2007

Hidden Markov model and its application in document image analysis

Songtao Huang
University of Windsor

Hidden Markov Model and Its Application In Document Image Analysis

by

Songtao Huang

A Dissertation
Submitted to the Faculty of Graduate Studies
through Electrical and Computer Engineering
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy
at the University of Windsor

Windsor, Ontario, Canada
2007

Your file Votre référence
ISBN: 978-0-494-35182-6
Our file Notre référence
ISBN: 978-0-494-35182-6

Abstract

The document image analysis has been intensively studied in the last decades, while Hidden Markov Model (HMM) turns out to be the mainstream method in image processing and computer vision. In this paper we strive to implement HMM in the most critical two steps of document image analysis: Binarization and Optical Character Recognition(OCR). We propose a HMM based binarization method, whose efficiency is demonstrated through the simulation results of document images with different degradations. At the same time we also introduce an edge based binarization method, which has lower computational burden and higher performance than other similar methods. Test results for OCR applications show 77% correct rate for HMM based binarization method and 67.3% for the edge based binarization method, while the best performance for other reference techniques is 57%.

The conventional HMM based classifier is a causal system, which means the deduction of the hidden states is unidirectionally obtained along a single path. Thereby, the performance of the HMM classifier can be degraded by the noise mixed in the signals, which is inevitable for the real world applications. Here we propose a new non-causal Self-Adaptive Hidden Markov Model(SAHMM) for Optical Character Recognition(OCR) application. The experiment results prove that it has stronger immunity to the noise than the conventional methods. We also extend the 1-D SAHMM into 2-D, where some new feature extraction methods and new architecture for the nodes in the model are introduced. The proposed 2-D SAHMM OCR engine achieves recognition rates of up to 96% on the MNIST database, which is higher than any reported single HMM based OCR engine.

Acknowledgements

First of all, I would like to express my appreciation to my supervisor, Dr. Majid Ahmadi. His overall enthusiasm and devotion to his work have deeply impressed me. He is highly critical, while remaining very supportive. He not only provides invaluable guidance to my study, but also sets up the best role model for me and my colleagues to follow.

Secondly, I want to thank Dr. Sid-Ahmed. Though he is not my supervisor, he always selflessly provides me some professional suggestions about my study and my career plan. I also would like to thank my committee memebrs, Dr. Boubakeur Boufama and Dr. Chunhong Chen, who monitor my work and take effort in reading and providing me with valuable comments on my theis.

I was lucky to work with so many energic and intelligent classmates and colleagues. I enjoyed every minutes I spend with them and cherish all of the help from them.

Especially, I would like to give my thanks to my parent, Shengchu Huang, Fei Xiao and my wife, Huifang Chen. Without their continuous encourage and support, it is impossible for me to finish all of the jobs in the thesis.

Contents

List of Figures

List of Tables

List of Abbreviations

1-D	One Dimensional.
2-D	Two Dimensional.
CC	Computation Cost.
CN	Convolutional Network.
DDE	Decision Directed Estimation.
EHMM	Embedded Hidden Markov Model.
HMM	Hidden Markov Model.
HNN	Hopfield Neural Network.
kNN	k-Nearest-Neighbor.
LUT	Look Up Table.
MAP	Maximum A Posteriori Probability.
MAO	Multiply-Accumulate Operation.
ML	Maximum Likelihood.
MMRF	Markov Mesh Random Field.
MNIST	Modified National Institute of Standards and Technology.
NN	Neural Network.
NSHP	Nonsymmetric Half-Plane.
OCR	Optical Character Recognition.
PDA	Personal Digital Assistant.
PHMM	Pseudo Hidden Markov Model.
RBF	Radial Basis Function.
RR	Recognition Rate.
SAHMM	Self-Adaptive Hidden Markov Model.
SDNN	Space Displacement Neural Network.
SOM	Self-Orginizing Map.
SVM	Support Vector Machine.
TDNN	Time Delay Neural Network.

List of Symbols

The notation used in this thesis is as follows. In general, Bold face upper case letters designate matrices, bold face lower case letters designate vectors, and scalers are designated by italics. A scaler element of a vector is denoted $v_i \in \mathbf{v}$, which is read as "the i^{th} element of vector $\mathbf{v}$," with indexing of vectors starting at $i = 1$ and ranging to the length of the vector. All vectors are assumed to be column vectors. Time is denoted in parenthesis for matrices, vectors, and scalers; for example, $v_i(n)$ is interpreted as the "i^{th}" element of vector $\mathbf{v}(n)$ at time instant "n". However, the time indication in parenthesis is typically dropped for scaler quantities since the vector index of most scalers is equivalent to the time index. Some commonly used symbols are listed below.

Notation	Definition
λ	Compact notation of all of the parameters in a model.
$\mu_\chi^{(g)}$	The mean vector of the g^{th} Gaussian kernel in the χ^