the customer's request. The machine was not designed to overcome these type of uncertainties. The beauty of the Coke machine is that the machine does not actually compute anything. The diameter of the coin openings were determined in the design phase. So was the *k* constant of the various springs. Once a coin is inserted into the machine, the responded retrogression of the machine is automated without the need of rigorous computation.

A traditional computer scientist or mathematician use computations to achieve a desired result. The resulted answer is decisive and is usually achieved by a precise computation. If the Coke machine's function was to be described as an algorithm, a traditional computer scientist that is not familiar with mechanical operation of that machine, will most likely produce an algorithm similar to the one of algorithm 1.1.

[1] Initialize Counter to 0
[2] Wait for a coin to be inserted into the machine
[3] Measure the coin's diameter
[4] If *not quarter diameter* goto [*9*]
[5] weigh the coin
[6] if *not proper weight* goto [*2*]
[7] Add 0.25 to counter
[8] goto [*100*]

•
•
•

[100] if Counter *is neither greater nor equal to* soft drink's price goto [*2*]
[101] weight for soft drink type bottom to be pressed
[102] if *Coke was pressed*, then open the Coke bin's door.

•
•
•

Algorithm 1.1 *Coke machine's operation.*

Although the execution of algorithm 1.1 will, in essence, generate an identical result to that of the mechanical Coke machine. Algorithm 1.1, however, is not an efficient algorithm. This algorithm processes each coin by actually measuring its diameter and weight, and computing the next action to be taken. If time is a major constraint, the mechanical Coke machine is far more superior since the actions taken by the machine are automated by its designed architecture. Neither computation of diameter nor computation of weight is needed to achieve the desired result.

Many methods have proposed thus far to translate scanned documents. Still, most techniques employ rigorous computational methods which lead to cumbersome, slow translation procedures. Statistical analysis of pixel distribution will lead to high accuracy recognition systems. Still, this system will not be commercially effective. Technology is available to reduce the character's digital image into a set of line segments. Still, the conventional schema of representing a line l by equation 1.7 will not be efficient to use on conventional computers.

$$y = ax + b \qquad (1.7)$$

where a is the line's slope and b is the intercept.

This is because floating point operations are significantly slower than integer computations. An emphasis is given to best utilize traditional computers, by representing image's information in a schema that is best processed by digital computers.

The optical character recognition designed in this dissertation is illustrated in the block diagram of Fig. 1.7. The information that is generated by the digitizer is first processed to reduce noise, isolate character images within the image, and employ some simple feature extraction from the character's image (such as thinning algorithm). Next, specific morphological attributes pertaining the character's image are gathered.

This information could be used to recognize the character, or to update the alphabet attributes knowledge base. This is to enable the system to learn attribute combination and their correlation to specific alphanumeric characters. The information stored on the mass storage device (such as the digital image, image's attributes, and the ASCII translation of the digital image) could be directed to the graphic display or printer (see Fig. 1.7).

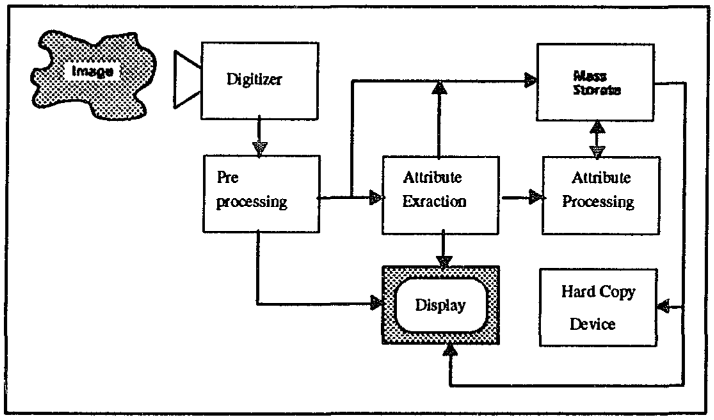


Fig. 1.7 Block diagram for an OCR system.

By virtue of eliminating improbable characters (given a digital image of an English figure) the recognition phase produce a probable ASCII character based on information upon which that system was exposed to before. The elimination process constituted mainly of simple integer computation, a preferred computation on digital computers. The image's information learned by the OCR system is a pre-computation of all possible morphological attributes and their alliance to each of the possible English characters. Therefore, the size of this attribute information on the mass storage is constant, regardless to the number of digital images that were learned by the system. In essence, the system maintains the attribute envelope boundary upon which an English figure could fall under. This boundary is based on image's extreme attributes extracted from images learned before.

1.5. Dissertation Outline

This dissertation is organized into five chapters. The chapter order is given in the progression in which the optical character recognition system processes the digital image. The first chapter has been devoted to present the context and framework under which this work was developed. This chapter has introduced some basic mathematical tools currently employed in image recognition and general digital processing. It also has presented fundamental concepts and guidelines followed in our development of this system. This section now presents a brief overview on the contents of the remaining chapters.

Chapter two introduces preprocessing actions which refine the raw digital image into a representation from which morphological attributes are somewhat desensitized to spatial deformations. We introduced two SIMD-MC^2 skeletonization algorithms which overcome deficiencies of existing algorithms.

Chapter three confronts the need to partition the digital image into a set of clusters each representing a single alphanumeric symbol. An on-line approach is developed using cluster Fusion and Fission techniques.

Chapter four present the background to fuzzy logic formalism used to develop the fuzzy knowledge base and fuzzy recognition process. It also furnishes the background of artificial neural networks and the difficulties they introduce. A description of a back propagation learning rule is given as used in our implementation which generalized the fuzzy recognition, achieved with the fuzzy knowledge base.

Chapter five provides a summary of contributions, concluding remarks and outlines directions along which this work can be extended.

1.6. Summary

This dissertation develops the theory and methodology upon which to process digital images of hand written English alphanumeric characters. The traditional approach of processing the huge collection of pixel's information with von Neuman architecture computers is quite slow and cumbersome. Many solutions to this problem have been introduced thus far. Still, an OCR system is commercially effective if it could improve or compete with manual translation of various documents by a skilled typist. Blesser, et al [14] have argued that the failures of traditional detection techniques lie in the ad hoc nature by which the image's features were chosen, and in order to build a machine which will recognize characters as accurately as humans, specific knowledge about human classification for characters must first be obtained. This dissertation employs Blesser's guidelines to achieve a recognition system of the English hand/printed character set independently to their size and font shape. Also, we reflect on the general schema upon which the ultimate vision system, the human's vision system, processes light that is reflected on the retina. The eye/brain combination is a true model of efficient methodology of assigning semantics to images who's discrete reflection is promptly analyzed.

The human eye is more than just a sensor to a small spectrum of electromagnetic frequencies. The eye transforms the light information into specific data. This data is not a simple mosaic representation of the light shades that was reflected on the retina through the lens. It represents specific features that compose the image. Although our peripheral view is quite large, we could only concentrate our detail vision analysis on a small portion of a vista[49]. The reflection of light onto the retina is fed via the rods and cones as an input to a neural network residing within the eye sensor. The two main levels of this neural network are the bipolar neurons, and the

gangion neurons. The output of this neural net is then passed via the fibers of the optic nerve to the brain. The eye then preprocesses the 'raw light reflection' into frequency of nerve impulses. This mapping function is quite complex and by no mean a one to one correspondence to the electromagnetic frequencies sensed by the cones and rods on the retina[49]. The receptive fields of cortical cells in the brain are organized so that cells respond best to a specific oriented line segments (such as narrow slits of light) rather than spots of light. That is, cells called simple cell cells, respond only when the line is oriented at a precise angle in a particular portion of a cell's receptive field[49]. This information is in response to lines and borders. Other cells, called complex cells, respond when the line is oriented at a precise angle regardless of its position in a cell's receptive field. This enable the recognition of the movement of a visual stimulus. Again, Mason maintain that the components of the vision pathway transmit coded messages about verity aspects of the image reflected on the retina rather than a mosaic pattern of the image[49]. We employ this schema, by first extracting specific features from the image of a characters. These features, referred by morphological attributes, have been chosen upon thorough consideration of the English character set contour pattern..

Since the implementation of this system was done on traditional SIMD-MC^2 and SISD machines, special consideration was given to design a system which represent the information regarding a particular image in a manner that will reduce the time complexity of processing the image during the recognition phase. Many methods were proposed thus far to translate scanned documents. Still, most techniques employ rigorous computational methods, which leads to cumbersome, slow translation procedures. Statistical analysis of pixel distribution, in general, will lead to high accuracy recognition system. Still, this system will not be commercially effective since the time complexity of statistical analysis of millions of pixels could be costly.

The recognition system achieves the identification of a character's image by way of eliminating improbable characters (given a digital image of an English figure). This phase produces a probable ASCII character based on information upon which that system was exposed to before. The elimination process constituted mainly simple integer computation, a preferred computation on digital computers. The image's information, learned by the OCR system, is a pre-computation of all possible morphological attributes and their alliance to each of the possible English characters. Therefore, the size of this attribute information on the mass storage is constant, regardless to the number of digital images that were learned by the system. In essence, the system maintains the attribute envelope boundary upon which an English figure could fall under. This boundary is based on image's extreme attributes extracted from images learned before.

2. Pre Processing

Whenever a picture is scanned, photocopied, transmitted, or displayed, the "quality" of the outputted picture may be lower than that of the inputted picture[7]. The objective of the pre-processing stage is to enhance the digital image so that the resulted pattern is more suitable than the original image to extract the morphological attributes needed for the optical character recognition process. This stage, in part, correlates to the processing that is performed by the Rods and Cons to the light that is reflected from the lens on to the retina of the human eye. Although most of this stage is executed on a digital computer, specialized hardware could greatly improve the overall speed of the enhancement procedure. Still, this stage could be easily implemented to be executed in parallel on a SIMD machine, MIMD machine, or a dedicated pipeline architecture.

The digital image is composed of aggregate of pixels referred to as the *spatial domain.* The functions that are employed in this stage on the spatial domain could be expressed as:

$$g(x,y) = T[f(x,y)] \tag{2.1}$$

where $f(x,y)$ is the input image gray shade value of pixels in the spatial domain at coordinates (x,y). The function $g(x,y)$ is the resulting gray shade of pixel $P_{(x,y)}$ processed by the operator function T that is defined over some neighborhood of (x,y). Large neighborhoods allow a variety of processing functions that go beyond image enhancement routines. The general approach is to let the values of f in a pre-defined neighborhood of (x,y) determine the value of $g(x,y)$[34].

2.1. Noise Reduction

Scanned images contain a level of noise, depending on the original quality of the scanned document and the resolution of the scanner employed. The noise blurs the image and confuses common vision operators. In binary images, noise takes the form of stray pixels; that is, white pixels that should be black, or black pixels that should be white. A common noise filtering routine is *median filtering.*

2.1.1. Median Filtering

Median filtering uses the values of the pixels contained in the pixel neighborhood to determine the new value given to the pixel of interest. However, it does not algorithmically calculate the new pixel's value from the pixels in the neighborhood. Instead, it sorts the pixels in the neighborhood into ascending order and picks the median gray shade value as the new value for the pixel of interest. The result of median filtering is that any random noise contained in an image will be effectively eliminated. Multiple applications of median filtering to an image can result in rather pleasing visual effects[16]. Median filtering, however, is not adequate with binary images. This is since the image's gray level function is an assortment of jumps between zero and one (black and white), rather than a discrete gradual transition of gray shades.

2.1.2. Convolution Method

Dealing with binary images (best used in OCR), the noise reduction general approach is to formulate the noise reduction function as a neighborhood mask (also referred to as templates, windows, or filters). In general, the mask is a two-dimensional array whose coefficients are chosen to detect a given property regarding $f(x,y)$. By convoluting the mask function on the digital image, one produces a refined image (see equation 1.2). A simple mask which detect isolated points that are different from a constant background is illustrated in Fig. 2.1.

x-2	*x-1*	*x*	*x+1*	*x+2*	
-1	-1	-1	-1	-1	*y-1*
-1	-2	-2	-2	-1	*y-2*
-1	-2	**32**	-2	-1	*y*
-1	-2	-2	-2	-1	*y+1*
-1	-1	-1	-1	-1	*y+2*

Fig. 2.1 Noise reduction mask, M.

The center of mask M, $M_{x,y}$ coordinate, is moved around the image. $T[f(x,y)]$ is then computed as illustrated in equation 2.2. This type of computation is under the convolutions processing methodology (see section 1.1), and its execution sits well under a SIMD-MC architecture. The neighborhood operation, defined by the convolution operation, could be easily implemented with neural networks, as well.

$$g(\mathrm{x,y}) = \begin{cases} 1 & 0 \le \sum_{i=-2}^{2}\sum_{j=-2}^{2} f(x+i,y+j)M(i,j) \le \delta \\ 1 & \sum_{i=-2}^{2}\sum_{j=-2}^{2} f(x+i,y+j)M(i,j) \le -\varphi \\ 0 & \textit{eitherwise} \end{cases} \qquad (2.2)$$

where φ and δ are pre-determined thresholds.

In processing a binary image, the summation of $\sum\sum f(\mathrm{x+i,y+j})M(\mathrm{i,j})$ is going to be positive only if $f(\mathrm{x,y}) = 1$. That is, $P_{\mathrm{x,y}}$ is on (dark color). Also, the summation will get smaller as additional neighborhood pixels get dark as well. It seems only logical that a region that constitutes a considerable number of dark pixels remain dark, where are a region that is combined with substantial collection of white pixels should remain white. On the other hand, if $f(\mathrm{x,y}) = 0$, then the summation is going to become negative. In this case, the summation will get smaller with additional dark neighborhood pixels. Again, if the summation is significantly negative, then $f(\mathrm{x,y})$ should most likely become dark as well.

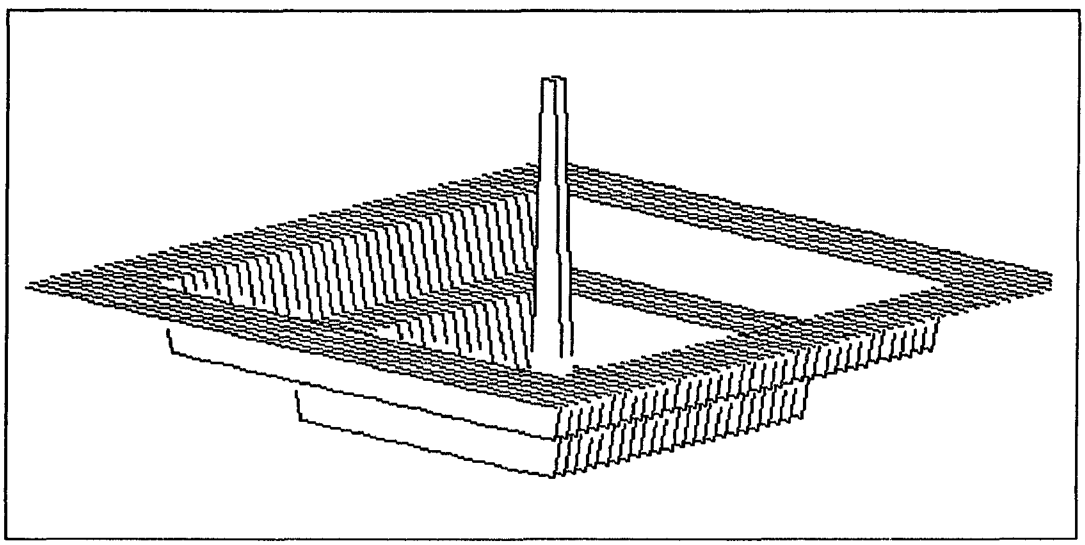


Fig. 2.2 Graphical description of mask M (defined in Fig. 2.1).

The noise reduction function represented by mask M and illustrated in Fig 2.2 could be modified by increasing the size of the consulted neighborhood, and employing a none linear function to dictate the correlation of $P_{x,y}$ and $P_{x+i,y+j}$.

If the scanner produces various gray shades rather than a binary image, one could easily convert the digital image into a binary image by employing a high pass filter. This filter accentuates the high frequency details of an image, while leaving the low frequency content intact. That is, the higher-frequency portions of the image will become brighter, while the lower-frequency portions become blacker. This filter will increase the image's sharpness, and also accentuate image's noise. This noise, then, will be more likely to be filtered with a median filter, or convolution filter analysis. The following 3x3 convolution mask functions are commonly employed in high pass spatial filtering:

-1	-1	-1
-1	9	-1
-1	-1	-1

0	-1	0
-1	5	-1
0	-1	0

1	-2	1
-2	5	-2
1	-2	1

Fig. 2.3 High-pass spatial filters

2.2. Thinning Algorithm

2.2.1. Introduction

A common approach for representing an image's structural shape is to reduce it into a graph[34]. This graph is obtained by approximating the shape's skeleton via a thinning algorithm. The skeleton is an important preprocessing step for many image analysis operations such as document processing[6], fingerprint recognition[58], and biomedical systems[25]. The skeleton must preserve the original image shape. No pixel can be removed from the skeleton without partitioning the skeleton or merging two background localities. This process should be fast to compute[72].

Many algorithms have been proposed so far for thinning digital patterns. These algorithms can be classified into sequential and parallel techniques. Although numerous serial approaches have been put forth by researchers [61,3,12,51,4,65,77], the importance of parallel procedures became important in recent years as the price of parallel SIMD machines dwindled which lead to their frequent availability. This paper presents two efficient parallel thinning algorithms. The main distinguishing feature of our algorithm over previous algorithms (Zhang, et al [78] and Holt, et al [39]) is in recognizing slanted 2-stroke contour-pixels and thus overcoming the degradation problem which produces, in some cases, skeletons that render no correlation to the original digital pattern. This make our algorithms a robust, fast method to approximate an image's shape skeleton.

2.2.2. Background

The concept of a parallel algorithm proposed by Zhang and Suen[78] is that each iteration, we eliminate the boundary of the image until all remaining pixels constitute the skeleton. Each iteration is composed of two sub-iterations. The first sub-iteration will remove the north-west boundary points, and the second sub-iteration will delete the south-east periphery points. The decision of removing a pixel is done by examining its 8-neighbors (see Fig. 2.4). A pixel will be removed in the first sub-iteration if it and its 8-neighbors satisfy the following four conditions:

(a) $2 \leq B(P) \leq 6$

(b) $A(P) = 1$

(c) $N*E*S = 0$

(d) $E*S*W = 0$

where,

A value of a pixel is '1' if it is *on* and '0' if it is *off.*
$B(P) = N + NE + E + SE + S + SW + W + NW$ and $A(P)$ is the number of '01' patterns in the circular sequence N,NE,E,...,W,NW,N) defined in Fig. 2.4.

In the second sub-iteration, only conditions (c) and (d) are changed as following:

(c') $N*E*W = 0$

(d') $N*S*W = 0$

		North		
	NW	N	NE	
West	W	*P*	E	*East*
	SW	S	SE	
		South		

Fig. 2.4 Pixel removal is determined by examining its 8-neighbors.

Lu and Wang[54] show that Zhang and Suen's algorithm tends to intensify noise and degrade some digital patterns into skeletons that do not render any correlation to their original images. They refine Zhang and Suen's algorithm by changing rule (a) in both sub-iterations to be:

(a*) $3 \leq \mathrm{B}(P) \leq 6$

```
.........................
......##########.........
.....############........
.....#############.......
.....####.....####.......
...............####......
...............####......
.......############......
......#############......
.....####.....#####......
....####.......####......
....###.........###......
....###.........###......
....####.......####......
....#####.....######.....
.....#################...
......#########.######...
........######...#####...
.........................
```

Fig. 2.5a Binary image of the character 'a.'

```
.........................
......^^^^^^^^^^.........
.....^^#########^^.......
.....^^^^^^^^^^^^#^^.....
.....^^^^.....^^#^.......
...............^#^^......
...............^#^^......
.......^##########^^.....
......^@@^^^^^^^^#^^.....
.....^@@^.....^^^#^......
....^@@^.......^^#^......
....^#^.........^#^......
....^#^.........^#^......
....^#^^.......^^#^......
....^^#^^.....^^^#^!.....
.....^^##^^^^#####^!!!...
......^^^####^^.^^!!!!...
........^^^^^^...^!!!!...
.........................
```

Fig. 2.5b Result may be degraded non-unitary skeleton.

In addition to the degradation problem, Zhang and Suen's algorithm may produce skeletons that are not unitary[2] (see Fig. 2.5). Lu and Wang's algorithm did not remedy this problem. A unitary skeleton is important if information such as junctions, edge points, and crossings are needed to be ascertained. This information is valuable for an OCR system. Abdulla, et al[1] upgrade Zhang and Suen's algorithm by the addition of two passes with a 3x4 and a 4x3 neighborhood windows to eliminate excess pixels, adding two iterations to the original algorithm.

Holt, et al[39] modifiy the above algorithms to eliminate both south-eastern and north-western boundary points at the same time. This bypasses the need for the two sub-iterations, reducing, in general, the number of overall iterations needed to approximate a skeleton. Holt et al claim that expanding the window for each element

to include edge information about its neighbors reduces the time required for thinning digital patterns. Although the number of iteration needed to thin a digital pattern has been reduced by Holt's algorithm, the time complexity of each iteration proves to induce algorithms that are consistently slower from that of Zhang's as will be shown in section 2.2.4.

Neither Zhang's algorithm nor Holt's algorithms guarantee the perseverance of the original digital pattern. To solve the degradation problem, we enhance the circumscription in which these algorithms determine if a pixel is a part of the skeleton. Thus, we present two thinning algorithms: Algorithms I and Algorithm II, are fast, robust parallel method for approximating a skeleton from a given digital pattern.

2.2.3. Improved Thinning Algorithms

2.2.3.1. Algorithm I

Zhang and Suen's provisions for removing contour pixels (described in section 2.2.2) will not guarantee the perseverance of the original digital pattern. A 2-stroke down slope, as shown in Fig. 2.6a, will be degraded into one pixel.

0	0	0	0	0
0	**1a**	**1b**	0	0
0	0	**1c**	**1d**	0
0	0	0	**1e**	0
0	0	0	0	0

Fig. 2.6a Down slope 2-stroke line

0	0	0	0	0
0	0	**1b**	**1a**	0
0	**1d**	**1c**	0	0
0	**1e**	0	0	0
0	0	0	0	0

Fig. 2.6b Up slope 2-stroke line.

Pixels *1a* and *1e* satisfied all the contour condition and are thus removed in the first sub-iteration. Pixels *1b* and *1d* will be removed in the second sub-iteration. The remaining pixel, pixel *1c*, is not an edge pixel and therefore it becomes the resulted skeleton — clearly a disfigured skeleton. This shortcoming is independent of the stroke's length. The scenario of pixels *1a*,*1e* and *1b*,*1d* will recur repeatedly, trimming the 2-stroke's edges consistently for each iteration. The same deficiency occurs in a 2-stroke up slope as depicted in Fig. 2.6b. In the same manner, pixel *1c* will be the resulted skeleton for this pattern as well. Although Lu and Wang[54] fix this imperfection, their algorithm fails to eliminate some contour pixels as $B(P) = 2$. This in turn results in a noise sensitive algorithm (see Fig. 2.9b). We refine Zhang's algorithm to eliminate the trimming cycle of a diagonal 2-stroke in algorithm I (see section 2.2.3.1) by removing a pixel in the first sub-iteration if it satisfies the following constraints :

(a) $2 \leq B(P) \leq 6$

(b) $A(P) = 1$

(c) $\neg E \vee \neg S \vee (\neg N \wedge \neg W)$

(d) $(B(P) = 2) \wedge (EE \vee WW)$ where EE,WW are illustrated in Fig. 2.7.

In the second sub-iteration, a pixel is removed if it satisfies :

(a') $2 \leq B(P) \leq 6$

(b') $A(P) = 1$

(c') $\neg N \vee \neg W \vee (\neg S \wedge \neg E)$

		NN		
	NW	N	NE	
WW	W	*P*	E	EE
	SW	S	SE	

Fig. 2.7 Pixel P and its neighbors.

Rule (d) of algorithm I involves two extra first-neighbor communications. Still, since this constraint is added solely to the first sub-iteration, on an average, one can consider it as the addition of one communication (were the first sub-iteration and the second sub-iteration are executed consecutively and each is counted as one iteration), a relatively small time increase. This addition guarantees the perseverance of a 2-stroke edges by ensuring that a pixel in which $B(P) = 2$, will be deleted in case it is not a part of a diagonal 2-stroke segment. In addition, one may add the seven-neighbor test to eliminate staircase occurrences as proposed by Holt and correct the non unitary skeleton problem as shown in Fig. 2.5b (see Appendix D for listings). The seven-neighbor test necessitate less communication needs than the 3x4 and 4x3 windows as suggested by Abdulla[2], and thus are faster to execute (see Table 2.2).

2.2.3.2. Algorithm II

Holt et al define an *edge* pixel to be as following:

$$edge(P) = \mathrm{P} \wedge t00 \wedge t11 \wedge \neg t01\mathrm{s} \qquad (2.3)$$

such that,

- P denotes the status if pixel P (true for dark and false for white).
- $t00$ indicates the existence of a '00' in the 8-neighbors sequences (NW,N,NE), (NE,E,SE), (SE,S,SW), (SW,W,NW).
- $t11$ indicates the existence of a '11' in the 8-neighbors sequences (NW,N,NE), (NE,E,SE), (SE,S,SW), (SW,W,NW).

- $\neg t01s$ is true if the run of '01' in the circular 8-neighbor sequence (NW, N, NE, E, SE, S, SW, W, NW) is less then two.

- NW, N, NE, E, SE, S, SW, W, NW are the 8-neighbor pixels shown in Fig. 2.3.

In each iteration Holt's algorithm eliminates the boundary of the image until all remained pixels constitute the skeleton. The survival of a pixel within an iteration is secured when pixel P maintains the following constraints :

$$P \wedge (\neg edge(P) \vee (edge(\mathrm{E}) \wedge \mathrm{N} \wedge \mathrm{S}) \vee (edge(\mathrm{S}) \wedge \mathrm{W} \wedge \mathrm{E}) \vee (edge(\mathrm{E}) \wedge edge(\mathrm{SE}) \wedge edge(\mathrm{S})) \qquad (2.4)$$

The guideline for the above logical criteria is that a pixel must be maintained if it is not an edge (not on the current boundary of the pattern); the pixel is a west edge of a vertical 2-stroke ($edge(\mathrm{E}) \wedge \mathrm{N} \wedge \mathrm{S}$); the pixel is a north edge of an horizontal 2-stroke ($edge(\mathrm{S}) \wedge \mathrm{W} \wedge \mathrm{E}$); the pixel is a north-west edge of a 2x2 square ($edge(\mathrm{E}) \wedge edge(\mathrm{SE}) \wedge edge(\mathrm{S})$). The algorithm biases the north-western edges of the pattern.

As in Zhang's algorithm, the provisions described above will not guarantee the perseverance of the original digital pattern. A 2-stroke down slope, as shown in Fig. 2.6a, will be degraded into one pixel. Pixel *1a* is an edge pixel. Since it is not on a west vertical 2-stroke, not on a north horizontal 2-stroke, and not a north-western edge of a 2x2 square pattern, it is eliminated. The same is true for pixel *1e*. The next iteration (consulting pixels *1b*, *1c*, and *1d*) will eliminate pixels *1b* and *1d*. The remaining pixel, *1c*, is not an edge pixel and therefore is the resulted skeleton — again a disfigured skeleton. Similar degradation will develop if one processes an up slope 2-

stroke line as shown in Fig. 2.6b. The resulted skeleton for this line is a pixel *1c* as well. The degradation problem occurs in both algorithms proposed by Holt et al.

To solve the degradation problem, we enhance the circumscription in which the algorithm will determine if a pixel is a part of the skeleton. The north-west edge of a 2x2 square (*edge*(E) $\wedge$ *edge*(SE) $\wedge$ *edge*(S)) condition is quite expensive since it involves three communications. Still, the north-west edge of a 2x2 square is not a common scenario. Therefore, we replace this condition to check for an edge pixel which satisfy the following conditions :

$$edge(P) \wedge \neg t111 \wedge \neg \mathrm{EE} \wedge \neg \mathrm{NN} \wedge \neg \mathrm{W} \qquad (2.5)$$

where,

$t111 =$ (NW $\wedge$ N $\wedge$ NE) $\vee$ (NE $\wedge$ E $\wedge$ SE) $\vee$ (SE $\wedge$ S $\wedge$ SW) $\vee$ (SW $\wedge$ W $\wedge$ NW)

and EE, NN and W are illustrated in Fig. 2.7. *(2.6)*

Clearly, this restraint super imposed the north-western edge of a 2x2 square. Moreover, it will guarantee that a 2-stroke diagonal pixel will be retained as well. In addition, since all processors have to fetch their 8-neighbor status (prerequisite of the *edge* function), the status of pixel NN should be available to the processor which processes pixel N. Thus, the processor which processes pixel *P* need to perform only one communication to find out the status of NN. The same is true for the status of EE (see listing Appendix D).

2.2.4. A Comparison of Performance

The algorithms have been implemented on a SIMD-MC^2 MasPar with 8,000 processing elements (see Appendix C). Algorithm I and Algorithm II were implemented both with staircase elimination and without (see listings Appendix D). The time measurement is the duration it took the algorithm to thin a digital pattern ten successive times in milliseconds. This time includes some initialization, the duration which is equivalent for all algorithms. Fig. 2.8 portrays extreme examples of the degradation process yielded by Zhang's and Holt's techniques as they were employed on an image of the letter '*x*.'

```
...............
...^......^....
...^^....^^....
....^^..^^.....
.....^^^^......
......#^.......
.....^^^^......
....^^..^^.....
...^^....^^....
...^......^....
...............
```

Fig 2.8a. Zhang.

```
...............
...^......^....
...^^....^^....
....^^..^^.....
.....^#^^......
......#^.......
.....^^^^......
....^^..^^.....
...^^....^^....
...^......^....
...............
```

Fig 2.8b. Holt I.

```
...............
...^......^....
...^^....^^....
....^^..^^.....
.....^^##......
......#^.......
.....^^##......
....^^..^^.....
...^^....^^....
...^......^....
...............
```

Fig 2.8c. Holt II

```
...............
...#......#....
...^#....#^....
....^#..#^.....
.....^##^......
......##.......
.....#^^#......
....#^..^#.....
...#^....^#....
...#......#....
...............
```

Fig 2.8d. Our I.

```
...............
...#......#....
...^#....#^....
....^#..#^.....
.....^##^......
......##.......
.....#^^#......
....#^..^#.....
...#^....^#....
...#......#....
...............
```

Fig 2.8e. Our II.

Table 2.1. emphasizes that the degradation increases the time it takes to thin an image. This is since additional iterations are executed when degradation is taking place. In general, one could see these consequences from Fig. 2.4 and Table 2.2.

Table 2.1 Processing the English letter 'X.'

Algorithm	**Stair Case Eliminatio n**	**Smoothing**	**Degraded Skeleton**	**Noisy Skeleton**	**Iterations**	**Time mSec**
Zhang	No	Yes	Yes	No	9	89
Holt I	Yes	No	Yes	No	10	160
Holt II	Yes	Yes	Yes	No	9	137
Our Algo I	Yes	Yes	No	No	3	31
Our Algo I	No	Yes	No	No	1	27
Our Algo II	Yes	No	No	No	3	55
Our Algo II	No	No	No	No	1	46

The elimination of the right bottom edge from the image of the letter 'a' in Fig. 2.5 could be critical for an OCR applications. Also, it took two extra iterations to thin the image (compared to our algorithm I with no stair case elimination) resulting in additional processing time. Lu and Wang[54] reduce the degradation problem still, their algorithm produce noisy skeletons as seen in from Fig. 2.9b.

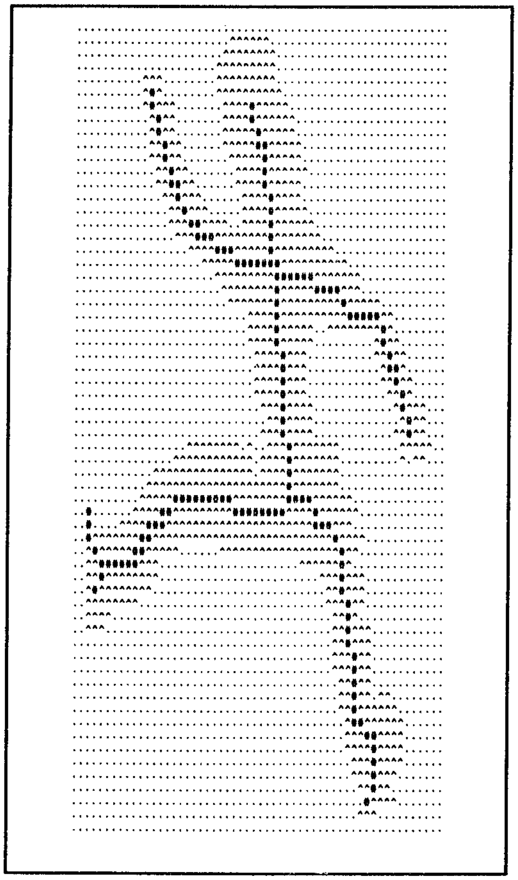


Fig. 2.9a Using Zhang's algorithm to thin the walking man pattern.

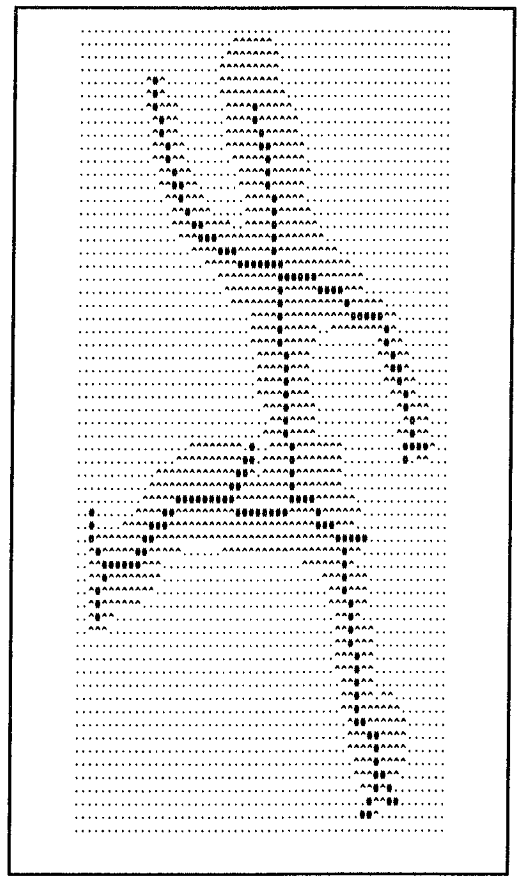


Fig. 2.9b Using LU's algorithm to thin the walking man pattern.

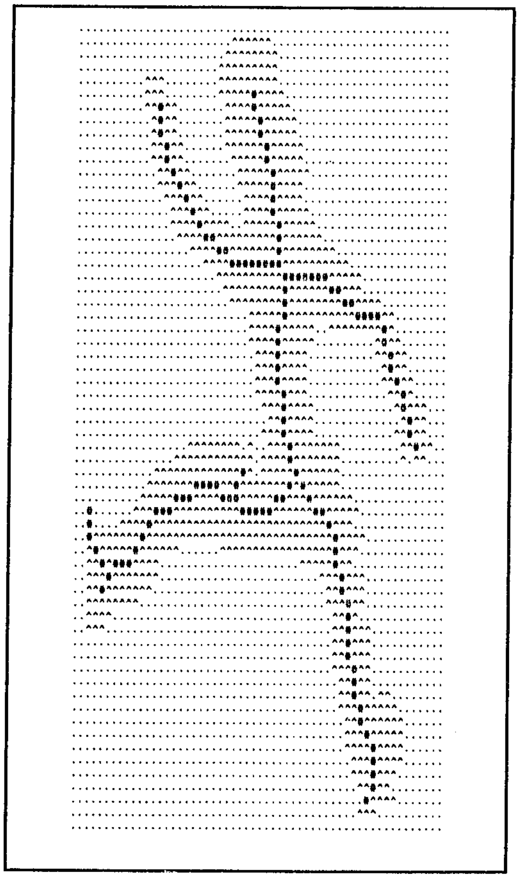


Fig. 2.9c Using Holt's I algorithm to thin the walking man pattern.

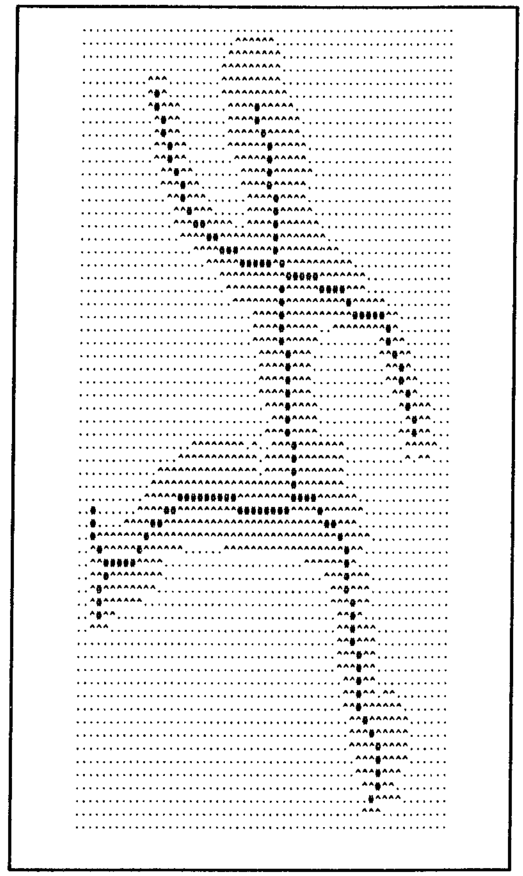


Fig. 2.9d Using Our Algorithm I to thin the walking man pattern.

The result of applying the modified algorithms to the walking man pattern used by Zhang and Suen (shown in Fig. 2.9a) and the English lower case 'a' pattern (shown in Fig. 2.10). It is evident from Table 2.2 that algorithms I, II are time comparable to their counterparts. These are typical results which were obtained to various digital patterns. Moreover, algorithms I and II constitute a fast robust method to thin digital

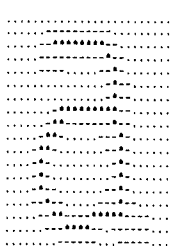


Fig.2.10a. Zhang's result on 'a.'

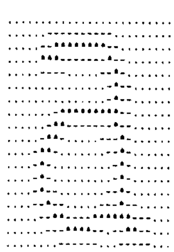


Fig. 2.10b Lu's result on 'a.'

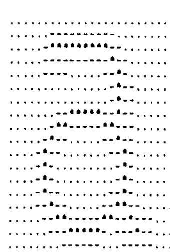


Fig.. 2.10c Holt I result on 'a.'

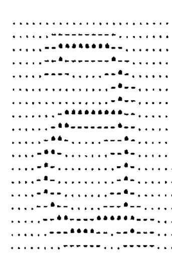


Fig. 2.10d Our I result on 'a.'

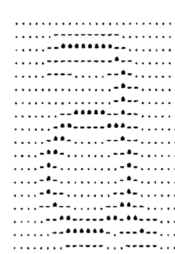


Fig. 2.10e Our II result on 'a.'

patterns. It is also noticeable that the stair case elimination proposed by Holt et al is faster then the 3x4 and 4x3 windows method suggested by Abdulla. Lu's algorithm seems to be quite fast. This, however, is the result of inadequate constraints which fail to remove all contour pixels — leading to fewer iterations.

Table 2.2 Algorithm statistics in thinning the walking man in Fig. 2.9, and the English character 'r'.

Algorithm	Stair Case Elimin.	Smoothing	Degraded Skeleton	Noisy Skeleton	Iterations *Walking Man*	Time mSec *Walking Man*	Iterations *English chr. 'a'*	Time mSec *English chr. 'a'*
Zhang	No	Yes	Yes	No	11	582	7	113
Abdulla	Yes	Yes	Yes	No	13	608	9	132
LU	No	No	No	Yes	11	575	5	101
Holt I	Yes	Yes	Yes	No	11	656	5	121
Holt II	Yes	No	Yes	No	11	668	5	124
Maha[34]	Yes	Yes	No	No	11	648	5	115
Our Algo I	No	Yes	No	No	11	580	5	101
Our Algo II	No	No	No	No	9	667	3	121
Our Algo I	Yes	Yes	No	No	13	605	7	113
Our Algo II	Yes	No	No	No	11	703	5	124

2.2.5. Conclusion

Parallel algorithms become important as the price of array-processor based SIMD computers declined in recent years, leading to their widespread availability. Thinning digital patterns is an important pre-processing phase in processing digital images. We have presented algorithms I and II algorithms, and have show them to be a fast robust skeletonizing methods. These algorithms show an improvement existing algorithms by eliminating the degradation problem. This has proven to be valuable when employed in optical character recognition system by better preserving the original image's characteristics and, in some cases, improvement in processing time.

2.3. Summary

The objective of the pre-processing stage is to enhance the original digital image so that the resulted image is more suitable to extract the morphological attributes needed for an optical character recognition process. This stage, in part, correlates to the processing that is perform to the light that is reflected from the lens on to the retina of the human eye (see section 1.2.1). Generally, this stage is executed on a digital computer. Specialized hardware, however, will greatly improve the overall speed of the enhancement procedures. Still, this stage could easily be implemented for parallel execution as a preprocessing stage for the attribute extracting and recognition phase (discussed in sections 4.3.2-4.3.4) on a MIMD machine or a dedicated pipeline architecture.

An optical scanner usually produces various gray shades rather than a binary image. One could easily convert a digital image into a binary image by a high pass filter. This filter accentuate the high frequency details of the image while leaving the low frequency content intact. That is, the higher-frequency portions of the image will become brighter while the lower-frequency portions become darker. This filter will increase the image's sharpness and also accentuate image's noise. This noise, then, will be more likely to be filtered with a median or convolution filter analysis.

The presence of noise within a digital image blurs the image and confuses other vision operators. In binary images, noise takes the form of stray pixels; that is, white pixels that should be black or black pixels that should be white. A common noise filtering routine is median filtering. Since the binary image's function (typically used in OCR applications) is an assortment of jumps between zero and one (black and white), rather than a discrete gradual transition of gray shades, median filtering is not adequate in processing binary images.

By way of convolution technique, one could employ a noise reduction function represented by a mask *M*. The essence of the convolution method is to determine pixel's gray shade (or a decision between black or white in binary images) upon the consideration of the pixel's neighbor pixels. Convolving a mask around a digital image sits well under SIMD-MC architecture.

Thinning digital patterns is an important pre-processing phase in processing digital images. A graph of an image is a common approach for representing an image's structural shape[34]. This graph is obtained by approximating the shape's skeleton via a thinning algorithm. This preprocessing phase is used in many image analysis operations such as document processing[6], fingerprint recognition[58], and biomedical systems[25]. The contribution of this phase is to accent the structural shape of the image, and reduce the number of pixels to be processed in determining the semantics of the image.

Algorithms I and II presented in sections 2.2.3.1 and 2.2.3.2, are fast robust skeletonizing methods. These algorithms improve existing algorithms by eliminating the degradation problem. Their usage has proven to be valuable when employed in optical character recognition systems by better preserving the original image's characteristics and, in some cases, the improvement of processing time.

3. Character Isolation

Isolating a character from an image X classically means the identification of an integer c $(2 \leq c \leq n)$ and a partitioning of X as c mutually exclusive, collectively exhaustive subset of X. These partitions are usually referred to as *clusters* or *regions*. In image processing, the members of each cluster bear more similarity to one another that to members of other clusters. At least, this will be true for mathematical similarities between the x_k's in some well-defined operational sense. One then hopes that the same substructure exists in the data-generating process itself. Cluster structures in X may reveal associations between individuals of a population. Associated with the clustering problem is (when not pre specified) the identification of c, and the most appropriate number of clusters in X. This important question is called the cluster validity problem[44]. A common question that arises in connection with clustering and classification is if the features (or characteristics) of data item $x_k \in X$ sufficiently represent the physical process to enable us to construct realistic clusters or classifiers. In other words, do we have the "right" data space in x_k? Should some features in x_k be discarded? modified? enriched? or transformed [53]?

3.1. Region-Oriented Segmentation

Segmentation of regions is based on intensity discontinuities. If X represents the entire image as a region, then segmentation is the process that partitions X into n subregions, x_1, x_2, x_3, x_4, ... x_n, such that

(a) $\bigcup_{i=1}^{n} x_i = X$,

(b) x_i is a connected region, $i = 1, 2, 3, \ldots, n$,

(c) $x_i \cap x_j = \varnothing$ for all i and $j \ni i \neq j$,

(d) $P(x_i) = True$ for $i = 1, 2, 3, \ldots, n$,

(e) $P(x_i \cup x_j) = False$ for $i \neq j$,

where $P(x_i)$ is a logical predicate defined over the points in cluster x_i, and $\varnothing$ is the null set.

These conditions indicate that the segmentation must be complete, a region comprises of pixels that are connected, regions are disjoint from each other, a region conforms to a specific property, and the property which deviate x_i and x_j.

3.1.1. Region Growing

Region growing procedures are based on the selection of a "seed" collection of pixels that are conceived to belong to the same region. Then the original region, x_i, is grown by adjoining neighbor pixels which have similar properties (gray level, texture, color and such) to x_i, a procedure also referred to by *pixel aggregation.*

Consider the example given by Gonzalez and Wints illustrated in Table 3.1a. where the numbers indicate the cell's representative gray-level values[34].

Table 3.1a Original Image gray levels.

	1	2	3	4	5
1	0	0	5	6	7
2	1	1	5	8	7
3	0	1	6	7	7
4	2	0	7	6	6
5	0	1	5	6	5

Table 3.1b Segmentation using 3 difference.

	1	2	3	4	5
1	a	a	b	b	b
2	a	a	b	b	b
3	a	a	b	b	b
4	a	a	b	b	b
5	a	a	b	b	b

Table 3.1c Segmentation using 8 difference.

	1	2	3	4	5
1	a	a	a	a	a
2	a	a	a	a	a
3	a	a	a	a	a
4	a	a	a	a	a
5	a	a	a	a	a

Let the points with coordinates (3,2) and (3,4) be used as seeds. Using two starting points will result in a segmentation consisting of, at most, two regions: x_1 associated with seed (3,2) and x_2 associated with seed (3,4). The property P that we used to include a pixel in either region is the absolute difference between the gray level of a pixel and the gray level of the seed to be less than a threshold T. Any pixel that satisfies this property simultaneously for both seeds is assigned to region x_1. The result obtained using $T = 3$ is shown in Fig. 3.1b. In this case, the segmentation consists of two regions where the points in x_1 are denoted by a's and the points in x_2 by b's. Note that any starting point in either of these two resulting regions would have yielded the same result. If, on the other hand, we had chosen $T = 8$, a single region would have resulted, as shown in Fig. 3.1c.

Once the logical predicate *P* and the seed coordinate are established, the region growing procedure is quite straight forward. It can be implemented efficiently on a MIMD machine by setting each CPU with a seed to be grown in parallel to other regions. Still, the selection of the seed coordinates depend on the image classification. In general, a threshold for bright pixel, T_h, is set for seeds to a background or object structures (depending on the digital image type), and a dark threshold, T_d, that is fixed for the complement seeds selected by T_h. When a priori information is not available one may proceed by computing at every pixel the same set of properties that will ultimately be used to assign pixels to regions during the growing process. If the result of this computation shows a set of cluster localities, then the pixels whose properties place them near the centroid of these clusters can be used as seeds. Fig. 3.1 is the gray histogram for the example depicted in Table. 3.1.

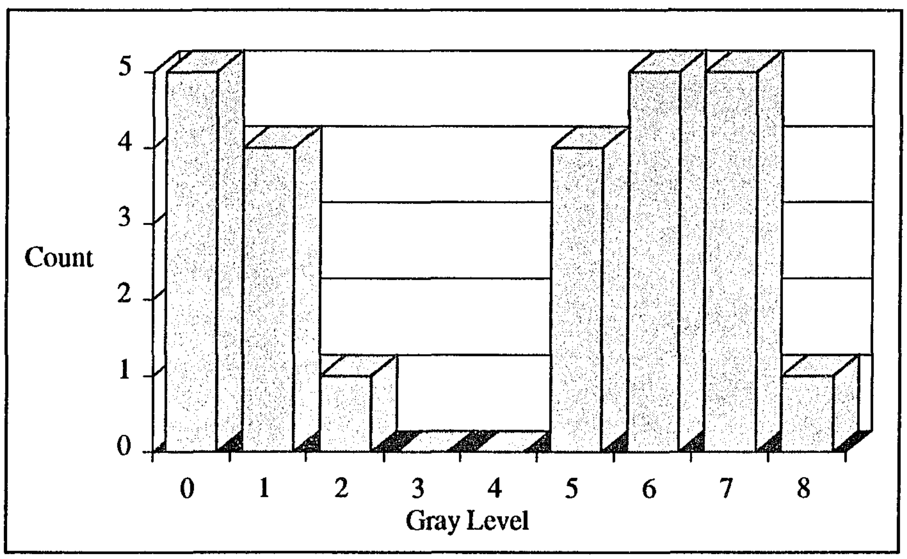


Fig. 3.1 Gray level histogram.

The histogram suggests that points with intensity of 0 and 7 are predominantly contrasted gray levels that could designate T_d and T_h respectively.

Selecting an appropriate logical predicate *P* depends not only on the problem under consideration, but also on the type of image data available. Color images

analysis of satellite imagery, for example, is significantly easier to handle than monochrome images alone. This is since differences in color are important information in differentiating region seeds. In monochrome images, region analysis must be carried out using a set of descriptors based on intensity and spatial properties. Descriptors alone can yield misleading results if connectivity or adjacency information is not used in the region-growing process. In a binary image of an English text, for example, it is possible that two or more characters are connected. Adjacency information of connected characters regions, and the type of connectivity should be considered in order to ensure that the region growing process will not combine regions that should not be merged.

Formulating a stopping rule for a region growing should be developed with great consideration to the processed image characteristic. Basically, we stop growing a region when no more pixels satisfy the criteria for inclusion in that region. The intensity criteria do not take into account the *history* of the grown region. Additional criteria that increase the power of a region-growing algorithm incorporate the concept of size, likeness between a candidate pixel, the pixel's growth thus far, and the shape of a given region being grown[34].

3.1.2. Region Splitting and Merging

The essence of the splitting and merging approach is to initially subdivide an image into a set of arbitrary disjointed regions, and then merge and/or split the regions in an attempt to satisfy the conditions of the region-oriented segmentation outline Splitting the region is done by partitioning a region x_i into smaller quadrant region making sure that :

$$\forall x_i \in X, \qquad P(x_i) = True \qquad (3.1)$$

where $P(x_i)$ is the logical predicate discussed in previous section.

If $P(x_i) = False$, then we repetitively subdivide that quadrant into sub quadrants until equation 3.1 is satisfied. This particular splitting technique has a convenient representation in the form of a so-called quadtree (see Fig. 3.2). Note that the root of the tree corresponds to the entire image and that each node corresponds to a subdivision. In this case, only x_4 was subdivided further[34].

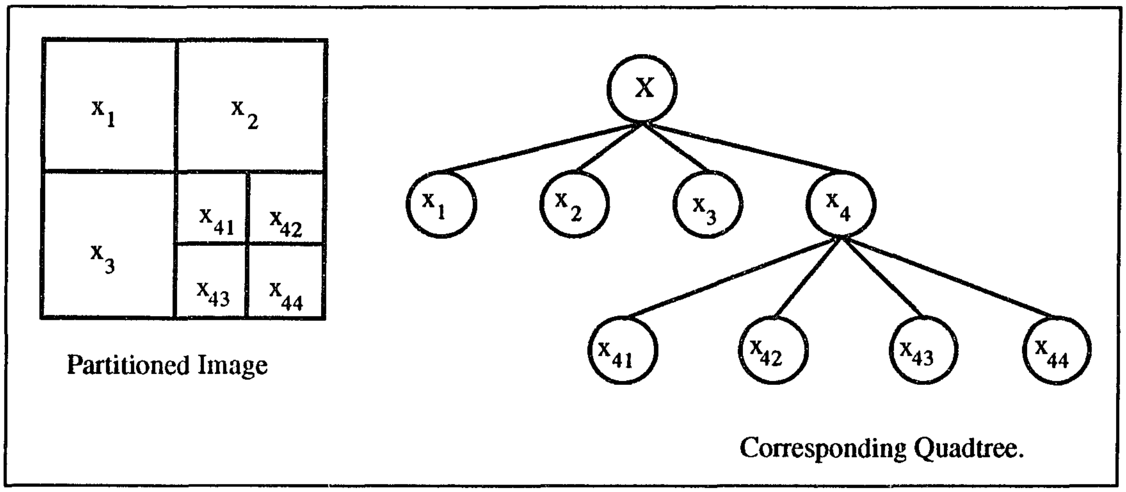


Fig. 3.2 Region splitting.

The splitting procedure partitions x_i into four quadrants without consideration of the scenario in which adjacent quadrants maintain the same property; that is, $P(x_i \cup x_j) = True$. This fault is then corrected by employing a merging action to be taken after a split operation. The merging activity is taken place only for adjacent regions whose combined pixels satisfy the predicate P. Various variations of the split/merge process are rendered by Horowitz[40].

Fig. 3.3 illustrates the split/merge algorithm as rendered by Fu, et al[32]. The image under consideration consists of a square image with one object (dark pixels) and a background. The object and background have constant gray levels, thus $P(x_i) = True$ is a simple predicate to employ. Since the gray level of the background and object are significantly distinct, then $P(X) = False$. Thus, the split process is taken place as shown in Fig. 3.3a. The left top region satisfies the predicate P, and therefore it is not processed further. No merging process is activated, since in the rest of the three quadrant P is not maintained. Next, those three quadrants are partitioned further as illustrated in Fig. 3.3b. At this point, several regions can be merged, with the exception of the two sub quadrants that include the lower part of the object; these do not satisfy the predicate and must be split further.

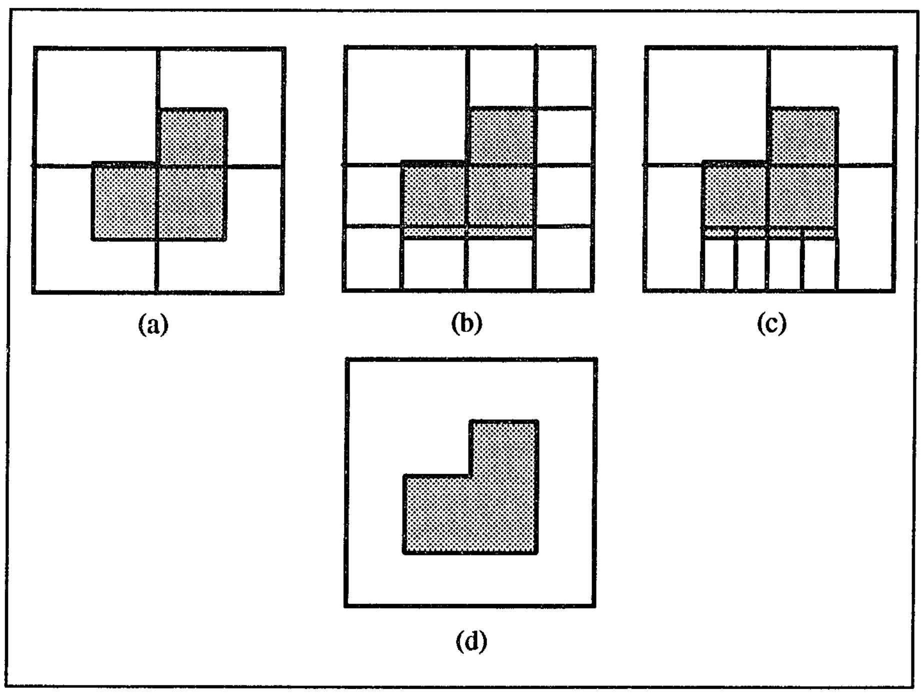


Fig. 3.3 Split and merge stages of one object and a background.

The results of the split-and-merge operation are shown in Fig. 3.3c. At this point all regions satisfy P, and merging the appropriate regions from the last split operation yields the final segmented result shown in Fig. 3.3d.

3.2. On-line Approach

The region-rriented segmentation methods described in the previous section are based on the notions that the entire digital image is available for processing. However, the main idea behind the development of an optical character recognition system is to enable a feasible batch translation of textual documents into ASCII. In essence, a Xerox-machine like scanner should process a batch of documents (positioned on its paper feeder) and pipe the digital reflection of the documents onto a recognition system. These fast scanners are available nowadays and are frequently used to capture digital images of various type of documents. If an OCR system is to process the digital images conceived by these fast scanners, it is advantageous that the recognition system begin its operation as the information from the scanner becomes available. This will speed up the system significantly. Moreover, a digital image of a document is comprised with vast number of pixels. Maintaining all these pixels in main memory for processing may not be possible. This is true since even a super computer has limited resources. It is best, then, to break the image into clusters such that background clusters will be discarded immediately.

The scanner produces its image reflection in a line by line fashion; that is, depending on the scanner's vertical resolution the document's Y axis is partitioned into a sequence of lines. Each line is then further partitioned into small partitions, whose size depends on the scanner's horizontal resolution. The spatial partition of the document, horizontally and vertically, results in a sequence of small squares, $p_{i,j}$, referred to as pixels. In a binary scanner (often use for OCR applications), a pixel is considered black if the reflected $p_{i,j}$ gray shade is over a certain threshold. A white pixel is recorded if the reflection gray level is below an established threshold. The scanning is performed

by bombarding the document with an horizontal line of bright light. The reflection of that light is rebounded (by way of mirrors) on to discrete electrical components that are sensitive to light. These components produce (after passing some logical circuitry) a sequence of pixel values (black or white) corresponding to the light slit reflected from the document at a particular vertical location. In this dissertation, we conformed to the rule that the background color is white, and that the textual image is reflected by dark pixels. Also, the pixel information obtained from the scanner is accomplished by acquiring a line of pixels at the a time.

Since an on-line recognition system, in general, obtains the image's spatial pixel distribution in a line by line fashion (either from a scanner, or by reading the image's information from a file), an on-line isolation technique should be used. This isolation technique is the combination of the region growing and the split/merge techniques described in previous sections. The recognition system developed in this dissertation is designed to recognize the English character that are not connected to each other (connected characters which form words should be recognized as one entity rather than the break down of a sequence of characters as described in section 5) or at worst loosely connected. Thus, the principle guideline is that the English character is surrounded with mostly background pixels. The essence of the isolation technique is to maintain a window, W, to relate to the latest $2\lambda+1$ lines introduced to the system from the scanner, where λ is a small constant[1]. Choosing λ should be based on the scanner's vertical resolution and the degree of possible background noise. It is possible that dark pixels which are part of the figure were not recorded. This results in pattern discontinuity. The height of W will determine the magnitude of discontinuity which will be considered in constructing a cluster of a single character figure. The width of this window, W^x, clearly depends on the horizontal resolution of the scanner. The intent of

1. In our implementation, we set $\lambda = 2$. Thus, we maintained a window of the latest 5 rows of pixels captured by the scanner.

W is to merge dark pixels (recognized by the scanner) with some degree of discontinuity into one distinct cluster x_i.

In essence, W is a two dimensional array of indexes to clusters (see Fig. 3.4). The dimension of W varies from $-\lambda$ to λ vertically, and from 0 to W^x-1 horizontally. At the beginnings W's elements are initialized to null; that is, no cluster, representing a subset collection of image pixels (which represents a reflection of a character), was detected yet. At the point where a line of pixels is introduced to the system's buffer, B, the information in W is shifted one line upward ($W_{i-1} = W_i$ and the bottom line W_λ is initialized to null).

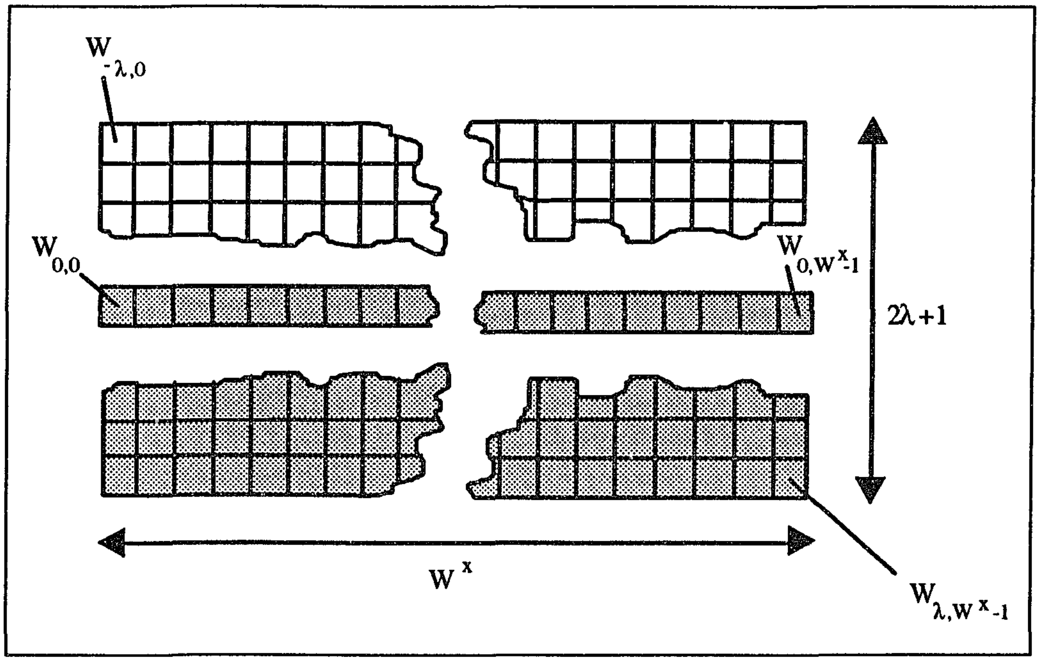


Fig. 3.4 W is a two dimensional array of indexes to clusters x_i.

A left to right scan is performed to find the column range, R=[L,R], of the first sequence of dark pixels in B. At that point, a window in W is examined for non-null elements. The elements consulted should have the spatial coordinates enclosed by $W_{0,L-\alpha}$ and $W_{\lambda-1,R+\beta}$ (that are the left top corner and the bottom right corner respectively), where

$$\alpha = Max\,(\mathrm{L}-\Delta,0) \tag{3.2}$$

and

$$\beta = Min\,(\mathrm{R}+\Delta,W^{x}-1) \tag{3.3}$$

where Δ is a small constant, L is the left most coordinate of the sequence, and R is the right most.

The value of Δ will determine the degree of horizontal discontinuity to be allowed in merging figure clusters[2]. In other words, λ rows of W (beginning in W_0) with column range of $[\alpha,\beta]$ will be examined for non null values. If $W_{i,j} = k$ and $k \neq$null, where $i \in [0,\lambda-1]$ and $j \in [\alpha,\beta]$, the system will add the pixel sequence in B to cluster x_k. In addition, $W_{\lambda,L}$ up to $W_{\lambda,R}$ will be set to k, indicating these location to belong to cluster x_k as well. This will designate that cluster x_k embodies the range of pixels in question (R=[L,R]) for future reference. If, on the other hand, all $W_{i,j}$ are null, then a new cluster will be established to embody this pixel range.

It is possible that a set of indexes, S, was found while examining $W_{i,j}$ for null entries. In this case, the clusters whose indexes are in S including the pixels in question will be merged into a single new cluster, x_{merge}. Also, for $\forall W_{i,j} \in$ S, $W_{i,j}$ will be set to the new index *merge*.

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	*	*	*			*	*	*			*	*	*
1											*	*	*
2			*	*	*								
3			*	*									
4			*	*	*	*	*	*	*	*	*	*	

Fig. 3.5 Binary image sample.

2. In our implementation we set $\Delta = 2$.

Consider, for example, the action taken by the isolation technique on the image illustrated in Fig. 3.5. Also, assume that $\Delta=1$, and $\lambda=2$. W^x in this case is 13. First, the elements in W are initialized to null (noted as ϕ). The first line entered to the buffer (B) is row_0 of the image. Since $W_{i,j} = \phi$ ($\forall$ i $\in$ [0,λ-1] and j $\in$ [0,12]), three clusters will be established x_1, x_2, x_3 where x_1 maintain the left range of three pixels, x_2 the second range and x_3 the right most range of three pixels. The window W is illustrated in table 3.2.

Table 3.2 Resulting W after processing the first image line.

	0	1	2	3	4	5	6	7	8	9	10	11	12
-2	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ
-1	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ
0	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ
1	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ
2	**1**	**1**	**1**	φ	φ	**2**	**2**	**2**	φ	φ	**3**	**3**	**3**

Next, the second image line (row_2) is submitted to the buffer and the rows in W are shifted one row upward resetting row W_2 to nulls. In this case the first range of pixels (the only range), R=[10,12] is detected.

Table 3.3 Resulting W after processing the second image line.

	0	1	2	3	4	5	6	7	8	9	10	11	12
-2	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ
-1	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ
0	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ
1	**1**	**1**	**1**	φ	φ	**2**	**2**	**2**	φ	φ	**3**	**3**	**3**
2	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ	**3**	**3**	**3**

In this case, using equations 3.2 and 3.3 α=9 and β=12. $W_{1,10}$, $W_{1,11}$, and $W_{1,12}$ are the only elements in the window [$W_{0,10}$, $W_{0,11}$, $W_{0,12}$, $W_{1,10}$, $W_{1,11}$, $W_{1,12}$] that are not null. Moreover, all three are pointing into cluster x_3. Thus the pixel range in B are added into cluster x_3. The resulted W is shown in table 3.3. Again, the third image line is submitted to the buffer and the rows in W are shifted one row upward resetting row W_2 to nulls. In this case, the first range of pixels (the only range) R=[2,4] is detected. Using equations 3.2 and 3.3 α=1 and β=5. $W_{1,1}$, $W_{1,2}$, and $W_{1,5}$ are the only elements in [$W_{0,1}$, $W_{0,2}$, $W_{0,3}$, $W_{0,4}$, $W_{0,5}$, $W_{1,1}$, $W_{1,2}$, $W_{1,3}$, $W_{1,4}$, $W_{1,5}$] that are not null. Thus, W's window points to two clusters, x_1 and x_2. Therefore, x_1 and x_2 are merged together with the buffer's pixel range into a new merged cluster, x_4. The resulted W is shown in table 3.4.

Table 3.4 Resulting W after processing the third image line.

	0	1	2	3	4	5	6	7	8	9	10	11	12
-2	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ
-1	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ
0	4	4	4	ϕ	ϕ	4	4	4	ϕ	ϕ	3	3	3
1	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	3	3	3
2	ϕ	ϕ	4	4	4	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ

At this point, we have only two clusters, x_3 and x_4. The result of processing the last two image lines will produce in the same fashion the following W (see table 3.5) reflecting on two clusters, x_3 and x_4.

A cluster is considered complete, when it is not active for ε iterations (processing a line from B is considered one iteration where ε is a predefined constant). Again, the magnitude of ε should be affirmed in consideration to the vertical resolution

of the scanner, the general vertical distance of textual character that are expected to be scanned, and the average character height. A completed cluster is then submitted into the recognition phase.

Table 3.5 Resulting W after processing the whole image.

	0	1	2	3	4	5	6	7	8	9	10	11	12
-2	4	4	4	φ	φ	4	4	4	φ	φ	3	3	3
-1	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ	3	3	3
0	φ	φ	4	4	4	φ	φ	φ	φ	φ	φ	φ	φ
1	φ	φ	4	4	φ	φ	φ	φ	φ	φ	φ	φ	φ
2	φ	φ	4	4	4	4	4	4	4	4	4	4	φ

When analyzing the image of Fig. 3.5, the isolation of two clusters x_3 and x_4 representing the digital pattern depicted in table 3.5 seems to be a desired solution. Still, in many cases when isolating figures of the English characters, clusters x_3 and x_4 could be subsets of a single pattern; that is, x_3 and x_4 should be merged into one cluster that depicts one symbol. This merging will be performed in the fusion procedure described in the next section. However, we are going to augment the procedure described above to include the following logic:

> At the end of each iteration just before accepting a new image line into B, the system should examine another subset of W, $W_{m,n}$ = k such that k $\neq$ null and where $m \in [-\lambda,-1]$ and $n \in [\alpha,\beta]$. In the case that $W_{\lambda,L} \neq W_{m,n}$, $W_{\lambda,L} = y$ and $W_{m,n} = z$, then clusters x_y and x_z are considered complete only when both are not processed in the last ε iterations (for L be the first columns of the pixel sequences in B).

The addition of the above logic will guarantee that clusters which have the potential to merge, will not be submitted to the recognition system until both clusters where considered for possible fusion analysis and found to be complete.

3.2.1. Cluster Fusion

In many cases humans do not print an English character by connecting the figure's pattern, as they should. Consider the character 'E' which was scanned with 200DPI binary scanner and is illustrated in Fig. 3.6. The on-line procedure discussed above will partition this image into three clusters. These clusters will maintain the pixel pattern of the bottom horizontal pattern, the top horizontal pattern, and the horizontal 'T' shaped junction pattern.

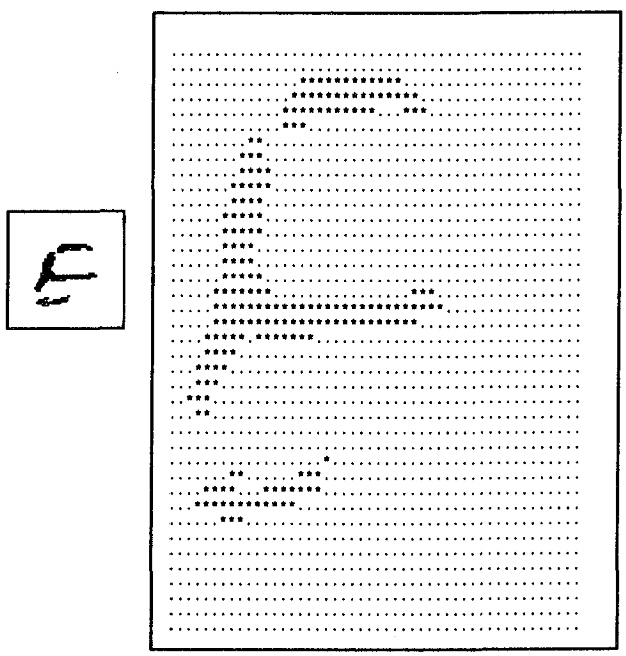


Fig. 3.6 Digital scan of the character 'E."

It is necessary, then, to be able to merge clusters which constitute a subset of a single figure. The fusion phase considers most clusters generated by the on-line isolation procedure. The investigation is accomplished by examining two clusters at a time, x_a, and x_b. Clearly neither x_a nor x_b are complete. Still, a cluster pair is considered for merging if at least one of x_a, or x_b had been idle (no pixel was embedded into the cluster) more than λ iterations (then the cluster's index will be above W_1). This is to avoid merging conflicts with the on-line procedure (the on-line isolation procedure can be performed in parallel with the fusion procedure). If one of x_a and x_b had not been idle for more then λ iterations, say x_a, than its index (*a*) is still recorded in the bottom portion of *W*. Thus, if a merging is to occur, x_b will be appended into x_a rather than to have the creation of a newly merged cluster. If a new merged cluster is obtained, then *W* would have to be updated; this is not a desired approach if the merging and on-line isolation are to be executed in parallel independently. The essence of the merging routine is to merge two clusters whose pixel spatial distribution overlap horizontally. Since both clusters are not complete, they overlap each other vertically as well. This is because when writing printed English characters these figures do not seriously overlap each other. Still, horizontal overlapping will not suffice to justify merging clusters that encompass figures with comparable spatial height. Consider the characters 'U', 'N', and 'I' scanned with 200DPI and illustrated in Fig. 3.7. In this case one should examine the clusters further to guarantee that the horizontal overlapping occurs on the same horizontal position. Although the figure 'U' and 'N' in Fig. 3.7 overlap horizontally, they never overlap on the same spatial row. Merging short figured clusters into tall clusters, will at worst add some noise to the merged cluster. Noise tolerance, however, is one of the characteristics of the recognition system; its ability to generalize.

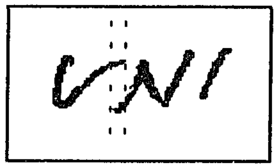


Fig. 3.7 Horizontal overlapping of approximately equal height clusters.

Then, merging x_a, and x_b will take place if the following conditions are satisfied :

> Neither x_a, nor x_b is complete. At least one of x_a, x_b, had been idle for more then λ iterations, say x_a. In this case if merging is to take place, x_a's pixels will be appended into x_b. Also, define *Height*(x_i) to be the spatial height of the figure maintained in x_i. Then merging will follow if x_a, and x_b's pixel distribution overlap horizontally, and *Height*(x_a) is significantly smaller or significantly greater then *Height*(x_b) (merge small figures into large figures). In the case in which the cluster's height are approximately the same, one should validate further that the horizontal overlapping[3] occurs on the same horizontal row.

This procedure could be executed in parallel with the on-line isolation, and with various cluster pair fusion consideration. The merging analysis frequency should be in response to ε.

3.2.2. Cluster Fission

Noise considerations is an integral part of image processing. In many cases, scanning images of textual print results in character images that are loosely connected. This could also be the result of the hand writing style of the original document. Thus,

3. The term "overlapping" should sustain some flexibility. This is to accommodate missing dark pixels as a result of background noise.

before cluster x_i is submitted for recognition, the system should consider, in some cases, partitioning the figure in x_i into two or more clusters as illustrated in Fig. 3.8.

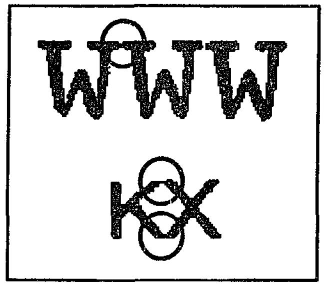


Fig. 3.8 Character figures that are loosely connected.

Careful examination of the English alphabet (see Appendix A) will show that in general, the height of a single alphanumeric character is greater than its width. It is rare, however, that a character's width is significantly greater than its height. This property can be used as a condition for a cluster fission consideration; that is, cluster x_i will be consider for a fission analysis if and only if the cluster maintains a spatial distribution of pixels whose width is significantly greater then it height.

The fission analysis is done by searching an existence of a *fuzzy path* (see Fig. 3.9) from the spatial top of the cluster to its bottom. This path may go through dark pixels as long the fuzzy path maintains a required criteria. The path construction is achieved by the formulation of a logical predicate, *P*, which determines if a path exists between neighboring pixels. The application of *P* to the figure's boundary is done by a method that is similar to the Jarvis march technique [66]. In order to decrease the chance for an erroneous path the march's horizontal boundary will be an horizontal central portion of the cluster (see Fig. 3.9). Moreover, a bias will be given to a search dowered then to the right. This will guarantee to find the left most fuzzy path.

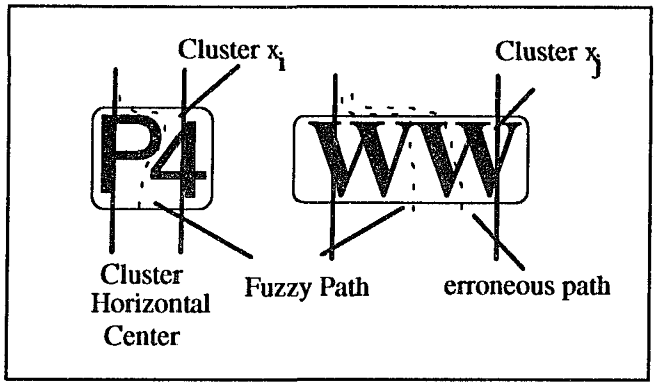


Fig. 3.9 Fuzzy path within a cluster.

Jarvine's march is an improved algorithm which finds the convex hall to a set of points that are distributed on the spatial plane[30]. First, we determine the central horizontal boundary upon which a fuzzy path could go thorough. Typically, this should be around 20% from each side. This is because the thikness of a character is usually smaller then 20% of the image's width. Then we go down to the first dark pixel $p_{i,j}$ on the figure's boundary. We keep going down as long as $P(p_{i,j})$ is true. At each location we add to a location set, **S**, the subset of all pixels that are on the boundary of the figure and satisfy P. Repeat this process until the walk reached the bottom of the cluster, or until all $p_{i,j} \in$ **S** were processed.

3.3. Summary

Before the textual structure of an image could be analyzed, one should break the given image into a set of partitions such that each partition encompasses a figure of a single character. Segmentation is a region partition methodology that is based on intensity discontinuities. If X represent the entire image as a region, then segmentation is the process that partitions X into n sub regions, x_1, x_2, x_3, x_4, ... x_n with regard to the conditions described in section 3.1. Most segmentation methods are based on capturing homogeneity contrast measures by linear or logarithmic gradient response of

the gray level distribution function[68]. In the recognition of textual images, all alphanumeric figures should have approximately the same gray levels. Two methods which employ binary images have been discussed: region growing, and region splitting and Merging. Since these methods are based on the notions that the entire digital image is available, we have developed an on-line version of those methods. The reason is to enable a feasible batch translation of textual documents into ASCII. Also, the on-line method does not necessitate the entire digital image to reside in main memory.

An on-line recognition system, in general, obtains the image's spatial pixel distribution in a line by line fashion (either from a scanner, or by reading the image's information from a file). The recognition system developed in this dissertation is designed to recognize English characters that are not connected to each other (connected characters which form words should be recognized as one entity rather than the break down of a sequence of characters as described in chapter 5) or at worst are loosely connected. Thus, given the knowledge of the expected features of the English alphanumeric figures, we have developed an isolation system which tries to merge collection of pixels into different image clusters. In addition, a fusion phase is used to enable the merger of clusters which constitute a single alphanumeric figure but were not close enough on the spatial plane. Also, a fission phase could be conditionally executed to enable the partitions of clusters which embody pixel patterns of more then one alphanumeric figure. This condition could be the result of excessive noise or the hand written style of the original document.

All three phases could be executed in parallel. Once the cluster's construction is completed it is submitted to the recognition system. The recognition stage, then, assumes that the digital pattern embodied in the cluster, is a single and unique digital pattern of an English alphanumeric pattern. Thus, an MIMD machine will be able to process a set of complete clusters asynchronously.

4. Recognition

The term "pattern recognition" embraces such a vast and diversified literature that a definition of it always invites debate [44], because this term describes an application which attempts to recognize a pattern from given data[75]. These patterns could be market trends, galactic positions, stock market trend, and object structures (given a data of a digital image). When processing digital images, different applications have different meaning to the semantics of the term "pattern recognition" as well. Recognizing a structure of a box within an image implies a recognition system that is quite general and flexible because the reflected shape of a box is greatly determined by the angle and elevation from which the box's reflection was captured. In addition, the term "box" is quite general in term of many possible sizes and geometrical shapes. Matching two fingerprint images, however, has no flexibility in its recognition process. The images should superimpose each other exactly in order for the recognition system to conclude a match.

In this dissertation we define the term "pattern recognition" by identifying particular morphological attributes within an image cluster that represents an image of a character in the English character set.

The process of detecting, classifying and recognizing a visual stimulus can be analyzed from the standpoint of statistical decision theory. The recognition system can be thought of as estimating the prior and posterior probabilities that the stimulus was actually present, and then using some criterion to make a decision. The choice of this criterion determines the probabilities of the system missing an actual stimulus and of reporting a stimulus when none was actually present. The tradeoff between these probabilities, as a function of the criterion used, is called the observer's "receiver

operating characteristic" (ROC). From the decision-theoretic standpoint, one should speak not only of a "threshold" detection, but also of criteria detection [21].

Perceptual decision criteria are influenced by many factors besides the physical nature of the stimulus. Expectation or "perceptual set" play an important role; in particular, it affects the estimation of prior probabilities[7].

4.1. Fuzzy Logic

Life is full of uncertainties. As human beings we must learn to accept that uncertainty is a part of our life, and will continue to be part of it in the future as well. As scientists we must also accept the fact that uncertainty is an ever increasing part of our work day as the speed with which information is transferred results in vast amount of information that is needed to be processed[6]. Fuzzy systems, then, seem to possess the ability to work well in a natural environment where events never repeat exactly, but, on the other hand, are not completely different. The term *fuzzy* was first introduced in [52]. The fuzzy set concept in Boolean algebra is known as the characteristic function of a set.

4.1.1. Introduction

In interpreting the semantics of an image, the human's vision system determine that an object is a part of that image if the system recognizes a subset of properties to which the system was exposed to in the past. This partial information is sufficient to

determine if an image indeed captures the reflection of the object in question. Still, a subset of properties does not necessarily guarantee the existence of that object to begin with. Consider, for example, how the vision system responds to the Kanizsa square as it is pictured in Fig. 4.1[48]. The Kanizsa square exists in our brain. It does not exist "out there" in the physical reality on the page. Out there there are only four symmetric ink patterns that stain the page[11].

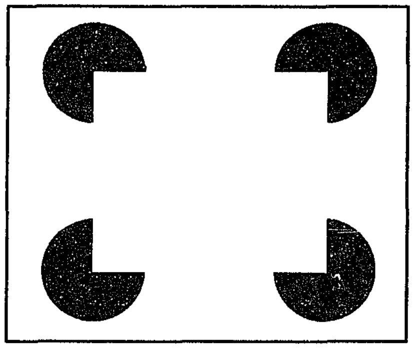


Fig. 4.1 Kanizsa square.

Although a white colored square was not explicitly drawn in Fig. 4.1, one could certainly acknowledge that this picture depicts a *partial* white colored square. At least, crucial information that is synonymous with square properties is depicted in that figure. This is the basis upon which fuzzy logic is built. In fuzzy logic there is no meaning to absolute *true* or absolute *false*[38].

In fuzzy logic, the meanings of true or false are used in conjunction to some information to the levels to the truthness or falsness of a given property. Consider the collection of n grains of send that are shaped into a heap of sand (noted by $heap_n$)[11]. Removing one grain leaves n-1 grains (denoted by $heap_{n-1}$). Still, one will continue to consider $heap_{n-1}$ as a heap of send (assuming that n is large enough). If one continue to remove grains of send from the heap repetitively, eventually, the heap will disintegrate. There is no distinctive point in time such that removing a grain of sand will transform the heap to a non heap. One could say that the value of $heap_k$ where $0 \leq k \leq n$ is a

gradual transition from True ($heap_k$ is a heap) to False ($heap_k$ is not a heap). In other words, the characteristic function of $heap_k$ should return a discrete value, between True and False, that is computed by a function, $\mu_{heap}(k)$, which reflect the linear transition of $heap_k$ from a heap to a non-heap.

Fuzziness describes event ambiguity[11]. Consider for example figure 4.2. This binary image illustrate the hand written letters 'e', 'l', 'e' (Figures 4.2a, 4.2b, and 4.2c respectively). However, one could interpret them as 'l', 'l', and 'e'.

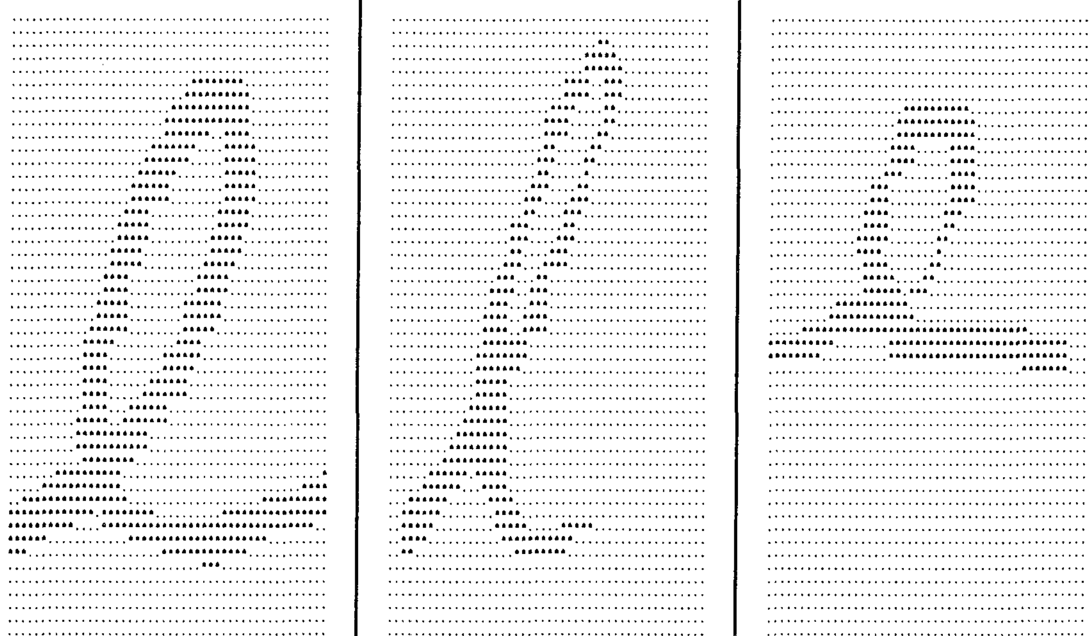


Fig. 4.2a 'e' character. **Fig. 4.2b** 'l' character. **Fig. 4.2b** 'e' character.

Fig. 4.2 The difference between characters may not be clear.

Determining which English character Fig. 4.2a actually represents depends greatly on the person which wrote it. Recognizing this image, then, would be done by comparing this image characteristics and information obtained from previous exposures to similar images. In addition, the content of the whole image upon which Fig. 4.2a was isolated could be a major factor in resolving the ambiguity and concluding Fig. 4.2a as an 'e' or 'l'.

4.1.2. Main Properties of Fuzzy Numbers

In a scenario in which a value is uncertain, one can accept that the uncertain value belongs to a referential set in **R** (the set of real numbers). We can then represent this value as a closed interval of **R** referred to by the *interval of confidence.* Assume two confidence intervals:

$$A = [a_1,a_2] \tag{4.1}$$

and

$$B = [b_1,b_2] \tag{4.2}$$

Hence, if $x \in A$ and $y \in B$ then,

$$x + y \in [a_1+b_1,a_2+b_2] \tag{4.3}$$

Symbolically, we write

$$A\ (+)\ B = [a_1,a_2]\ (+)\ [b_1,b_2] = [a_1+b_1,a_2+b_2] \tag{4.4}$$

Also,

$$A\ (-)\ B = [a_1,a_2]\ (-)\ [b_1,b_2] = [a_1-b_1,a_2-b_2] \tag{4.5}$$

The commutative and associative properties are preserved as are regular set operations. Multiplication and division are illustrated in equations 4.6 and 4.7:

$$A\ (\cdot)\ B = [a_1,a_2]\ (\cdot)\ [b_1,b_2] = [a_1 \cdot b_1,a_2 \cdot b_2] \tag{4.6}$$

$$A\ (:)\ B = [a_1,a_2]\ (:)\ [b_1,b_2] = [a_1/b_1,a_2/b_2] \tag{4.7}$$

The interval of confidence is the interval in which the uncertain value could vary within the interval [0,1]. Still, one could measure a confidence interval with a level that is different from 0, such as $[\alpha,1]$. In this case the property of the confidence interval at a particular level has the following property:

$$\forall \alpha_{1,}\alpha_2 \in [0,1],\ (\alpha_1 < \alpha_2) \Rightarrow ([a_1^{(\alpha_2)}, a_2^{(\alpha_2)}]) \subset ([a_1^{(\alpha_1)}, a_2^{(\alpha_1)}]) \qquad (4.8)$$

This means that if α increases, the interval of confidence must decrease. Such a situation is shown in Fig. 4.3[6].

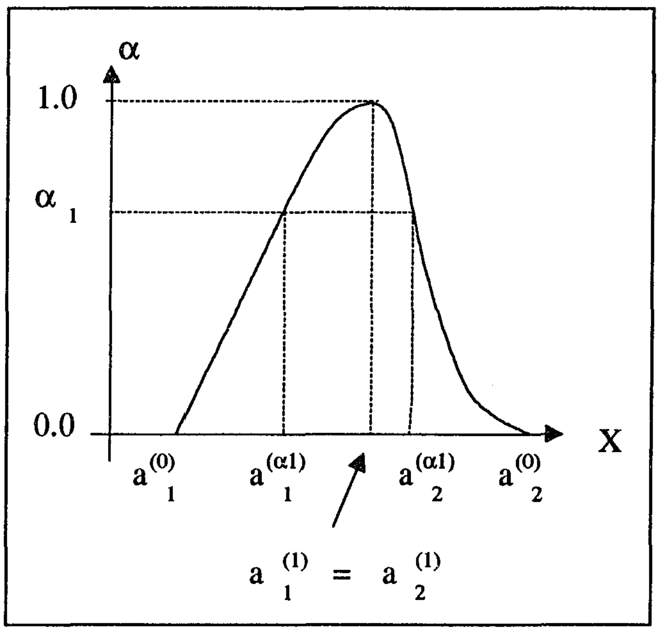


Fig. 4.3 Definition of fuzzy numbers from [6].

All the basic principles of set theory, except the axiom of extension, are designed to make new sets out of old ones[64]. In classical set theory and logic a "parallel postulate" is that $\mathbf{A} \cap \mathbf{A}^c = \varnothing$ where $\mathbf{A}^c$ is the complement of **A**. This is not the case with fuzzy logic and fuzzy numbers. Let the set of all glasses that are partially empty denoted by *E*, which is also the collection of glasses that are neither full, nor empty. Then E^c is the set of all glasses that are not partially empty. If one is to

interpret the meaning of E^c intuitively, rather than logically, then $E \cap E^c \neq \varnothing$. To resolve ambiguity of this kind the concept of a *fuzzy number* is presented using the combination of a level of presumption, α, and the interval of confidence, $[a_1,a_2]$ as suggested by Kaufmann, et al [6]. Then, if E is some referential set (such as **R** or **Z**), and A is an ordinary subset of E, then:

$$\forall x \in E \qquad \mu_A(x) \in \{0,1\} \tag{4.9}$$

where μ_A is the characteristic function of set A.

This shows that an element of E belongs to, or does not belong to A according to the value of the characteristic function. The characteristic function of a regular set is either zero or one. The characteristic function of a fuzzy set, however is in the interval of [0,1] as illustrated in Fig. 4.4 as drawn by Kaufmann[6]. A fuzzy subset $\mathbf{A} \subset \mathbf{R}$ is considered normal if and only if:

$$\exists x \in \mathbf{R} \ni \mu_A(x) = 1. \tag{4.10}$$

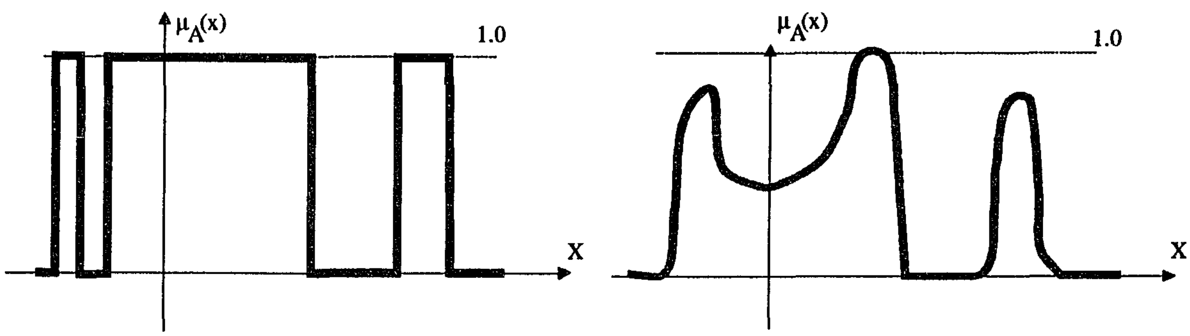


Fig. 4.4a An ordinary subset in **R**.

24.4b A fuzzy subset in **R**.

4.1.2.1. Addition of Convex Fuzzy Numbers

The addition of fuzzy numbers follows the same process as adding intervals of confidence (see equation 4.4), but level by level. If **A** and **B** are two fuzzy numbers and A_α and B_α are their interval of confidence for the level of presumption $\alpha \in [0,1]$. We can then define:

$$A_\alpha(+)B_\alpha = [a_1^{(\alpha)}, a_2^{(\alpha)}] \,(+)\, [b_1^{(\alpha)}, b_2^{(\alpha)}] = [a_1^{(\alpha)} + b_1^{(\alpha)}, a_2^{(\alpha)} + b_2^{(\alpha)}]. \qquad (4.11)$$

where,

$A_\alpha = \{x \mid \mu_A(x) \geq \alpha\}$, and

$B_\alpha = \{x \mid \mu_B(x) \geq \alpha\}$

The pictorial description of the addition of two fuzzy numbers is illustrated in Fig. 4.5. In this case, both fuzzy numbers are a triangular fuzzy numbers; that is, the interval of confidence is linearly and symmetrically lessen as the level of presumption gets larger.

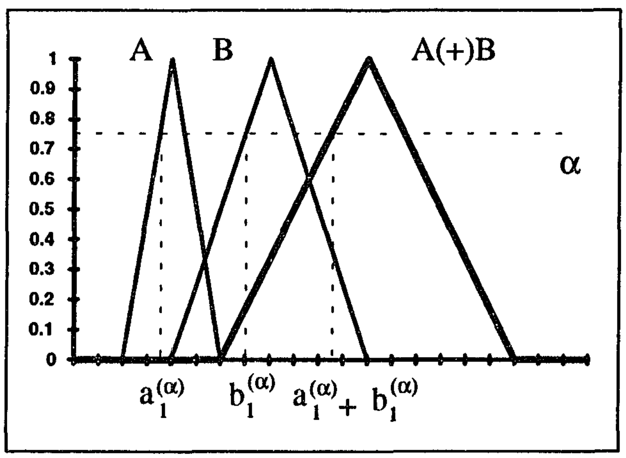


Fig. 4.5 Addition of two fuzzy numbers from [6].

4.1.2.2. Subtraction of Convex Fuzzy Numbers

The subtraction operation of fuzzy numbers is the extension of the addition operation and is similar to the addition of interval of confidence in equation 4.5. If **A** and **B** are two fuzzy numbers and A_α and B_α are their intervals of confidence for the level of presumption $\alpha \in [0,1]$. We can then define:

$$A_\alpha(-)B_\alpha = [a_1^{(\alpha)}, a_2^{(\alpha)}]\ (-)\ [b_1^{(\alpha)}, b_2^{(\alpha)}] = [a_1^{(\alpha)} - b_2^{(\alpha)}, a_2^{(\alpha)} - b_1^{(\alpha)}]. \qquad (4.12)$$

where,

$A_\alpha = \{x \mid \mu_A(x) \geq \alpha\}$, and

$B_\alpha = \{x \mid \mu_B(x) \geq \alpha\}$

Subtraction is, in fact, the addition of the image of $\mathbf{B}^-$ to **A**, where

$\forall \alpha \in [0,1], \quad B^-_\alpha = [-b_2^{(\alpha)}, -b_1^{(\alpha)}]$.

A pictorial description of the subtraction of **A** and **B** is illustrated in Fig. 4.6.

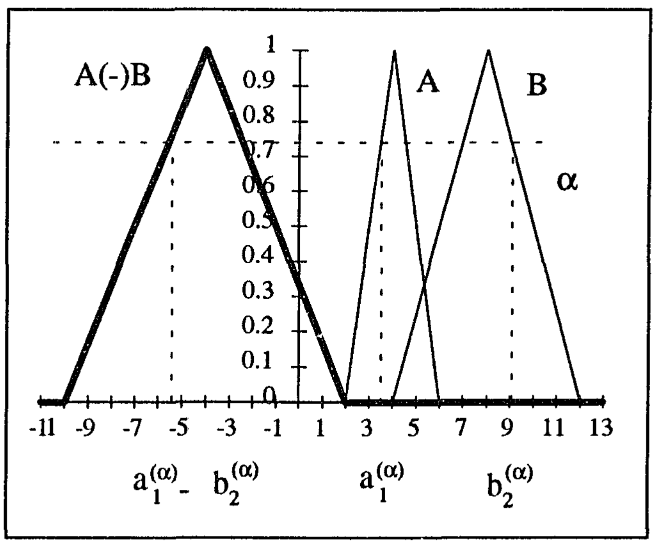


Fig. 4.6 Subtraction of two fuzzy numbers from [6].

Fuzzy numbers could be represented as a discrete combination of level of presumption and the confidence interval. The addition, as given by Kaufmann[6], is depicted in Fig. 4.6 could be numerically represented as following:

Table 4.1 Discrete subtraction of Fuzzy numbers.

	2	3	4	5	6
1.0			1		
0.75			1		
0.5		1	1	1	
0.25		1	1	1	
0.0	1	1	1	1	1
	0	0.5	1	0.5	0

A

	4	5	6	7	8	9	10	11	12
1.0					1				
0.75				1	1	1			
0.5			1	1	1	1	1		
0.25		1	1	1	1	1	1	1	
0.0	1	1	1	1	1	1	1	1	1
	0	0.25	0.5	0.75	1	0.75	0.5	0.25	0

B

	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2
1.0							1						
0.75						1	1	1					
0.5				1	1	1	1	1	1	1			
0.25			1	1	1	1	1	1	1	1	1		
0.0	1	1	1	1	11	1	1	1	1	1	1	1	1
	0	0	0.25	0.5	0.5	0.75	1	0.75	0.5	0.5	0.25	0	0

A - B

4.1.2.3. Fuzzy Minimum and Maximum

Consider **A** and **B** to be fuzzy numbers in **R**,

$$A_\alpha = [a_1^{(\alpha)}, a_2^{(\alpha)}]$$

and

$$B_\alpha = [b_1^{(\alpha)}, b_2^{(\alpha)}]$$

then if

$\forall\alpha \in [0.1]$ $a_1^{(\alpha)} \le b_1^{(\alpha)}$ and $a_2^{(\alpha)} \le b_2^{(\alpha)}$ then we can write $\mathbf{A} \le \mathbf{B}$.

The definition of the fuzzy minimum of **A** and **B** is defined in equation 4.13.

$$\forall\alpha \in [0.1] \quad MIN(A_\alpha, B_\alpha) = MIN\,([a_1^{(\alpha)}, a_2^{(\alpha)}],[b_1^{(\alpha)}, b_2^{(\alpha)}]) = \qquad (4.13)$$

$$= [MIN(a_1^{(\alpha)}, b_1^{(\alpha)}), MIN(a_2^{(\alpha)}, b_2^{(\alpha)})].$$

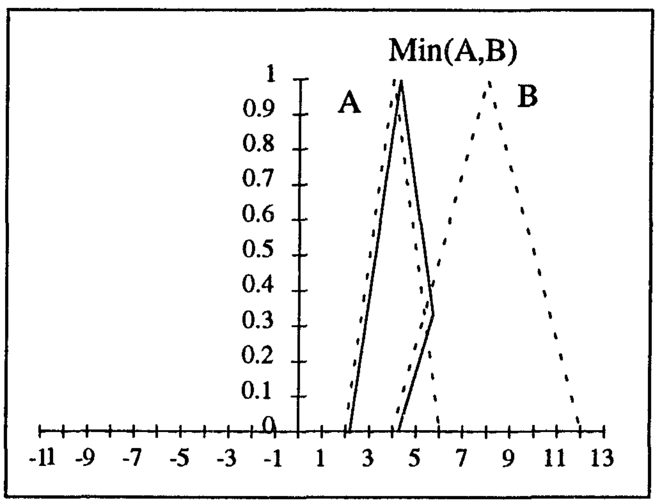


Fig. 4.7 Minimum of two fuzzy numbers from [6].

The *MAX* operation is computed by inverting the condition of *4.13* into the maximum condition.

4.1.3. Triangular Fuzzy Numbers (T.F.N.s)

A triangular fuzzy number (TFN) can be defined by a triplet (a_1,a_2,a_3) whose characteristic function is defined as following:

$$\mu_A(x) = \begin{cases} 0, & x < a_1 \\ \dfrac{x-a_1}{a_2-a_1}, & a_1 \le x \le a_2 \\ \dfrac{a_3-x}{a_3-a_2}, & a_2 \le xa_3 \\ 0, & x > a_3 \end{cases} \tag{4.14}$$

The fuzzy number **A** depicted in Fig. 4.7 could be represented as **A** = (2,4,6). A TFN is closed under addition, subtraction, multiplication, and division. Although the multiplication of T.F.N.s does not result in a straight line function, but rather a parabola, it could be approximated by a TFN as illustrated in Fig. 4.8[5].

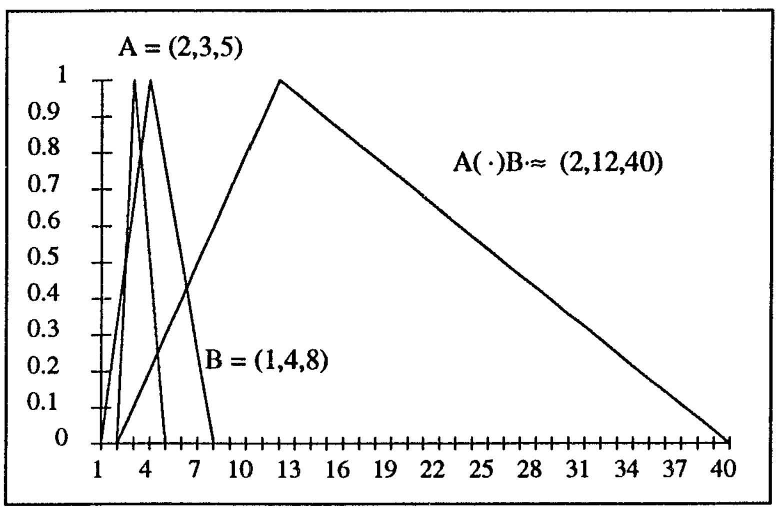


Fig. 4.8 Multiplication approximation of two T.F.N.s from [6].

4.2. Neural Networks

Neural networks and neural engineering in the wide sense are among the most rapidly developing scientific fields today. These fields are interdisciplinary in its nature, and the success formula is to combine expertise and experience from several scientific fields[63]. Artificial intelligence has not been able to deliver on many of its anticipated promises. Since AI representations are, in general, formal and precise, adding *common sense* or intuitive thinking has proven to be a dominant problem. The search for efficient computational approaches to artificial intelligence and cognitive engineering has evolved into a reasoning system comprised of parallel connectionist neuroprocessing rather than discrete symbolic reasoning. Neural networks are rough models of the mental processes, as their name implies. Because of their massive parallelism they can process information and carry out solutions almost simultaneously. They learn by being shown examples and the expected result, or they can form their own associations without being prompted and rewarded. They are good at pattern-matching types of problems. These characteristics can thus address many of today's problems.

Although neural networks show a great potential in solving many intractable AI problems, the implementation of a neural net architecture are constrained by a wide range of limitations. Some of the reasons for the limited technical success in emulating some of the more fundamental aspects of human intelligence lies in the differences between the organization, structuring of knowledge, and the dynamics of biological neuronal circuitry. Thus, we have but emulation using *the symbolic processing paradigm* [10,24,38]. Since it has been widely hypothesized that "analogy and reminding guide all our thought patterns and that being attuned to vague resemblance is the hallmark of intelligence," it would be naive to expect that logical manipulation of

symbolic descriptions is an adequate tool[36]. Also, there is substantial psycho physical evidence that while the beginner learns through rules, the expert discards such rules. Instead, the expert discriminates among thousands of patterns in his domain of expertise and learns how to respond to them through experience[15].

4.2.1. Background

Neural network technology is different from any of the other conventional computing methodologies. The information laid up in neural nets is not stored as in the conventional storage schema. Knowledge is distributed throughout the network. The network output is the dynamic response to the combination of stimuli input and network's architecture. The work of McCulloch and Pitts mark the beginning of neural networks development, as they have demonstrated that synchronous neural nets could perform all quantifiable processes[57]. The characteristics of a neural network is influenced by the biological neurons (see Fig. 4.9). It is based on a distributed computational system which comprises large number of nodes (neurons) each with a selective connection to other nodes creating interconnected structure similar to a directed graph of verying configuration. Hebb showed that when neurons activate one another through a particular synapse, it leads to strongly connected assemblies of neuron groups[37].

A neural network's associative memory properties makes it an important tool in pattern recognition, pattern reconstruction, adaptability fault tolerance, and resistance to fuzzy or noisy input. In order to have a network display its intelligent behavior, a neural network must be arranged to accommodate its purpose. This means that the correct connections must be made, and the connections must have the proper weight.

Because of the size and complexity of a network, the connections and weights cannot be pre-set. A network must learn or self-organize into a topology which exhibits the desired characteristics.

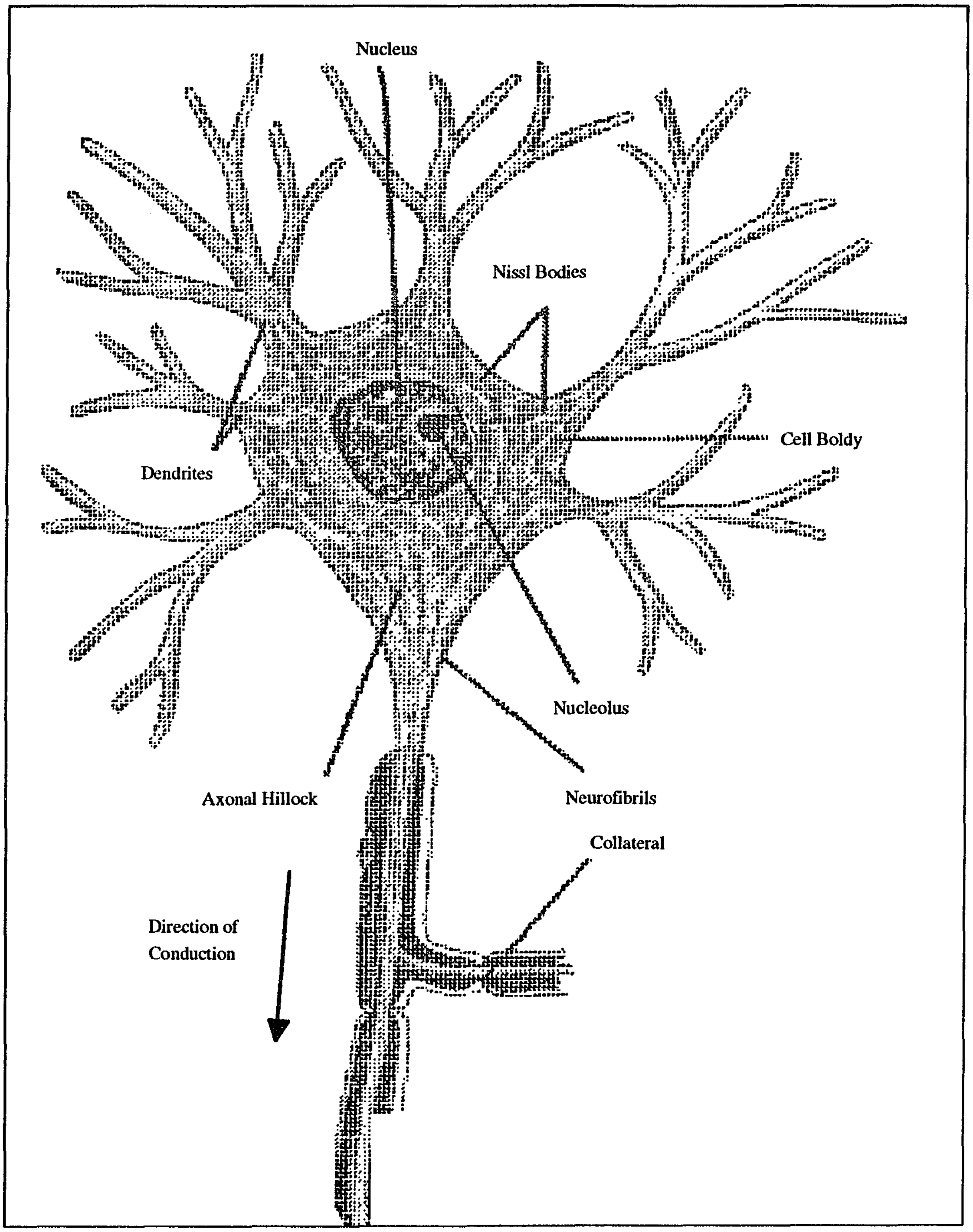


Fig. 4.9 The structure of a motor neuron from [28].

Fig. 4.10 illustrates the structure of an artificial neural system. The input to a neuron is a collection of outputs of other neurons, and some error correction feedback. The stimuli of a particular neuron to an input from a different neuron depends on the connection weight, $W_{j,i}$ (synapse connection). The output of the neuron is contrived by an activation function, *f*, on the summation of the inputted stimuli.

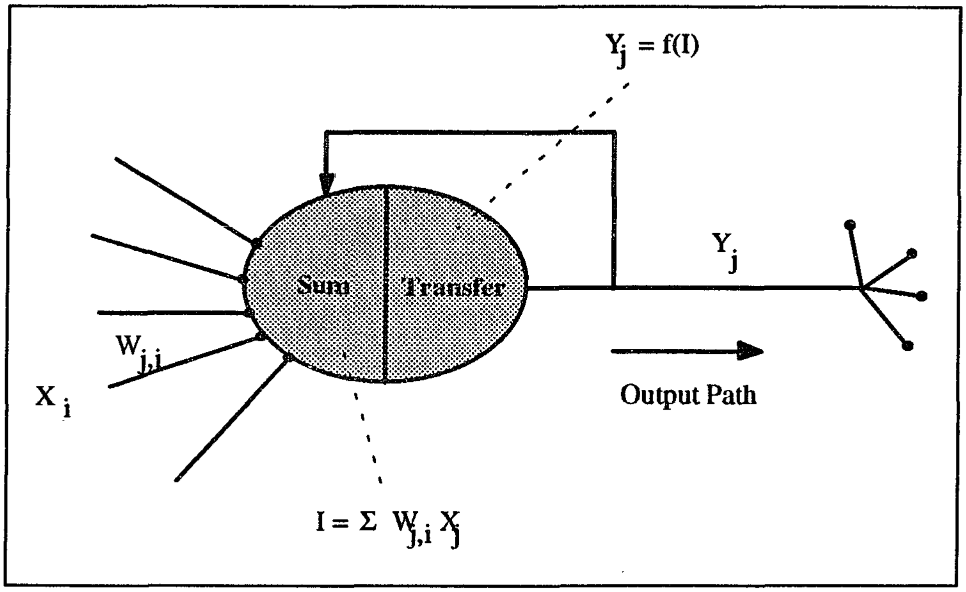


Fig. 4.10 Artificial motor neuron.

The network depicted in Fig. 4.11, for example, computes the summation of two binary inputs. Feeding a binary stimulus inputs **A** and **B** will induce a result based on the binary addition property. The system does not actually *compute* the summation of the input. The resulted output is determined by way of the connection weight between the neurons, as well as the schema in which the neurons interconnect with each other. Learning can be defined as a change in behavior, which is to a significant degree permanent in nature and which results from activity, training or observation[1]. In general, learning rules are based on the theory developed by Donald Hebb: If two

units are simultaneously active on, increases the strength of the connection between them.

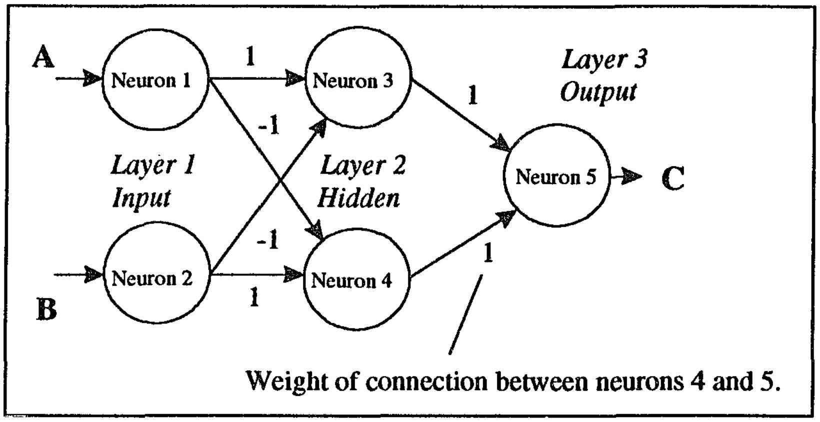


Fig. 4.11 A neural net which computes the summation of the binary inputs A and B. The threshold function results in 1 if the combined inputs are greater then 1, and 0 otherwise.

For instance, if one wishes to store the set of *m* patterns each represented by a vector **V** then for each $W_{i,j}$ (the synoptic connection between the i^{th} and j^{th} neurons) set.

$$W_{i,j} = \Sigma\, \mathbf{V}_i\, \mathbf{V}_j^T \qquad (4.15)$$

A variation of the Hebbian learning is the *delta rule.* The updating of the synoptic weight matrix is done according to the following iterative rule:

$$W_{i,j}(t+1) = W_{i,j}(t) + c[V_i(t)\text{-}O_i(t)]V_j(t) \qquad (4.16)$$

where t is the iteration number, c is a learning constant, and O_i is the output of neuron_i. The delta rule may require some iterative repetition. Also, it nullifies the symmetric property of the synoptic matrix and decreases the size of the attraction basin of the information embodied within the network. Still, this rule enable a network to increase the amount of information it shoud be able to absorb.

In the McCulloch-Pitts model (feedback network), each neuron performs a thresholding function. It hard-limits the output of a neuron to +1 or -1 by comparing the weighted sum $\Sigma_i T_{ij} O_i$ to its threshold, which is zero. In the feedforward network (as in Fig. 4.11) the use of hard limiting creates problems because in the hidden level information which come from the previous layer may be lost by this approach. The solution for the feedforward network is to emplcy an activation function, *f*, that is continuous, non-decreasing, differentiable, such that the output of a neuron $f(\Sigma_i T_{ij} O_i)$ is a real value in the range (0,1). The most frequently used activation function is the sigmoid function defined in 4.17.

$$O_j = \frac{1}{1+e^{-(\sum W_{i,j} O_i + \Phi_i)}} \qquad (4.17)$$

were ϕ_i is the bias of neuron_i.

The delta rule for the above activation function and a feed forward neural network would be computed recursively by computing the change in the synoptic weights from the output layers toward the inner layers as following :

Outer layer: $\Delta W_{i,j} = \eta \delta_i O_i$ *(4.18)*

and $\delta_i = (V_i - O_i) O_i (1 - O_i)$

Inner Layer: $\Delta W_{i,j}(n) = \eta \delta_i O_i + \alpha \Delta W_{i,j}\,(n)$ (4.19)

and $\delta_i = O_i (1 - O_i) \Sigma \delta_k\, W_{ik}$

for k being the index of a neuron in the previous layer. In addition, it is recommended that the activation function's bias, ϕ, would be learned as a regular weight as a unit that is always on.

So far hardware implementations of neural networks are primarily an explorations of various design possibilities[63]. The optimal solution in designing a neural network cheep depends on the application and on the system in which the network is integrated. This is since the complexity of the interconnections is inversely proportioned to the network's size. High resolution learning capability will necessitate a large interconnected neural system which will not fit onto a chip. Since pattern recognition applications requires large numbers of interconnections, they could be integrated onto chips only if they are very small and therefore quite simple. Paolo and Veljko maintain that [63]:

> *"So far, hardware designers have primarily been trying to build circuits based on theorists' models that were in turn inspired by neurobiology. It is crucial for this field that theorists and hardware developers work closely together and that theoretical models are not only inspired by biological wetware but also take into account the limitations of the electronic hardware."*

4.2.2. Neocognition

In 1975, Kunihiko Fukushima, a Japanese scientist presented a self organizing, multi-layer neural network named the Neocognitron,. The Neocognitron is a further development of a network which uses a supervised learning method[46]. The Neo-cognitron is a unique network because it has been developed for one specific purpose,

the recognition of handwritten characters. The network is modeled after the structure of the cognitron. All the neural elements employed in the cognitron are of the analog type. Each cell has many inputs and on output; these take non-negative analog values proportional to the pulses densities. The cognitron is based on two types of cells, excitatory cells and inhibitory cells.

The excitatory cell is an analog cell with a shunting inhibition mechanism. If u_i is an input from excitatory afferent synapses where i $\in$ $[1,n]$ and n is the number of inputs and v_j is an input from inhibitory afferent synapses where j $\in$ $[1,m]$ and m is the number of inputs from the inhibitory synapses, then the output , O, of that neuron will be defined as following:

$$O = \varphi\left[\frac{1+\sum_{v=1}^{n} a_v u_v}{1+\sum_{u=1}^{m} b_u v_u} - 1\right] = \varphi\left[\frac{\sum_{v=1}^{n} a_v u_v - \sum_{u=1}^{m} b_u v_u}{1+\sum_{u=1}^{m} b_u v_u}\right] \qquad (4.20)$$

were,

$$\varphi(x) = \begin{cases} x & x \geq 0 \\ 0 & \textit{eitherwise} \end{cases}$$

and, the synapses weights, a_v and b_u take non-negative values.

The cognitron network is constructed with a cascade of similar structured neural layers. The l^{th} layer consists both excitatory cells and inhibitory cells as illustrated in Fig. 4.11.

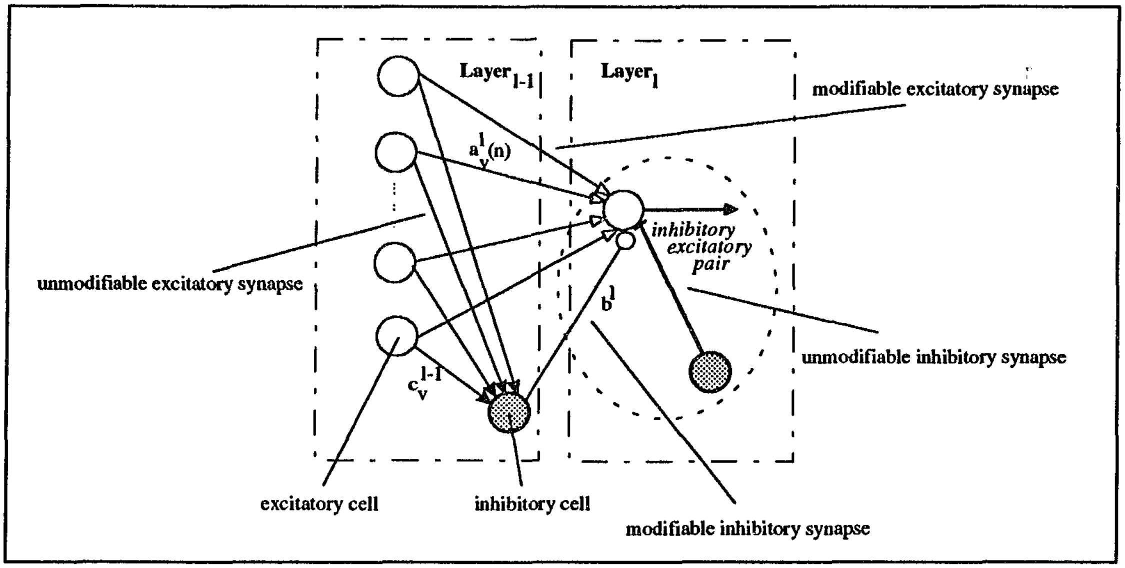


Fig. 4.12 The interconnection structure of the cognitron.

The inhibitory cell receives fixed excitatory synaptic connection such that,

$$\sum c_v^{l-1} = 1$$

Reinforcement of the afferent synapses to a cell u^l_n take place only when none of the cells situated in its vacinity is firing more strongly then u^l_n (see Fig. 4.13). The reinforcement amount to the input synapses is defined as following:

$$\Delta a_v^l(n) = qc_v^{l-1}u_{v+n}^{l-1}\delta(n), \qquad (4.21)$$

$$\Delta b^l(n) = \frac{\sum a_v^l(n)u_{v+n}^{l-1}}{2v^{l-1}(n)}$$

and,

$$\delta^l(n) = \begin{cases} 1 & u^l(n) > u^l(n+v) \\ 0 & \textit{eitherwise} \end{cases}$$ where, $v \in \Omega^l$, the competition area.

and where q *is a positive learning constant.*

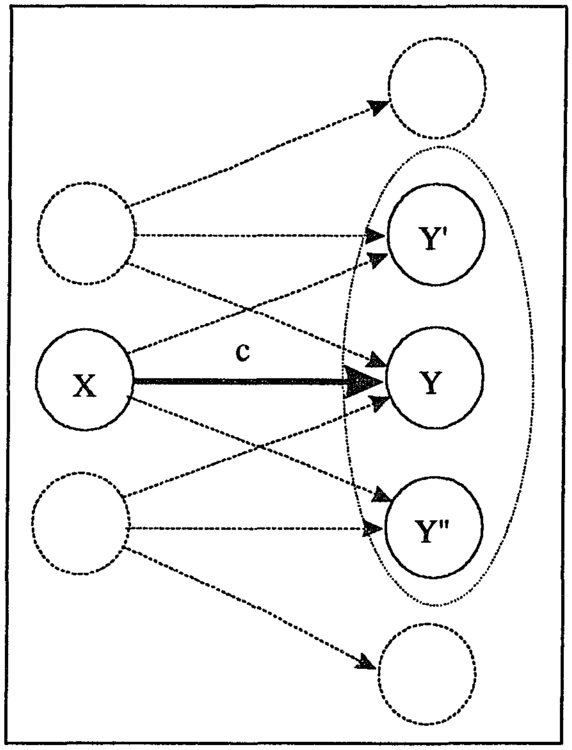


Fig. 4.13 Modifiable synapse c is reinforced if and only if Y = max(Y,Y',Y") and X > 0.

The Neocognition network suggested by Fukushima et al, is based on the network composition of modular structures of two layer of cells, namely, a layer U_S consisting of *S* cells, and a layer U_C consisting of *C* cells[46]. An *S* cell has a response characteristic similar to a simple cell or a lower order hyper complex cell according to the classification by Hubel and Wiesel[20]. The *C* cells, however, resembles a complex cell or a higher order hyper complex cells. Within the network, a cell in a higher stage generally has a tendency to respond selectively to a more complicated feature of the stimulus pattern and at the same time has a larger receptive field and is more insensitive to the shift in position of the stimulus pattern. In the Neocognitron S cells recognize specific features in a pattern, while the C cells make the network tolerant to shifts in position (see Fig. 4.14). Each C cell has afferent synapses leading from a group of *S* cells which have receptive fields of similar characteristics at approximately the same position on the input layer. This means that all of the pre synaptic *S* cells are to extract

almost the same stimulus feature but from slightly different positions on the input layer. The efficiencies of the synapses are determined in such a way that the C cell will be activated whenever at least one of its pre synaptic S cells is active. Hence, even if a stimulus pattern which has elicited a large response from the C cell is shifted a little in position, the C cell will keep responding as before because another pre synaptic S cell will become active instead of the first one.

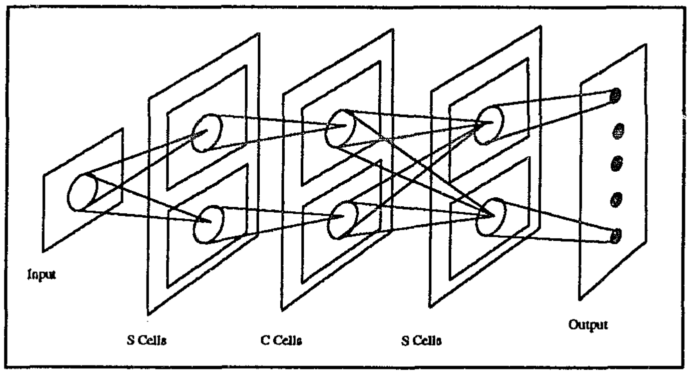


Fig. 4.14 The structure of a Neocognitron network

With the algorithms suggested by Fukushima et al [45], in each of the levels a small amount of positional error is tolerated. The operation by which positional errors are tolerated, little by little, plays an important role in endowing the network with an ability to recognize even distorted patterns. When a stimulus pattern is presented at a different position or is distorted in shape the responses of the cells in intermediate layers, especially the ones near the input layer, vary with the shift in position or the deformation of the stimulus pattern. However, the deeper the layer is the smaller become the variations in the response of the cells. Thus, the response of the cells in the deepest layer is entirely unaffected by shifts in position or by deformations of the stimulus pattern. He also showed some computer simulation to demonstrate the response of the cells of layers to pattern stimuli of numerical patterns.

4.2.3. Linear Learning

A linear mapping of an n space X×X×X×X×...×X into Y×Y×Y×Y×...×Y could be achieved by a feed forward network implementations that does not employ any hidden layers. Learning, then, could be learned using a pseudoinverse technique. Extensions of those techniques correspond to simple multilayer networks[63,62]. By operating on a suitable enlarged input space containing relevant (not necessarily linear) features of the problem, a linear system can in fact approximate nonlinear mapping. This approximation, however, must compute an optimal linear estimators of relatively large dimensional input spaces. As an example, consider the "retinex" algorithm. The retinex lightness algorithms assume that the visual system performs the same computation in each of three independent chromatic channels[26]. The algorithms assume that in each channel the image intensity signal (irradiance), s', is proportional to the product of the illumination intensity e' and the surface spectral reflectance r' in that channel:

$$s'(x,y) = r'(x,y)e'(x,y) \qquad (4.22a)$$

Taking the logarithms of both sides converts it to a simple sum:

$$s(x,y) = r(x,y) + e(x,y) \qquad (4.22b)$$

where $s = log(s')$, $r = log(r')$ and $e = log(e')$.

Retinex algorithms seek to solve the first of these equations for *lightness*, which is an approximation to $r'(x,y)$ in each channel. The resulting triplet of lightness labels a

constant color in a color space. To make a solution possible, retinex algorithms restrict their domain to a world of Mondrians, two-dimensional surfaces covered with patches of random colors. The algorithms then disentangle lightness from illumination by exploiting two assumptions[63]:

(i) $r'(x,y)$ is uniform within patches but has sharp discontinuities at edges between patches.

(ii) $e'(x,y)$ varies smoothly across the Mondrian.

This type of mapping could be used to refine the fuzzy recognition discussed in section 4.3.4.

4.3. Actual Recognition

Several high accuracy algorithms have been proposed for recognition of handwritten numerals recently[42,69,29]. Most techniques consists of a sequential combination of a fast structural classifier and robust relaxation algorithms. The classification is based on the configuration of a set of primitives derived from the images of numerals. Although very low error rates are realized, the method is relatively slow owing to an extensive preprocessing of the numeral image prior to feature extraction and the complexity of the relaxation algorithms[29]. Most of these algorithms make use of a combination of statistical classification (utilizing histograms and direction vectors derived from the contours of the character), template matching techniques, and structural classification of features such as size and stroke placement.

Typical requirements of a commercial numeral recognition system are:

- the recognition system is writer-independent.
- the system can operate at high speeds for commercial application.
- the recognition system will have low error rates and low rejection rates.
- the recognition system is robust in the presence of noise or varying background.

Complying with these requirements generally produces a system strategy based on the use of multiple recognition algorithms, controlling the error rates for a specific application by simply adjusting some rejection thresholds[29].

The approach taken in this dissertation is to design a system that is best suited to the recognition of the complete set of alphanumeric characters (rather than the subset of numerals) with digital computers. Although statistical analysis, in general,

results in accurate recognition system, it requires thorough use of a complex computation analysis which degrade the system's speed dramatically. The fundamental guideline has been to transform the image information into specialized data schema best suited for processing by digital computers. Therefore, the recognition analysis is transformed into a simple robust computation resolution.

4.3.1. Background

The recognition system developed here is based on the combination of fuzzy logic and neural networks. Although neural networks best utilize parallel computations and best suited to overcome information fuzziness, we still have great difficulties in controlling the network's respond to specific input stimuli. This is because once the network's architecture and its learning methodology are developed, one is not able to influence a particular attraction basin without modifying the nature of the network's entire characteristics. In addition, it is evident that local minimas could influence the system's characteristics, an uncertainty that can not be overlooked.

4.3.1.1. ANN Local Minima Problems

In order to demonstrate the local minima problem with artificial neural networks consider, as an example, the network depicted in Fig. 4.15 given by Rumelhar et al [23],whose function is to copy the value of the input neuron unit to the output unit.

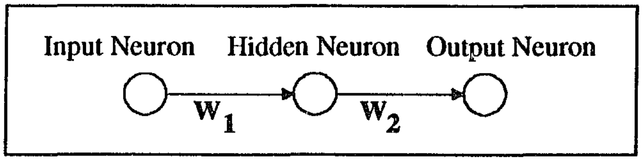


Fig. 4.15 Feed forward network consisting of a single input, hidden, and output neurons.

There are two basic solutions. The system may have positive biases on the hidden unit and on the output unit and large negative connections from the input unit to the hidden unit and from the hidden unit to the output unit, or it can have large negative biases on the two units and large positive weights from the input unit to the hidden unit and from the hidden unit to the output unit[23]. These solutions are illustrated in Table 4.2.

Table 4.2 Weights and biases of the solutions for copying the input into the output from [23].

Minima	**W_1**	**W_2**	**$Bias_1$**	**$Bias_2$**
Global	-8	-8	+4	+4
Global	+8	+8	-4	-4
Global	-8	-8	0	0
Local	+8	+0.73	0	0

In the first case, feeding a '0' will cause the input of the hidden neuron to have a 0 as input plus the bias, producing more then a 0.5 value (using a sigmoid function) and thus turning the hidden neuron on. The output neuron will receive a large negative value of input and thus the output . If we feed a '1' to the network, the hidden neuron will receive a '-8' as input, turning it off. A '0' as an input plus the bias will cause the output neuron to be turned on.

The second case is a symmetric complement of the first case. Thus it appears that we have two symmetric solutions to the problem resulting in global minima since

they have the same error. If we are not allowing the biases to change, however, we will have the third and forth solutions. As one can see, the error in the third case is 0.5 (and thus the squared error is 0.25). It turns out that this is the best the system could do with zero biases. The second solution has a squared error of about 0.45 (0.67 error) — a local minima. Fig. 4.16 depicts the contour map for the error space of this problem. It is easy to see that if the system will start at a point below the line $W_1 + W_2 = 0$, the system will follow the contours down and to the left into the minimum in which both weights are negative. If, however, the system begins above the anti-diagonal, the system will follow the slope into the upper right quadrant in which both weights are positive. Eventually, the system moves into a a gently sloping valley in which the weight from the hidden unit to the output unit is almost constant at about 0.73 and the weight from the input unit to the hidden unit is slowly increasing. It is slowly being sucked into a local minimum. There is no way the system can sense the depth of a minimum from an edge's minimum, and once it has slipped in a particular direction, there is no way out.

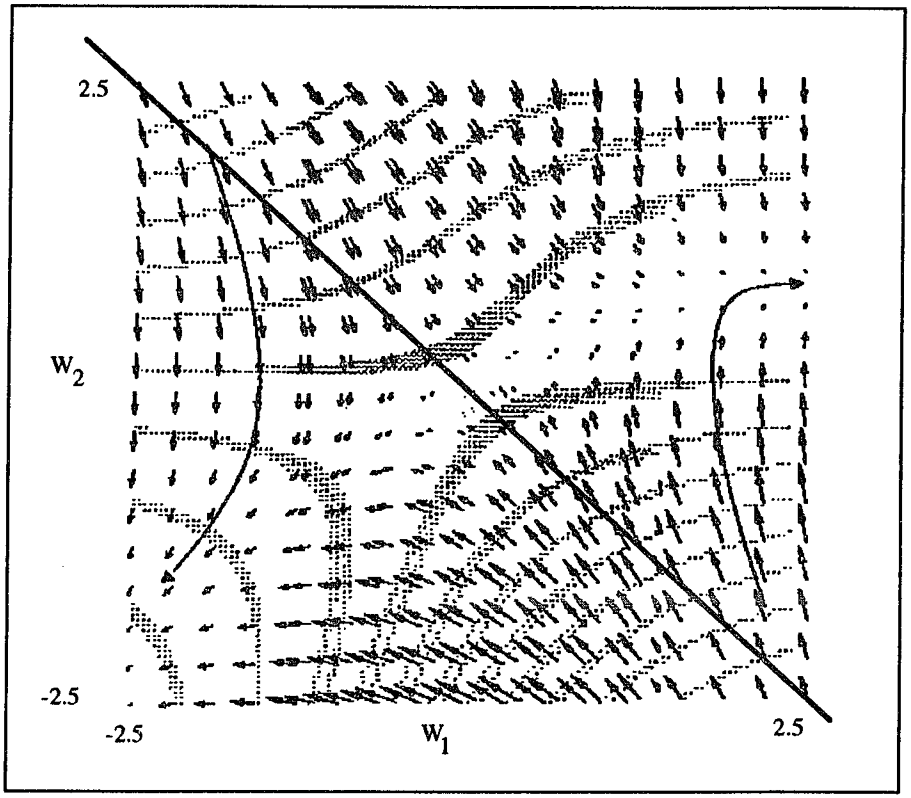


Fig. 4.16 A contour map for the identity problem with biases fixed at 0. from [23].

4.3.1.2. Symmetry Problem of Back Propagation

If all the synaptic weights are initialized to the same value, and the solution requires that unequal weights be developed, the system could never learn the proper solution. This is because the propagated error is in proportion to the values currently employed. This means that all hidden units connected directly to the output units will get identical error signals, and since the weight changes depend on the error signals, the weights from those units to the output units must always be the same. The system is starting out as a kind of unstable equilibrium point that keeps the weights equal, but it is higher than some neighboring points on the error surface, and once it moves away to one of these points, it will never return. Thus, one must initialized the weights into different random values.

4.3.1.3. The Bias Problem in Back Propagation

The identity network illustrated in Fig. 4.15 above shows that if the biases are set to 0, there is an error of 0.5 (squared error of 0.25). This, it turns out, is the best the system can do with zero biases. Thus, a system which will iterate in a search of a minima with error value of 0.01 will never stop. One must let the biases be changed for each neuron independently. This will increase the power of the network and enable to reach into a deeper minima.

4.3.1.4. Conclusion

Recognizing the English alphanumeric character set necessitates a system that could be easily modified its reaction to a particular information stimuli without effecting the system characteristic response to prevalent input stimuli. Therefore, we design a system in which the first generalization of input stimuli is achieved with fuzzy logic which then feeds its information into an artificial neural network (see Fig. 14.17).

After refining the original digital image by reducing noise and thinning the textual figures, the image is then partitioned into clusters; each maintaining the skeleton image of a single character. Each cluster, x_i, is then processed separately (in parallel) by first extracting a set of morphological attributes which form the principal information used in the recognition phase. The dynamic fuzzy system generate a set of probabilities for possible English characters that could fit the given set of morphological attributes. These probabilities are fed into a feedforward neural network to further generalized these probabilities into possible (if any) English character which correspond to the image depicted by x_i.

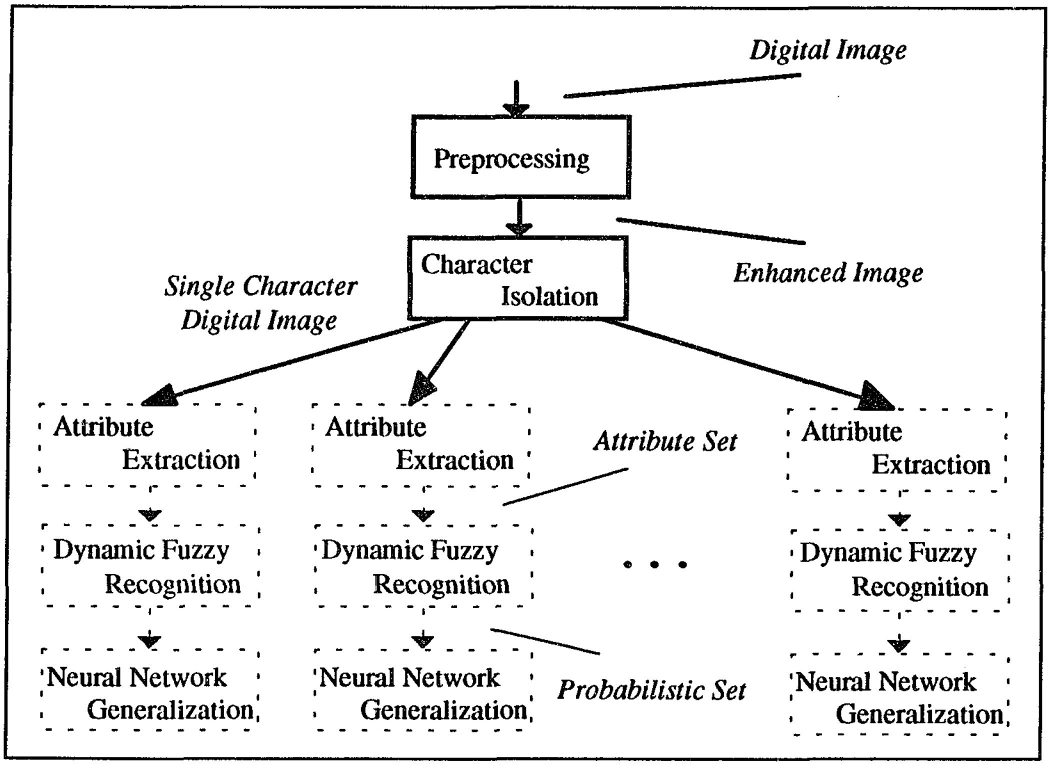


Fig. 4.17 OCR recognition system diagram.

4.3.2. Attribute Extraction

Template matching could not be used effectively in an OCR system that processes hand written alphanumeric characters. This is since the pattern's height and width or any other spatial deformation[4] should have no bearing on the recognition analysis. Therefore, one should maintain a set of n special morphological attributes, Ψ, where each attribute, ψ_i $(1 \leq i \leq n)$ is unique to a small subset of the alphabet (the English alphabet in our case), ϑ. The measurement of ψ_i in regard to a given digital image is represented by a triangular fuzzy number $\mathbf{A}^{\varphi_i} = (a_1, a_2, a_3)$. Extracting the set

4. Spatial differences between digital images of the same alphanumeric image could be the result of the scanner's resolution as well as a person's unique hand writing style.

of fuzzy numbers A^{φ_i} (for $\psi_i \in \Psi$) from a given digital pattern will then result in a unique *n* space response function. This response can then be associated with a given character in ϑ (learned) or compared with all response recognition functions of characters in ϑ. The characteristics of ψ_i depends on the unique features of the elements in ϑ. The attribute set represented by Ψ should be carefully defined in order to achieve significant deviation in the response among elements in ϑ. Examining the English alphabet figures (see appendix A), for example, reveals that the spatial characteristics of the English characters are quite methodical. Most characters could be easily partitioned into four quadrants which maintain a unique stroke configuration. Still, the meaning of *stroke configuration* is quite vague. Consider the digital figure of the character 'O' depicted in Fig. 4.18c. In trailing the left profile of the figure from top to bottom, one can see that at beginning the contour leads consistently left, whereas the bottom contour leads consistently right.

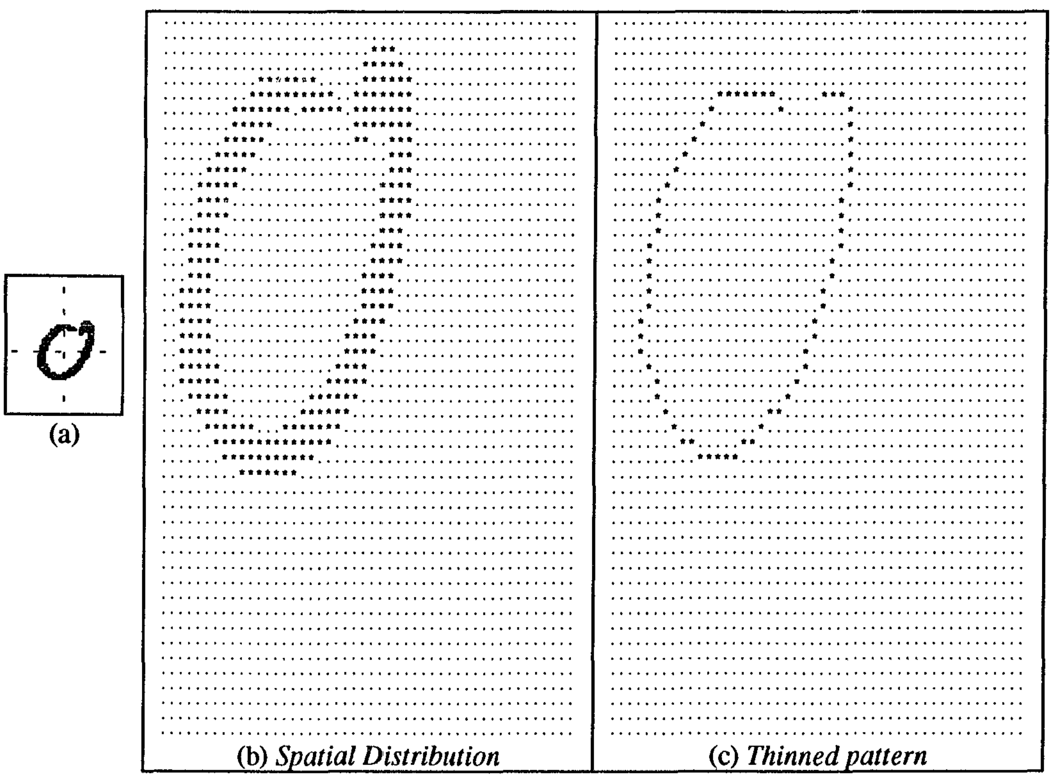


Fig. 4.18 Digital pattern of the character 'O.' (a) original, (b) pixel spatial distribution, (c) Image's skeleton.

In general, one could examine the contour for left transitions (if any) or right transitions when walking the pattern's contour from top to bottom. Since the pattern is a discrete distribution of pixels, one can represent this scenario mathematically by the ratio of the left directional movement, Δl, to the down directional movement Δd as a procedure that returns three evaluation as defined in equation 4.23.

$$\varphi(\Delta d, \Delta l) = \begin{cases} \begin{array}{c} R = \dfrac{\Delta l}{\Delta d} \\ Small = Left, Lz = No \end{array} & (\Delta l \leq \Delta d) \wedge (\Delta l > 0) \\ \begin{array}{c} R = \dfrac{\Delta d}{\Delta l} \\ Small = Right, Lz = No \end{array} & (\Delta d > \Delta l) \wedge (\Delta l > 0) \\ \begin{array}{c} R = 0 \\ Small = Left, Lz = Yes \end{array} & eitherwise \end{cases} \tag{4.23}$$

Where R is the computed ratio such that $0 \leq R \leq 1$, *Small* designate the Smallest of Δl, and Δd, and Lz designates whether Δl is zero. Note that this example assumes that $\Delta d > 0$; that is, the height of the image is greater then one pixel. Table 4.3 depicts ψ as it was evaluated from Fig. 4.18c.

Table 4.3 ψ's computation for the character 'O' illustrated in Fig. 4.18 for the left upper and lower quadrants.

	Δl	Δd	Δr	R	*Small*	*Lz*
Left Top Quadrant	12	14	0	0.86	Left	No
Left Lower Quadrant	1	12	11	0.08	Left	No

The ratio attribute, ψ, could be represented by A^{φ_u} (for the upper left quadrant) and A^{φ_l} (for the lower quadrant) using the schema illustrated in Fig. 4.19 where A^{φ_u}=

$(0_{left}, 0.86_{left}, 0_{down})$ and $A^{\varphi_l} = (0_{left}, 0.08_{left}, 0_{down})$. The *Lz* zero flag is needed to refine the recognition of similar response to particular attributes. The essence of ψ is to record the degree of left transitions which comprise the left contour. Zero transitions have different meaning depending on whether the height of the pattern is large or small. This is true because a small ratio of two values does not necessarily designate values that are significantly different in magnitude. Consider the following ratios:

$$\frac{0}{1}, and, \frac{5}{1,000,000,000}$$

Both are small ratios. Still, the first ratio is the result of magnitudes that are quite small and similar. Their semantics differ to that generated by the second ratio. Recording *Lz* could be useful in processing ψ. The *Lz* flag should be represented, however, as a triangular fuzzy number rather than a binary flag. This is because the semantics of zero transitions is a fuzzy concept as well.

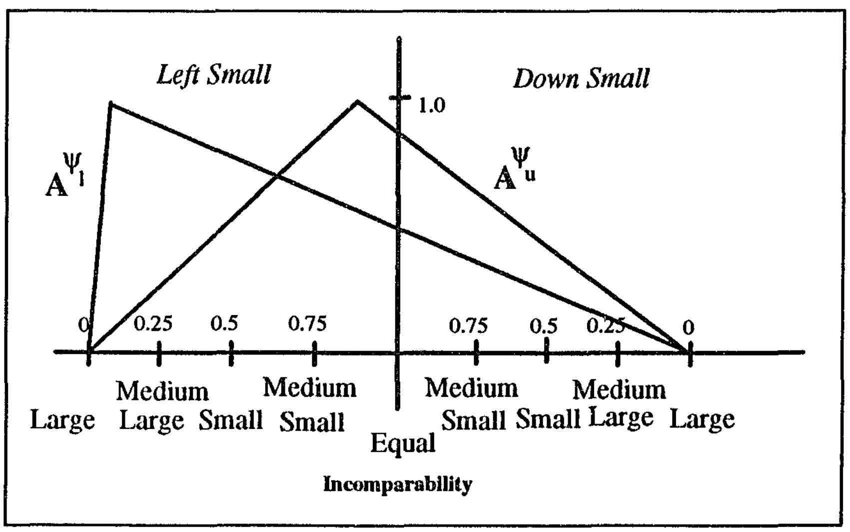


Fig. 4.19 A^{Ψ} fuzzy representation of two attributes in the upper and lower quadrants.

Most attributes defined in our implementation of the recognition system involve a comparative measurement of two image features such as the comparison of the

number of dark pixels on a squared frame (with pre-defined thickness) fitted on the image's contour to the number of pixels that are at the center of that image (deviating between the characters 'o' and 'e'). The attribute set Ψ was constructed to produce an overall unique response by each of the English alphabet. In a establishing a recognition threshold α (the level of presumption), one can measure the similarities in the patterns responses to Ψ by comparing their fuzzy responses $A^{\psi_i} \ni \psi_i \in \Psi$. This measurement is established by assigning a pre-defined constant weight, w^{ψ_i}, to each A^{ψ_i}. This weight correlate to the significance of ψ_i in Ψ.

Extracting $\psi_i \in \Psi$ is a simple computation that could be achieved in parallel. At that point the digital image is transformed into a set of fuzzy number representing specific attributes. By changing this representation schema, we reduce the recognition complexity substantially. This is because by defining $A^{\psi_i} \ni \psi_i \in \Psi$ we are not concerned with the height or width of the character or any other spatial distribution of pixels. From this point on (when an images response to Ψ is extracted), the digital image is discarded.

Comparable A^{ψ_i} designate images with similar attributes. The semantics of the attributes need not have any meaning to the recognition system. This is since ψ_i and w^{ψ_i} are established during the system's design phase. It is based on the same analogy given to the design of the Coke machine described in section 1.4. When the coin slides on the sequence of openings, it does not matter to the machine itself through which hole the coin is going to fall or why. These resolutions were made while designing the machine.

Maintaining an image's attribute information necessitate to store $A^{\psi_i} \ni \psi_i \in \Psi$ in correspondence to the alphanumeric symbols in ϑ that an image represents. Learning various digital patterns which correlate to the same character $a \in \vartheta$ require the modification of A^{ψ_i} as described in the next sections. Thus, retaining the

information regarding a character $a \in \vartheta$ requires O(|Ψ|) amount of space, where the cardinality of Ψ is a constant for a given system.

4.3.3. Attribute Learning

The OCR system should dynamically update its configuration to augment its knowledge as various pattern are introduced to it. The system has two modes: a learning mode and a recognition mode. In the learning mode, a given isolated cluster of the digital pattern (representing an isolated character), x_i, is designated with character $a \in \vartheta$. In this case the system should add the attribute information $\mathbf{A}^{\psi_i} \ni \psi_i \in \Psi$ (extracted from x_i) to its knowledge of a. The essence of the system's fuzzy knowledge database is to maintain the response of each character $a \in \vartheta$ to each of the attributes $\psi_i \in \Psi$; that is, to maintain an array of size $|\vartheta|$, where each of its elements (correlating to $a \in \vartheta$) is also an array of size $|\Psi|$. This last array's elements are fuzzy numbers correlating to a character's ($a \in \vartheta$). response to attribute $\psi_i \in \Psi$. This response is the result of previous responses to ψ_i that were extracted from past exposure to various digital patterns known to represent a. An attribute response is represented as a triangular fuzzy number $\mathbf{A}^{\psi_i} = (a_1, a_2, a_3) \ni \psi_i \in \Psi$. Still, when embedding a collection of fuzzy responses to an attribute ψ_i into fuzzy numbers representing the system fuzzy knowledge data base, they lose their triangular characteristics. Actually, the fuzzy numbers loose their convexity. However, they still maintain their normality ($\exists_x \mu_{\mathbf{A}^{\psi}}(x) = 1$). Consider, for example, the fuzzy numbers $\mathbf{A}^{\psi_l}$ and $\mathbf{A}^{\psi_u}$ from Fig. 4.19 are to be merged into $\mathbf{A}^{\psi_{merged}}$ such that $\mathbf{A}^{\psi_{merged}}$ responds favorably to both instances $\mathbf{A}^{\psi_l}$ and $\mathbf{A}^{\psi_u}$. This merger is illustrated in 4.20 and is defined in equation 4.24.

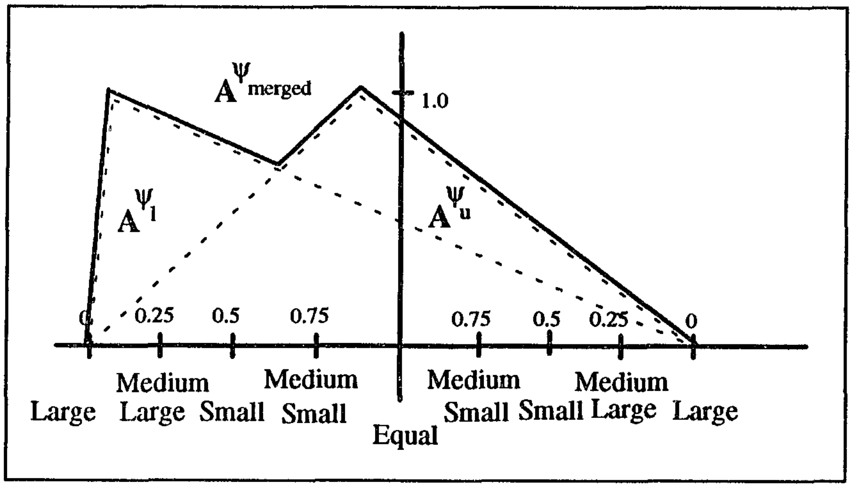


Fig. 4.20 Merging two fuzzy numbers to preserve favorable response to both

$$\mu_{A^{\psi_{merged}}}(x) = Max(\mu_{A^{\psi_l}}(x), \mu_{A^{\psi_u}}(x)) \qquad (4.24)$$

It is evident from equation 4.24 that learning the same pattern as $a \in \vartheta$ will not change the fuzzy knowledge base. Moreover, if an attribute $\psi_i \in \Psi$ is contradictory to a particular $a \in \vartheta$ (such as A^{ψ_l} and A^{ψ_u} in Fig. 4.20), the response of $A^{\psi_{merged}}$ will eventually become favorable to the entire attribute domain. This lessens the influence of ψ_i in discriminating a with various characters in ϑ. The knowledge base for each $a \in \vartheta$ is then a fuzzy response to all $\psi_i \in \Psi$. In essence, the recognition of $a \in \vartheta$ from a digital pattern is achieved when the pattern's attribute response fall within the envelope's peak response of the fuzzy knowledge base. Consider $|\Psi|=2$ such that A^{ψ_l} and A^{ψ_u} from Fig. 4.19 are the responses to the two attributes, ψ_l, and ψ_u that are defined for the English alphabet. Then the response of the character 'O' to Ψ is depicted in Fig. 4.21.

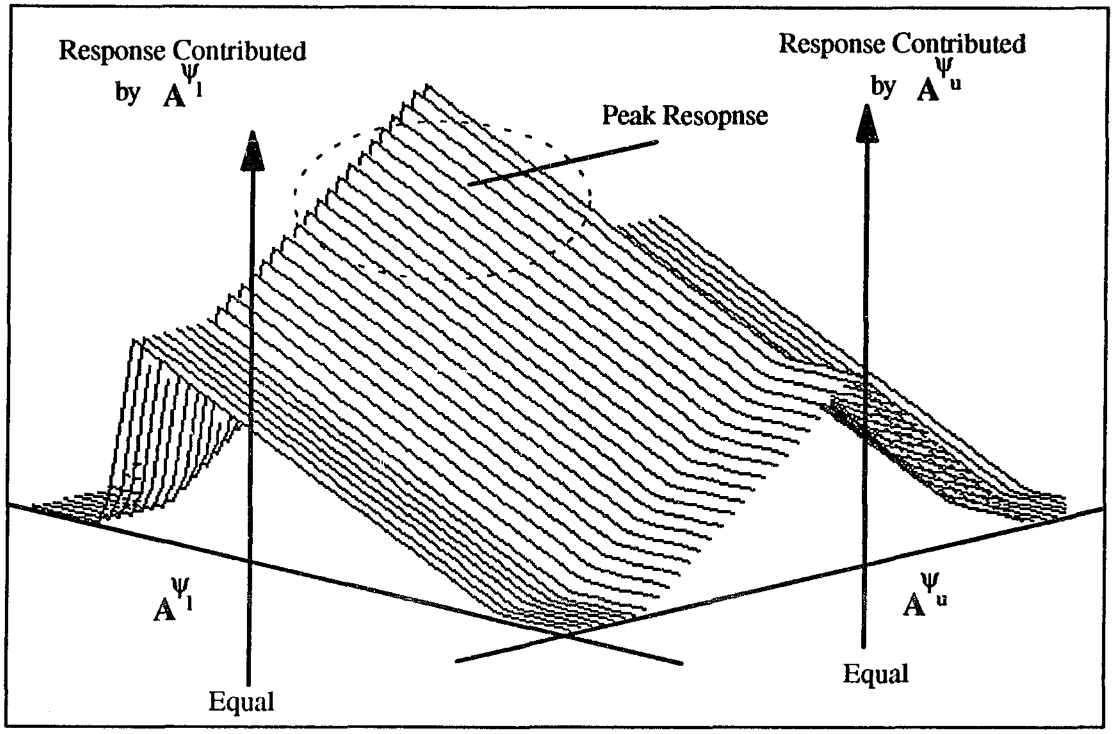


Fig. 4.21 'O' recognition response after learning A^{ψ_l} and A^{ψ_u}.

4.3.4. Attribute Recognition

The recognition system probes the response of a digital pattern to Ψ. In this recognition mode, triangular fuzzy numbers are extracted from the pattern in questions for each $\psi_i \in \Psi$ (as discussed in previous sections). These fuzzy numbers represent the response of the pattern to Ψ. This response function is then compared with the response of each element's $a \in \vartheta$ response to Ψ. These comparison, which can be done in parallel, results in a set of probabilities, p_i, assigned to each $a \in \vartheta$ and $1 \le i\ |\vartheta|$ (see equation 4.25). The likelihood that the pattern represents $a_c \in \vartheta$, is, then, represented by p_c:

$$p_c = \frac{\sum_{j=1}^{|\Psi|} w_j f(\mu_{A_c^{\psi_j}}(a_2^j))}{\sum_{j=1}^{|\psi|} w_j f(1)} \tag{4.25}$$

where w_j is a pre-defined weight assigned to $\psi_j \in \Psi$ representing its significance, the function, f, is some exponential function, a_2^j is the middle element of the pattern's fuzzy number $A_p = (a_1,a_2,a_3)$ representing the pattern's response to attribute ψ_j, and $A_c^{\psi_j}$ is the fuzzy knowledge response of character a_c to ψ_j. The introduction of function, f, sharply penalizes different attributes. Consider two attribute response a and b. Also assume that $|\Psi| = 4$, and all attributes have the same significance, w. Then in the case that the comparisons result in :

pattern a we have $\quad \forall_j \mu_{A_a^{\psi_j}}(a_2^j) \approx 0.75 \quad$ *for* $1 \le\ i \le\ 4$

pattern b we have $\quad \forall_j \mu_{A_b^{\psi_j}}(a_2^j) \approx 1 \quad$ *for* $1 \le\ i \le\ 3$, *and*

$$\mu_{A_b^{\psi_4}}(a_2^4) \approx 0.25$$

which mean that in the first comparison all attributes of the pattern a are quite favorable. Pattern b however, reveals that the first three attributes have high response. The forth attribute, however, is sharply incomparable. In both cases the average response is the same. Since we are dealing with uncertainty, the first comparison should be considered as a possible attribute match to Ψ. The second comparison implies that there is a major difference between the fourth feature of the digital pattern and the knowledge about ψ_4. Thus, employing f will guarantee that the system will favor the first case, a's response, as a possible match to Ψ.

Calculating p_c for each character in ϑ could be quite expensive, even in parallel. The fuzzy knowledge data base could be simplified further by representing the response function as a discrete numerical representation. The fuzzy number domains could be partitioned into n segments S_i for $1 \leq i \leq n$ (see Fig. 4.20). An average probability of this domain segment as defined in 4.26 is then maintained:

$$p_c = \frac{\sum_{j}^{|\Psi|} w_j f(\overline{\mu}_{A_c^{\psi_j}}(s_i))}{\sum_{j=1}^{|\Psi|} w_j f(1)} \tag{4.26}$$

where $\overline{\mu}$ be the average response of the fuzzy number A_c that represents character $c \in \vartheta$ response to ψ_j in the fuzzy knowledge base and S_i is the segment in which the pattern's response to ψ_j is 1 (where a_2 falls under).

These numerical representation could be stored within a three dimensional array, $A_{c,j,i}$, during the learning mode, where $1 \leq c \leq |\vartheta|$ and $1 \leq j \leq |\Psi|$ and $1 \leq i \leq |S|$ such that S is the segment set partitioning the domain of all fuzzy numbers. Thus the response level of each character m to attribute n in the kth segment is recorded in $A_{c,j,i}$.

Recognizing a digital pattern is then achieved as following: after preprocessing, attribute extraction from the digital image is performed we maintain the response of the pattern to Ψ. Rather than representing the response as a triangular fuzzy number **A** = (a_1,a_2,a_3), the segment index, i, under which a_2 falls is preserved. Thus, the image is transformed into a set of indexes $I = \{i_j \mid 0 \le i \le |S|,\ 1 \le j \le |\Psi|\}$ depicting the segments under which the pattern's respond to each attribute in Ψ, a_2, falls. Computing the probability that a digital pattern represents a character $c \in \vartheta$ is then computed as following:

$$p_c = \sum_{j=1}^{|\Psi|} A_{c,j,s_{p(a_2^j)}} \qquad (4.27a)$$

where $s_{p(a_2^j)}$ is the segment index under which the patterns fuzzy response of a_2 falls.

The fuzzy knowledge base A is updated as following:

$$A_{c,j,i} = Max[A_{c,j,i}, w_j \frac{f(\mu_{A_p^{\psi_j}}(s_i))}{f(1)}] \qquad (4.27b)$$

where c is the character index to ϑ, j is the index to attribute ψ_j and S_i is the segment average response extracted from the digital pattern's response to attribute ψ_j. Clearly the complexity of the recognition computation is $O(|\vartheta|)$ (in 4.27a), a constant for a given system. The time complexity is shifted to the learning phase (4.27b) to update A to reflect the fuzzy knowledge base.

Recognizing a digital pattern using 4.27a proves to be quite robust in our implementation. The recognition is based on a set of attributes each assigned a weight w_i denoting its significance regarding other attributes in Ψ. The recognition system could be refined further by allowing the weight's magnitude to change dynamically as

attribute information is gathered from various exposure to digital patterns. A linear transformation (see section 4.2.3) using a feedforward neural network will do just that. The transformation's input will be defined as following:

$$\mathbf{I} = \frac{f(\overline{\mu}_{A_c^{\psi_j}}(a_2^j))}{f(1)} = A'_{c,j,s_j} \qquad (4.28)$$

where $\mathbf{I}$ is the neural input and $A_c^{\psi_j}$ is the knowledge response of the fuzzy recognition response to attribute ψ_j and $0 \leq \mathbf{I} \leq 1$ and A' is the fuzzy knowledge base where learning is performed with w_j=1 for $1 \leq j \leq |\Psi|$ (defined in 4.27b).

If sufficient CPUs and resources are available for a more complex neural architecture to remain efficient, one could let the neural network emulate function f of 4.28 as well. In this case, the neural network's input will be defined as following :

$$\mathbf{I} = \overline{\mu}_{A_c^{\psi_j}}(a_2^j) \qquad (4.29)$$

The architecture of this neural network which performs a nonlinear generalization to f and w is illustrated in Fig. 4.22. The neural network is comprised of two feed forward networks with one hidden layer each. The input to the first layer is the fuzzy response of all attributes from all alphanumeric symbols in ϑ, in addition to the fuzzy response from the digital pattern as defined in 4.29. The first three levels are stimulated by favorable attribute responses both from the digital pattern and each of the alphanumeric symbols in ϑ. The last three layer, direct these stimulus into a set of $|\vartheta|$ output neurons. The output of these neurons represent the probabilities for each symbol in ϑ. Each probability P_c denotes the likelihood that $a_c \in \vartheta$ is depicted in the digital pattern

in question. The learning in this network is a back propagation delta learning ruled defined in equations 4.18 and 4.19.

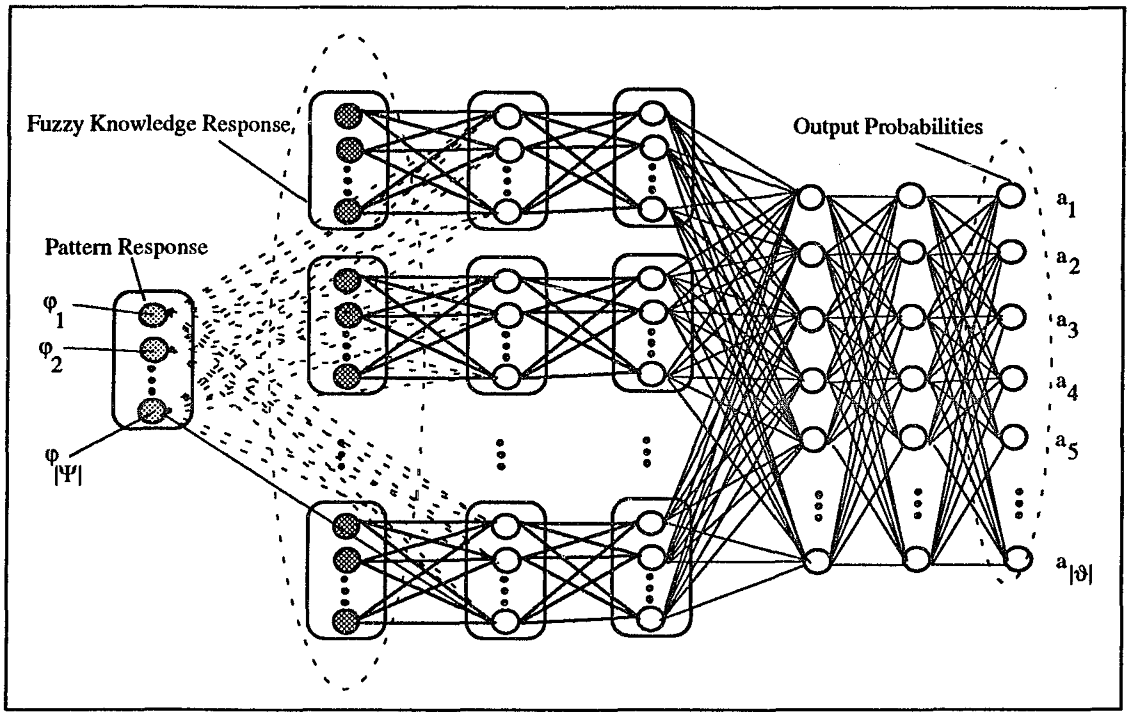


Fig. 4.22 Feed forward neural network to generalize fuzzy response.

4.4. Summary

When processing digital images, different application have different meaning to the semantics of the term "pattern recognition." Recognizing a structure of a box within an image imply a recognition system that is quite general and flexible. Reflected shape of a box is greatly determined by the angle and elevation from which the box's reflection was captured. In addition, the term "box" is quite general to many possible sizes and geometrical shapes. In this dissertation we define the term "pattern recognition" by identifying particular morphological attributes within an image cluster that represents an image of a character that is a member of an alphabet ϑ.

Our implementation of this theory is designed to recognize the English character set. After preprocessing the raw digital image and then partitioning the image into a set of clusters depicting a single alphanumeric element in ϑ, we use fuzzy logic to extract a set, Ψ, of unique morphological attributes from the digital pattern. These attributes, represented as fuzzy numbers, are then the basis of the learning/recognition behavior of the recognition system. Although neural networks are regarded as advantages mechanism for generalizing information, their usage in image processing of this nature will require a sizable system whose response to numerous combination of input stimuli is hard to control. Therefore, we first perform a fuzzy recognition process to eliminate most unfavorable outcomes. These results are then given to a feedforward neural network to further refine the recognition outcome (see Fig. 4.17). The system's output is a set of probabilities for each $a \in \vartheta$. These probabilities represent likelihood that the digital pattern in question represents $a \in \vartheta$. Similar analysis is performed when learning a pattern's stimuli. In the learning mode, the digital pattern is submitted with an element $a \in \vartheta$. After preprocessing and attribute extraction stages the fuzzy knowledge base is updated in regard to a.

Moreover, a back propagation algorithm is used to update the neural interconnection synaptic weights. The storage schema of the fuzzy knowledge data base, and the usage of a neural network reduced the time complexity of the recognition mode on the expanse of the learning mode. That is, it reduces the digital pattern to a set of indexes stored in an array and representing segments of fuzzy number, which in tern depicts various morphological attributes, is done during the learning phase. So was the synaptic interconnection process of the neural net. The recognition phase reduces the digital pattern into a set of indexes that are easily compared with those of the fuzzy knowledge base. An emphasis was given to the recognition mode since the recognition system is mainly used to recognize digital patterns. Extensive learning should be performed as significant new pattern styles are introduced to the system. The time complexity of the system is taken as a average measurement of the system's recognition speed.

Table 4.4 illustrate the time it took the implemented system to recognize various hand written characters of the English alphabet. The recognition time is the measurement it took to isolate and recognize the pattern twenty times on a single CPU in milliseconds. The system has been implemented in C++ on an SIMD machine and used 77 morphological attributes similar to the one illustrated in section 4.3.2. The results of the fuzzy logic is given to a feedforward artificial neural network, as depicted in Fig. 4.22. The digital pattern are recorded with a half page scanner scanning at 200DPI (105 bytes per scanned line).

The times recorded in table 4.4 illustrate that the recognition time greatly depends on the number of pixels (dark and white) which comprise the digital pattern. Still, these times reveal that the system, although inherently parallel in nature, would be commercially effective on simple personal computers as well. The recognition error rate depends on the number of unique pattern exposures that were learned by the system.

Table 4.4 Recognizing digital pattern on a single CPU with and without thinning the pattern.

Digital Pattern	*Symbol Recognized*	*Recog. Time With Thinning*	*Recog Time With No Thinning*
	L	28.87msec	16.5msec
	O	101.6msec	52.2msec
	U	123.2msec	52.2msec
	I	11.1msec	10.3msec
	S	20.3msec	13.7msec
	N	33.6msec	19.2msec
	E	57.7msec	105.3msec
	A	35.7msec	28.5msec

For a single handwriting style, six times exposures to various patterns of a single character in ϑ provide recognition rate that rises to approximately 98%. Still, when a pattern fails to be recognized, it could be learned, augmenting the fuzzy knowledge base. Also, when employing the thinning algorithm in preprocessing the image, the system requires less learning occurrences to achieve high recognition rate. The recognition, however, is significantly slower. Still, thinning time complexity on SIMD machines is notably faster since implementing the thinning algorithms on a SISD machine requires one to serial the algorithms given in Appendix D.

5. Conclusion and Future Research

This dissertation addresses a fundamental computational strategy in processing digital images of hand written English characters using traditional parallel computers. We develop the theory and methodology upon which to process digital images of hand written alphanumeric characters. The traditional approach of processing the huge collection of pixel's information with von Neuman computer architecture is quite slow and cumbersome. Many solutions of this nature have been introduced thus far. Still, an optical character recognition (OCR) system is commercially effective if it can improve or compete with manual translation of various documents by a skilled typist. No OCR system is commercially effective today. This dissertation reflects the general schema upon which the ultimate vision system, a human vision system that processes light that is reflected on the retina (see section 1.2). The eye/brain combination is a true model of efficient methodology of assigning semantics to images whose discrete light reflection is promptly analyzed.

The eye, functioning as the brain's preprocessor element, transforms the light information into specific data. tThese data are not simple mosaic representations of the light shades reflected on the retina through the lens, they represent specific features that compose the image. The eye and the brain's vision system process the light reflection with a complex neural network which embodies billions of neurons. Designing artificial neural networks with this magnitude is not feasible with the available technology. Still, we use the guidelines of the human vision system to develop a robust optical character recognition system. In developing the recognition system we try to transform the image's information domain into a different representation schema in order to reduce the complexity of the recognition problem. Extracting unique morphological attributes, as humans do, enables one to reduce the image recognition process into an attribute

recognition strategy. These attributes were chosen with respect to the guidelines set by Blesser. These guidelines assert that specific knowledge about human classification for characters must first be obtained.

Prior to the feature extraction action, the raw digital image is further refined to reduce noise and to generate skeletonized image that is accommodating in obtaining the pattern's morphological attributes. Two SIMD-MC^2 algorithms have been developed to thin digital images. These algorithms fix deformation problems that exist in previous algorithms. This refinement does not increased the overall execution time complexity. Since the recognition system recognizes a single character at a time, an on-line character isolation technique has been developed to partition the refined image into a set of clusters each representing a single alphanumeric symbol.

The core of the recognition system is a fuzzy knowledge base. Each attribute extracted from an image cluster is recorded as a fuzzy response. Comparing the similarities of digital pattern is then reduce to the fuzzy comparison of morphological attributes. The system itself has no regard to the semantics of each attribute. An attribute's semantic and definition is chosen in correspondence to a specific alphabet at the design phase, enabling automatic processing. Automated computation relieves the system's need to re-compute the meaning of the pattern's attribute response. The fuzzy attribute responses are represented as segment indexes to the domain of the fuzzy numbers. These representation schema enable us to simplify the processing of fuzzy numbers.

The last stage refine the recognition accomplished by the fuzzy logic system. The function of this stage, employing a feedforward neural network, is to further generalize the attribute analysis. This is because each attribute maintains different significance of weight in the overall recognition criteria. This weight may differ when comparing different pattern combination. Table 4.4 illustrates the recognition time of various English alphanumeric pattern on an SISD machine. These results show that the

approach we took in this dissertation in shifting the processing time complexity to the learning phase and architecture schema. This gives us a system that is commercially effective. The essence of this work is to design a system that was best suited to recognize digital pattern using the digital computers employed nowadays. Thus, controlling the complexity of the problem's solution is achieved by changing the problem representation schema.

The recognition system developed in this dissertation should be extended. Future research should continue by augmenting the system to recognize whole words. Consider the two words illustrated in Fig. 5.1.

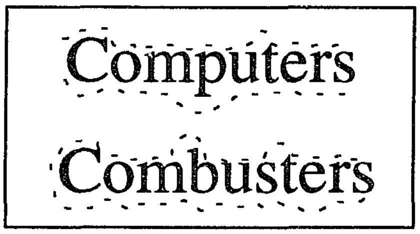


Fig. 5.1 Word feature recognition.

Employing our vision system, we do not analyze the meaning of the patterns in Fig. 5.1 by recognizing each symbol that constitute the two words. Since the word 'Computers' is quite common, we recognize the word as an entity rather than the sequence of characters constituting it. We can have a dictionary knowledge base contrived by past exposure to various words, their meaning, and their spatial pattern. Examining the second word, 'Combusters,' one would perform, in general, some character analysis to achieve its recognition. The textual construction of the word 'Combusters' is similar to that of 'Computers.' Still, we differentiate between them easily, since they maintain significantly different morphological features. The recognition of isolated characters is still needed to differentiate between similarly featured words, or to recognize words that are not in the dictionary knowledge base.

Developing a word recognition system would speed up document recognition significantly and increase the system error rate. This is because it is easier to isolate words then characters. Hand written text usually contain characters that are severely deformed. A character recognition system would not be able to cope with these problems, neither would a human's vision system. It is impossible to recognize severely deformed characters without knowing of the context upon which the characters were recorded under. A word recognition system would have the ability to generalize upon key morphological features. The essence of the word feature analysis should be similar to that used by character recognition. The difference would be the definition of the attributes, and the development of a database structure maintaining the vast number of words to be stored in the knowledge dictionary. Actually, a refiner system would be one that could comprehend the meaning of a sentence. This would enable higher tolerance of ambiguity generated by noise and hand written style.

Bibliography

[1] A. Jackson, Machine Learning, *Expert Systems.,* Oxford., Vol. 5(1988)

[2] Abdulla W. H., Saleh A.O.M, and Morad A. H A preprocessing algorithm for hand-written character recognition., *Pattern Recognition Letters,* Vol. 7, No. 1(1988), 13-18.

[3] Arcelli C., A Condition for digital points removal, *Signal Processing,* Vol. 1(1979), 283-285.

[4] Arcelli C., Pattern Thinning by Contour tracing, *Computer Graphics and Image Processing* Vol. 17, No. 2(1981), 130-144.

[5] Arnold Kaufmann and Masan M. Gupta, *Fuzzy Mathematical Models in Engineering and Management Science, Elsevier Science Publishers B. V.,* (1988).

[6] Arnold Kaufmann, and Madan M. Gupta, Introduction to Fuzzy Arithmetic, *Van Nostrand Reinhold Company Inc.,* (1985).

[7] Azriel Rosenfeld, and Avinash C. Kak, Digital Picture Processing, *Academic Press, Inc.,* Second Edition, Vol 1(1982).

[8] B. K. P. Horn, Robot Vision, *The MIT Press*, Cambridge, Mass.,(1989)

[9] Ballard D. H. and Brown C. M, Computer Vision, *Prentic-Hall*, (1982).

[10] Baron, R. J., The Cerebral Computer, *Hillsdale*,(1987).

[11] Bart Koso, Neural Networks and Fuzzy Systems, *Prentice-Hall, Inc.,* (1992).

[12] Bel-Lan A. and Montoto L., A Thinning transform for digital images, *Signal Processing,* Vol. 3(1981), 37-47.

[13] Beun, M. A flexible method for automatic reading of hand-written numerals. *Philips Technical Review* 33, 4(1973), 89-101.

[14] Blesser B. Shillman, R. Kuklinski T., Cox C. Eden, M. Ventura J., A Theoretical Approach for Character Recognition Based on Phenomenological

Attributes, *International Journal of Man-Machine Studies*, Vol. 6(1974), 701-714.

[15] Caianiello E. R., Outline of a Theory of Thought Processes and Thinking Machines., *Jou. Theor. Biol.* No. 1, (1961)204-235.

[16] Craig A. Lindley, Practical Image Processing In C, *John Wiley & Sons, Inc.*,(1991)

[17] D. Ballard and C. Brown, Computer Vision, *Prentice Hall, Englewood Cliffs*(1982).

[18] D. Dubios and H. Prade., A class of fuzzy measures based on triangular norms — A general framework for the combination of uncertain information., *Int. J. Gen. Sys.* Vol 8., No. 1, (1982)43-61.

[19] D. Dubois and H. Parade, Fuzzy Sets and Systems: Theory and Applications, *Academic Press*, (1980).

[20] D. H. Hubel and T. N. Wiesel, Receptive fields and functional architecture in two non straight visual area (18 and 19) of the cat, *Jour. Neurophysiol.*, Vol 28, (1965)229-289.

[21] D. M. Green and J. A. Swets, Signal Detection Theory and Psycophysics., *Wiley*, (1966).

[22] David Andrews, Delrina's New Fax Software Adds OCR, *BYTE*, Vol 18, No. 1(1992) 62-64.

[23] David E. Rumelhar, T James , L. McClelland, and PDP Research Group., Parallel Distributed Processing Exploration in Micro structure of Cognition., Vol. 1., *MIT.*, (1988).

[24] Denning P. J. Blindness in Designing Intelligent Systems., *American Scientist.*, No. 76, (1988)118-120.

[25] Dill, A.R., and Levine, M.D. Multiple resolution skeletons. *IEEE Trans. Pattern Analysis and Machine Intelligence* PAMI-9,4(1987), 495-504

[26] E. H. Land.,, Color Vision and the Natural Image., *Proceedings of the National Academy of Science.*, Vol. 45, (1959)115-129.

[27] Eduard Hovy, MT at Your Service, *BYTE*, Vol 18, No. 1(1992) 160-161.

[28] Elliott B. Mason, Human Physiology, *The Benjamin/Cummings Publishing Company*(1983),185-202.

[29] F. Kimura and M. Shridhar., Handwritten Numerical Recognition Based on Multiple Algorithms., *Pattern Recognition.,* Vol. 24., No. 10., (1991)969-983.

[30] Franco P. Preparata, and Michael Ian Shamos, Computational Geometry, An Introduction, *Springer-Verlag*,(1985).

[31] Fu K. S. and Mui J. K, A Survey of Image Segmentation, *Pattern Recognition.,* Vol. 13, No. 1, (1981)3-16.

[32] Fu K. S., Gonzalez R. C., and Lee C.S.G., Robotics: Control, Sensing, Vision and Intelligence., *McGraw-Hill,* New York(1987).

[33] George N. Bebis and George M. Papadourakis, Object Recognition Using Invariant Object Boundary Representations and Neural Network Models, *Pattern Recognition,* Vol. 25, No. 1(1992), 25-44.

[34] Gonzalez Rafael C. and Wintz Paul, Digital Image Processing, *Addition-Wesley*(1987),398-402.

[35] Grosky W. I. and Jain R., Optimal Quadtrees for Image Segments., *IEEE Trans. Pattern Anal. Machine Intell.,* Vol. PAMI-5, No. 1,(1983)77-83.

[36] Gulati, S. and S. T. Venkataraman, Perceptual Robotics: A Vehicle for Synergistic Man-machine Systems, *IEEE Conf. on Bio. Eng.,* (1990).

[37] Hebb, D. O., Organization of Behavior, *Wiley.,* (1949).

[38] Hinton G. E., Distributed Representations., *Tech. Report,* CMUCS-84-157, Dep. of Computer Science, CMU, Pittsburgh.

[39] Holt Christopher M, Stewart Alan, Clint Maurece, and Perrot Ronald H., An Improved Parallel Thinning Algorithm, *Comm. of the ACM,* Vol. 30 No. 2(1987)156-157.

[40] Horowitz S. L. and Pavlidis T., Picture Segmentation by a Directed Split-and-Merge Procedure., *Proc. 2nd Int. Joint Conf. Pattern Recog.,* (1974)423-433.

[41] Itshak Dinstein and Gad M. Landau., Parallel computable contour based feature strings for 2-D shape recognition., *Pattern Recognition Letters.,* Vol. 12., (1991)299-306.

[42] J. J. Hull, S. N. Srihari, E. Cohen, C. L. Kuan, P. Cullen and P. Palumbo., A blackboard-based Approach to Handwritten ZIP Code Recognition., *Proc. USPS advd Technol. Conf.*,(May 1988)855-866.

[43] J. Sklansky, On the Hough Technique for Curve Detection, *IEEE Transactions on Computers.*, Vol. C-27, No. 10(1978)923-926.

[44] James C. Bezdek, Patter Recognition with Fuzzy Objective Function Algorithms, *Plenum Press,* (1981).

[45] K. Fukushima, and Sei Miyake, Neocognitron: A New Algorithm for pattern recognition Tolerant of Deformations And Shifts in Position., *Pattern Recognition.,* Vol. 15., No. 6.,(1982)455-469.

[46] K. Fukushima, S. Miyake, and T. Ito., Neocognitron: A Neural Network Model for a Mechanism of Visual Pattern Recognition. *IEEE Transactions on Systems, Man and Cybernetics.* SMC-13 Vol. 5(1983) 826-834.

[47] K. Hanahara, T. Maruyama, and T. Uchiyama, High Speed Hough Transform and its Application to Automatic Inspection and Measurement, *IEEE Int. Conference on Robotics and Automation,* (1986)1954-1959.

[48] Kanizsa, G., Subjective Contours, *Scientific American,* Vol. 234(1976)48-52.

[49] Kent M. and Van De Graff, Human Anatomy, *Wm. C. Brown Publishers*(1988),436-487.

[50] Klinger A., Experiments in Picture Representation Using Regualr Decomposition., *Comput. Graphics Image Proc.*, Vol. 5, (1976)68-105.

[51] Kwok Paul C., A Thinning Algorithm by Contour Generation, *Comm. of the ASM,* Vol. 31, No. 11(1988), 1314-1324.

[52] L. A. Zadeh, Fuzzy Sets, *Information Control,* Vol. 8(1965)338-353.

[53] Levine, M. D.,, Feature Extraction: A Survey, *Proc. IEEE,* Vol. 57, (1969)1391-1407.

[54] Lu H. E. and Wang P. S. P., A Comment on "A Fast Parallel Algorithm for Thinning Digital Patterns", *Comm. of the ACM,* Vol. 29, No. 3(1986), 239-242.

[55] M. Celenk and C. K. Lim, Parallel Implementation of Low-level Vision Operators on a Hypercube machine, *Optical Engineering,* Vol. 30, (1991)275-283.

[56] Mark D. M. and Abel D. J., Linear Quadtrees from Vector Representations of Polygons, *IEEE Trans. Pattern Anal. machine Intell.,,* Vol. PAMI-7, No. 3, (1985)344-349.

[57] McCulloch W. S. and Pitts W. H, A logical Calculus of the Ideas Immanent in Nervous Activity, *Bulletin of Mathematical Biophysics.,* Vol. 3, No. 115, (1943).

[58] Moayer, B., and Fu, K.S. A tree system approach for fingerprint pattern recognition. *IEEE Trans. Comput.* C-25, 3(1976), 262-275

[59] Muerle J.L. and Allen D. C., Experimental Evaluation of Techniques for Automatic Segmentation of Object in a Complex Scene." In *Pictorial Pattern Recognition, Thompson Book Co.,* (1968).

[60] Muriel Vasconcellos, Machine Translation, *BYTE,* Vol 18, No. 1(1992) 153-166.

[61] Naccache Nabil J. and Shinghal Rajjan, SPTA: A Proposed Algorithm for Thinning Binary Patterns, *IEEE Trans. on Systems Man and Cyber.* Vol. SMC-14, No 3(1984), 409-418.

[62] P. J. Dais., Interpolation and approximation., *Dover,* (1975).

[63] Paolo Antognetti and Veljko Milutinovic, Neural Networks Concepts, Applications, and Implementations., *Prentice Hall,* Vol. 1,(1991).

[64] Paul R. Halmos, Naive Set Theory, *D. Van Nostrand Company, Inc.* (1960).

[65] Pavlidis T., Algorithms for graphics and image processing. *Computer Science Press,* Rockville, Md(1982).

[66] R. A. Jarvis, On the identification of the convex hull of a finite set of points in the plane, *Info. Proc. Lett.* Vol. 2, (1973)18-21.

[67] R. T. Chin and H. K. Wan and D. L. Stover and R. D. Iverson, A One-Pass Thinning Algorithm and Its Parallel Implementation, *Computer Vision, Graphics, and Imge Processing,* Vol. 40, (1987)30-40.

[68] S. K. Pal, and N. R. Pal, Segmentation using contrast and homogeneity measures, *Pattern Recognition Letters.,* Vol. 5.,(1987)293-304.

[69] S. N. Srihari, E. Cohen, J. J. Hull and L. Kuan., A system to locate and recognize ZIP codes in hand-written addresses., *IJRE.,* Vol. 1., (1989)34-45.

[70] Schantz Herbert F, The History of OCR Optical Character Recognition, *Recognition Technologies Users Asso.*(1982).

[71] Shozo Kondo, On Determining Distinctive Features of Hand printed Characters Using Statistical Techniques, *Computer Processing of Handwriting*, Eds. R. Plamondon and C. G. Leedham, World Scientific(1990), 208-219.

[72] Sirjani Abolfazl and Cross George R., On representation of a shape's skeleton, *Pattern Recognition Letters* 12(1991) 149-154

[73] Sklansky J., Chazin R. L. and Hansen B. J, Minimum-Perimeter Polygons of Digitized Silhouettes, *IEEE Trans. Comput.*, Vol. C-21, No. 3(1972] 260-268.

[74] V. K. Govindan, and A. P. Shivaprasad., Artificial database for character recognition research., *Pattern Recognition Letters.*, Vol. 12., (1991)645-648.

[75] Verhagen C. J. D. M., Some General Remarks about Pattern Recognition; Its Definition; Its Relation with other Disciplines; A Literature Survey, *Journal of Pattern Recognition.*, Vol. 8, No. 3,(1975)109-116.

[76] Vogt Robert C., Automatic Generation of Morphological Set Recognition Algorithms, *Springer-Verlag New York Inc.*,(1989).

[77] Wang P.S.P and Zhang Y.Y, A Fast and Flexible Thinning Algorithm, *IEEE Trans. on Comp.*, Vol.38, No. 5(1989)741-745.

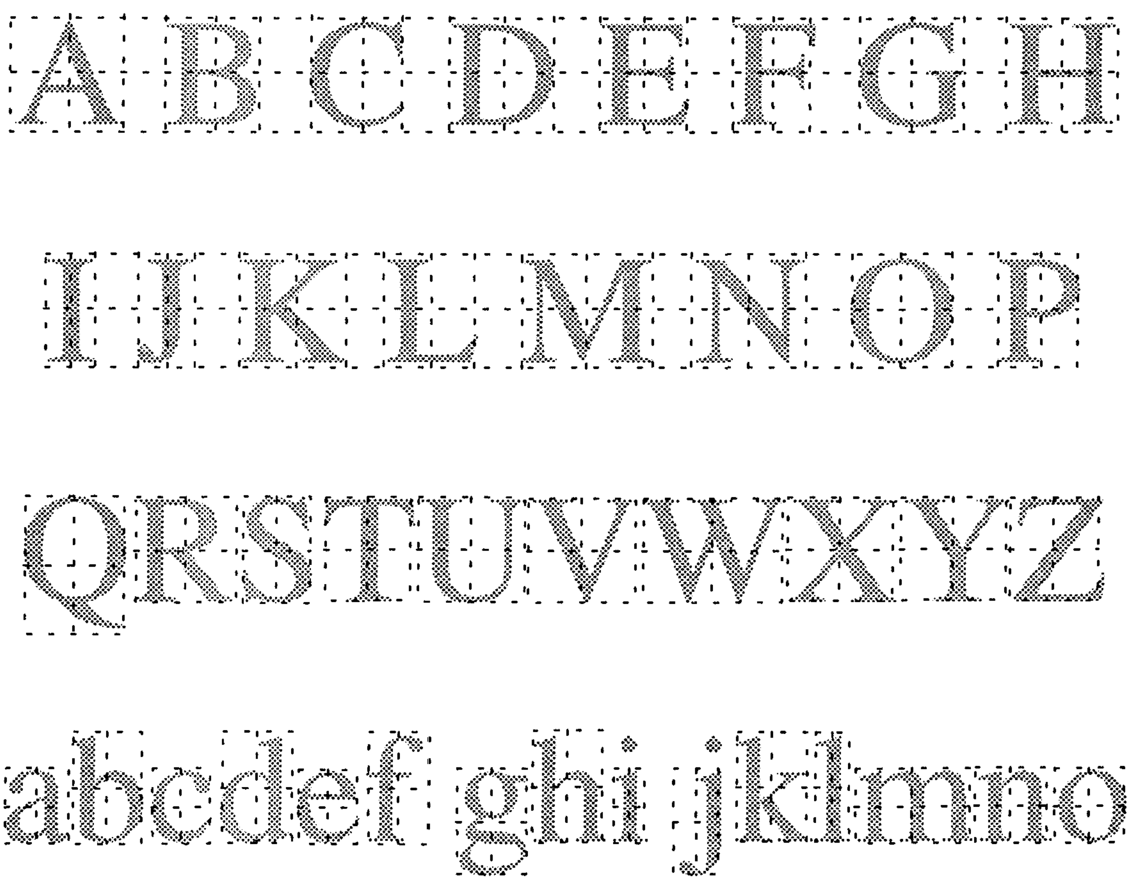
[78] Zhang T. Y. and Suen C. Y., A Fast Parallel Algorithm for Thinning Digital Patterns, *Comm. of the ACM,* Vol. 27, No. 3(1984), 236-239.

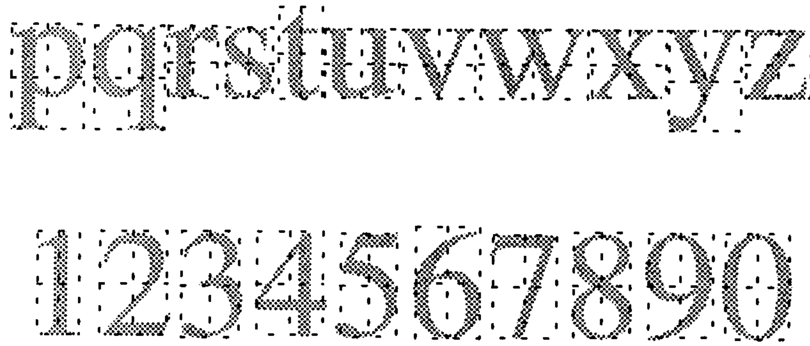
[79] Zucker S. W., Region Growing: Childhood and Adolescence, *Comput. Graphics Image Proc.* Vol. 5,(1976)382-399.

[80] Zuguang Cai., A Handwritten Numeral Recognition System Using a Multi-Microprocessor., *Pattern Recognition Letters.,* Vol. 12., (1991)503-509.

Appendix A: The English Alphabet

Close examination of the English alphanumeric characters reveals that their spatial pattern, in general, could be partitioned into four quadrants each of which maintain a simpler spatial design. Different print fonts and, of course, human hand written text may not conform to the quadrant partition as uniformly as illustrated in the following figures. Still, the main contour characteristics are usually maintained. Employing morphological attributes of the figure's quadrant partitions has been shown to be invaluable when used as a discriminator criteria among the various English alphanumeric characters.




pqrstuvwxyz
1234567890

Appendix B: Hough Transformation

The Hough transform is a machine vision algorithm that detects and locates straight lines in digital images. It will succeed in the presence of noise, and can also find lines that are not continuous. It can also be generalized to detect curves[43]. Part of the Hough transform is inherently parallel. The contribution to the transform from any particular point in the image can be calculated independently of the value of any other point in the image. It is only when the contribution of all points are summed that the algorithm becomes sequential. For this reason, no inter-nodal communication is needed during the computation of the transform.

$$y_i = a x_i + b \qquad (B.1)$$

or

$$b = -x_i a + y_i$$

The essence of the technique is to consider the possibility that infinite lines could pass through each spatial coordinate within an image. Since a line could be represented as in equation B.1, at each coordinate pair (x_i, y_i) we can consider the *ab* plane (also called parameter space). Then, each point in the *xy* plane will add a counter on the *ab* plane for each possible slope *a*. Thus, a line in which many pixels pass through will result in a large counter on the respective coordinate on the *ab* plane. The problem of equation B.1 is in representing a line that both the slope and intercept approach infinity as the line approaches a vertical position. Thus, a plane transformation is obtain by using equation B.2 (see Fig. B.1).

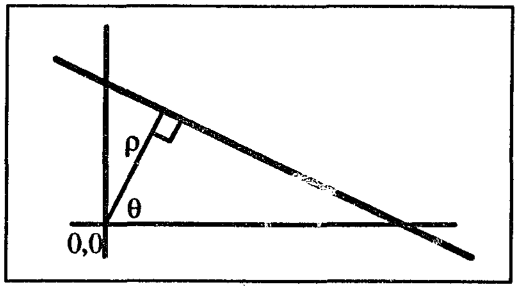


Fig. B.1 The coordinate system is based on the angle θ and the normal ρ.

The algorithm maps the image to a transform space characterized by ρ and θ coordinates. The (ρ,θ) transform space is quantized, depending on how much accuracy is needed for the final answer, and each coordinate is originally set to zero. Then each 1-valued coordinate (x,y) in the image space is put into the parametric equation B.2.

$$\rho = x\cos(\theta) + y\sin(\theta) \qquad (B.2)$$

Rho is varied from zero to 180 degrees, and the value of the transform is incremented at every (ρ,θ) pair that matches. The result is that every point in the image is transformed into a sinusoidal curve in the transform space. The final value of each of (ρ,θ) coordinate in the transform space will be the number of curves that intersect that point. Points that form a line in the image space will intersect at a value of (ρ,θ), where ρ is the length of the normal of that line from the origin, and θ is the angle of the normal from the axis as seen in Fig. B.1. In equation B.2, ρ is a signed quantity. This is because θ actually ranges from zero to 360 degrees. If ρ is calculated to be negative, this signifies that the line is in the lower quadrants, between 180 and 360 degrees.

Appendix C: MasPar SIMD-MC^2

The MasPar MP-1 System is a massively parallel Single Instruction, Multiple Data (SIMD) computer system with 8,000 microprocessors (PEs). The system uses a register-based load/store architecture with separate instruction and data memory areas. The front end is a UNIX subsystem that provides network and graphical services, including the software environment in which all the MasPar tools and utilities (such as compilers) are executed. It contains the interface to the Data-Parallel Unit (DPU), which through the PE array performs the computational intensive, parallel portions of applications. The Array Control Unit (ACU) executes PE array instruction and executes instructions on the ACU proper (such as array housekeeping and front end communication). The PE array supports the following communications mechanisms:

- Neighbor communication - one bit per clock cycle where any number of PEs can take part in this communication phase.
- Router communication - communication of one bit per cycle from any PE to any other PE. These operations are distance-insensitive.
- Broadcasting - This allows data to be broadcast from the ACU to all PEs, where each PE can independently elect to receive the data.
- Or-Reduction Communication - a boolean logical OR reduction network that enable global PE data conditions to be easily detected.

Programming the MasPar is done by the MPL language. MPL is an extension of K&R C which includes features that are specific to the MasPar environment.

Appendix D: Thinning Algorithm Listing

The following is a *C* like code for a cyclic mesh connected SIMD machine where $PE_{i,j}$ is the reference to the processor element in the i^th row and the j^th column.

Algorithm I (SIMD-MC^2),

INPUT : *binary pattern*

OUTPUT : *pattern's skeleton*

[1] Stop = True

[2] for all PE_{ij} do

[3] $North_{i,j} \Leftarrow Pixel_{i-1,j}$

[4] $NEast_{i,j} \Leftarrow Pixel_{i-1,j+1}$

[5] $East_{ij} \Leftarrow Pixel_{i,j+1}$

[6] $SEast_{i,j} \Leftarrow Pixel_{i+1,j+1}$

[7] $South_{i,j} \Leftarrow Pixel_{i+1,j}$

[8] $SWest_{i,j} \Leftarrow Pixel_{i+1,j-1}$

[9] $West_{i,j} \Leftarrow Pixel_{ij-1}$

[10] $NWest_{i,j} \Leftarrow Pixel_{i-1,j-1}$

[11] $EEast_{i,j} \Leftarrow East_{i,j+1}$

[12] $WWest_{i,j} \Leftarrow West_{i,j-1}$

[13] $B_{i,j} = North + NEast_{i,j} + East_{i,j} + SEast_{i,j} +$
$South_{i,j} + SWest_{i,j} + West_{i,j} + NWest_{i,j}$

[14] $A_{i,j} = (!North_{i,j}\ \&\&\ NEast_{i,j}) +$
$(!NEast_{i,j}\ \&\&\ East_{i,j}) + (!East_{i,j}\ \&\&\ SEast_{i,j}) +$
$(!SEast_{i,j}\ \&\&South_{i,j}) + (!South_{i,j}\ \&\&SWest_{i,j}) +$
$(!SWest_{i,j}\ \&\&West_{i,j}) + (!West_{i,j}\ \&\&NWest_{i,j}) +$
$(!NWest_{i,j}\ \&\&\ North_{i,j})$

[15] $C_{i,j} = !East_{i,j} \parallel !South_{i,j} \parallel (!North_{i,j}\ \&\&!West_{i,j})$

[16] $D_{i,j} = (B>2) \parallel ((B==2)\ \&\&\ (EEast_{i,j} \parallel WWest_{i,j}))$

[17] if $((B_{i,j}>=2)\&\&(B_{i,j}<=6)\&\&(A_{i,j}==1)\&\&C_{i,j}\&\&D_{i,j})$

[18] $Pixel_{i,j}$ = False

[19] Stop = False

[20] fi

[21] rof

[22] if (Stop) *goto* [45]

[23] Stop = True

[24] for all PE_{ij} do

[25] $North_{i,j} \Leftarrow Pixel_{i-1,j}$

[26] $NEast_{i,j} \Leftarrow Pixel_{i-1,j+1}$

[17] $East_{ij} \Leftarrow Pixel_{i,j+1}$

[28] $SEast_{i,j} \Leftarrow Pixel_{i+1,j+1}$

[29] $South_{i,j} \Leftarrow Pixel_{i+1,j}$

[30] $SWest_{i,j} \Leftarrow Pixel_{i+1,j-1}$

[31] $West_{i,j} \Leftarrow Pixel_{ij-1}$

[32] $NWest_{i,j} \Leftarrow Pixel_{i-1,j-1}$

[33] $EEast_{i,j} \Leftarrow East_{i,j+1}$

[34] $WWest_{i,j} \Leftarrow West_{i,j-1}$

[35] $B_{i,j} = North_{i,j} + NEast_{i,j} + East_{i,j} + SEast_{i,j} +$

$South_{i,j} + SWest_{i,j} + West_{i,j} + NWest_{i,j}$

[36] $A_{i,j} = (!North_{i,j} \&\& NEast_{i,j}) + (!NEast_{i,j} \&\& East_{i,j}) +$

$(!East_{i,j} \&\& SEast_{i,j}) + (!SEast_{i,j} \&\& South_{i,j}) +$

$(!South_{i,j} \&\& SWest_{i,j}) + (!SWest_{i,j} \&\& West)_{i,j} +$

$(!West_{i,j} \&\& NWest_{i,j}) + (!NWest_{i,j} \&\& North_{i,j})$

[37] $C_{i,j} = !West_{i,j} \,||\, !North_{i,j} \,||\, (!East_{i,j} \&\& !South_{i,j})$

[38] if $((B_{i,j}>=2) \&\& (B_{i,j}<=6) \&\& (A_{i,j}==1) \&\& C_{i,j})$

[39] $Pixel_{i,j}$ = False

[40] Stop = False

[41] fi

[42] rof

[43] if (Stop) *goto* [45]

[44] *goto* [1]

[45] *End.*

Algorithm II (SIMD-MC^2),

INPUT : *binary pattern*

OUTPUT : *pattern's skeleton*

[1] Stop = True

[2] for all PE_{ij} do

[3] $North_{i,j} \Leftarrow Pixel_{i-1,j}$

[4] $NEast_{i,j} \Leftarrow Pixel_{i-1,j+1}$

[5] $East_{ij} \Leftarrow Pixel_{i,j+1}$

[6] $SEast_{i,j} \Leftarrow Pixel_{i+1,j+1}$

[7] $South_{i,j} \Leftarrow Pixel_{i+1,j}$

[8] $SWest_{i,j} \Leftarrow Pixel_{i+1,j-1}$

[9] $West_{i,j} \Leftarrow Pixel_{ij-1}$

[10] $NWest_{i,j} \Leftarrow Pixel_{i-1,j-1}$

[11] $EEast_{i,j} \Leftarrow East_{i,j+1}$

[12] $WWest_{i,j} \Leftarrow West_{i,j-1}$

[13] $NNorth_{i,j} \Leftarrow North_{i-1,j}$

[14] $t00_{i,j} = (!North_{i,j} \&\& (!NWest_{i,j} \,||\, !NEast_{i,j})) \,||$

$(!East_{i,j} \&\& (!NEast_{i,j} \,||\, !SEast_{i,j})) \,||$

$(!South_{i,j} \&\& (!SEast_{i,j} \,||\, !SWest_{i,j})) \,||$

$(!West_{i,j} \&\& (!SWest_{i,j} \,||\, !NWest_{i,j}))$

[15] $t111_{i,j} = (NWest_{i,j} \&\& North_{i,j} \&\& NEast_{i,j}) \,||$

$(NEast_{i,j} \&\& East_{i,j} \&\& SEast_{i,j}) \,||$

$(SEast_{i,j} \&\& South_{i,j} \&\& SWest_{i,j}) \,||$

$(SWest_{i,j} \&\& West_{i,j} \&\& NWest_{i,j})$

[16] $t11_{i,j} = (North_{i,j} \&\& (NWest_{i,j} \,||\, NEast_{i,j})) \,||$

$(East_{i,j} \&\& (NEast_{i,j} \,||\, SEast_{i,j})) \,||$

$(South_{i,j} \&\& (SEast_{i,j} \,||\, SWest_{i,j})) \,||$

$(West_{i,j} \&\& (SWest_{i,j} \,||\, NWest_{i,j}))$

[17] $t01_{i,j} = (!NWest_{i,j} \&\& North_{i,j}) +$

$(!North_{i,j} \&\& NEast_{i,j}) + (!NEast_{i,j} \&\& East_{i,j}) +$

$(!East_{i,j} \&\& SEast_{i,j}) + (!SEast_{i,j} \&\& South_{i,j}) +$

$(!South_{i,j} \&\& SWest_{i,j}) + (!SWest_{i,j} \&\& West_{i,j}) +$

$(!West_{i,j} \&\& NWest_{i,j})$

[18] $EdgeF_{i,j} = Pixel_{i,j} \&\& t00_{i,j} \&\& (t01_{i,j}<2) \&\& t11_{i,j}$

[19] $\mathrm{EdgeSouth}_{i,j} \Leftarrow \mathrm{EdgeF}_{i+1,j}$

[20] $\mathrm{EdgeEast}_{i,j} \Leftarrow \mathrm{EdgeF}_{i,j+1}$

[21] $\mathrm{Keep}_{i,j} = !\mathrm{EdgeF}_{i,j}\ ||$

$(\mathrm{EdgeSouth}_{i,j}\ \&\&\ \mathrm{West}_{i,j}\ \&\&\ \mathrm{East}_{i,j})\ ||$

$(\mathrm{EdgeEast}_{i,j}\ \&\&\ \mathrm{North}_{i,j}\ \&\&\ \mathrm{South}_{i,j})\ ||$

$(\mathrm{EdgeF}_{i,j}\ \&\&\ !\mathrm{t111}_{i,j}\ \&\&\ !\mathrm{West}_{i,j}\ \&\&$

$!\mathrm{NNorth}_{i,j}\ \&\&\ !\mathrm{EEast}_{i,j}\ \&\&\ !\mathrm{WWest}_{i,j})$

[22] if $(\mathrm{Pixel}_{i,j}\ \&\&\ !\mathrm{Keep}_{i,j})$

[23] $\mathrm{Pixel}_{i,j} = \mathrm{False}$

[24] Stop = False

[25] fi

[26] if (!Stop) *goto* [1]

[27] rof

[28] *End*

Stair Case Elimination (SIMD-MC^2),

(Seven Neighbor Test)

INPUT : *skeleton with stair case extra pixels*

OUTPUT : *unitary skeleton*

[1] for all PE_{ij} do

[2] $\mathrm{North}_{i,j} \Leftarrow \mathrm{Pixel}_{i-1,j}$

[3] $\mathrm{NEast}_{i,j} \Leftarrow \mathrm{Pixel}_{i-1,j+1}$

[4] $\mathrm{East}_{ij} \Leftarrow \mathrm{Pixel}_{i,j+1}$

[5] $\mathrm{SEast}_{i,j} \Leftarrow \mathrm{Pixel}_{i+1,j+1}$

[6] $\mathrm{South}_{i,j} \Leftarrow \mathrm{Pixel}_{i+1,j}$

[7] $\mathrm{SWest}_{i,j} \Leftarrow \mathrm{Pixel}_{i+1,j-1}$

[8] $\mathrm{West}_{i,j} \Leftarrow \mathrm{Pixel}_{ij-1}$

[9] $\mathrm{NWest}_{i,j} \Leftarrow \mathrm{Pixel}_{i-1,j-1}$

[10] $\mathrm{Ok}_{i,j} = \mathrm{Pixel}\ \&\&!(\mathrm{North}_{i,j}\&\&((\mathrm{East}_{i,j}\&\&!\mathrm{NEast}_{i,j}\&\&$

$!\mathrm{SWest}_{i,j}\ \&\&\ (!\mathrm{West}_{i,j}\ ||\ !\mathrm{South}_{i,j}))\ ||$

$(\mathrm{West}_{i,j}\&\&!\mathrm{NWest}_{i,j}\ \&\&\ !\mathrm{SEast}_{i,j}\ \&\&$

$(!\mathrm{East}_{i,j}\ ||\ !\mathrm{South}_{i,j}))))$

[11] if $(!\mathrm{Ok}_{i,j})$ $\mathrm{Pixel}_{i,j} = \mathrm{False}$

[12] rof

[13] for all PE_{ij} do

[14] $\mathrm{North}_{i,j} \Leftarrow \mathrm{Pixel}_{i-1,j}$

[15] $\mathrm{NEast}_{i,j} \Leftarrow \mathrm{Pixel}_{i-1,j+1}$

[16] $\mathrm{East}_{ij} \Leftarrow \mathrm{Pixel}_{i,j+1}$

[17] $\mathrm{SEast}_{i,j} \Leftarrow \mathrm{Pixel}_{i+1,j+1}$

[18] $\mathrm{South}_{i,j} \Leftarrow \mathrm{Pixel}_{i+1,j}$

[19] $\mathrm{SWest}_{i,j} \Leftarrow \mathrm{Pixel}_{i+1,j-1}$

[20] $\mathrm{West}_{i,j} \Leftarrow \mathrm{Pixel}_{ij-1}$

[21] $\mathrm{NWest}_{i,j} \Leftarrow \mathrm{Pixel}_{i-1,j-1}$

[22] $\mathrm{Ok}_{i,j}=\mathrm{Pixel}_{i,j}\&\&!(\mathrm{South}_{i,j}\&\&((\mathrm{East}_{i,j}\&\&!\mathrm{SEast}_{i,j}$

$\&\&!\mathrm{NWest}_{i,j}\&\&(!\mathrm{West}_{i,j}||!\mathrm{North}_{i,j}))\ ||$

$(\mathrm{West}_{i,j}\&\&!\mathrm{SWest}_{i,j}\ \&\&\ !\mathrm{NEast}_{i,j}\ \&\&$

$(!\mathrm{East}_{i,j}\ ||\ !\mathrm{North}_{i,j}))))$

[23] if $(!\mathrm{Ok}_{i,j})$ $\mathrm{Pixel}_{i,j} = \mathrm{False}$

[24] rof

[25] *End*

Vita

Gili Mendel is a member of the technical staff with the Robotics Research Group of Louisiana State University. Mendel received his B.S. degree in Computer Science from Nicholls State University in August 1989. He is currently completing his doctoral degree in Computer Science form Louisiana State University. His present research interests include neural network theory and applications, fuzzy logic and image processing. During his undergraduate studies Mendel received numerous awards for scholastics excellence such as the Who's Who Among Students in American Universities and Colleges award and the National Dean's List award. He is also a member of Phi Kappa Phi, Epsilon Pi Epsilon, Phi Eta Sigma, and Alpha Lambda Delta honor societies. During his doctoral studies Mendel performed research with the Center for Space Microelectronics Technology within the Jet Propulsion Laboratory, California Institute of Technology, and with the AIX/ESA source development team at IBM, Kinston New York.

DOCTORAL EXAMINATION AND DISSERTATION REPORT

Candidate: Gili Mendel

Major Field: Computer Science

Title of Dissertation: Optical Character Recognition Using Morphological Attributes.

Approved:

Major Professor and Chairman

Dean of the Graduate School

EXAMINING COMMITTEE:

Date of Examination:

3/10/93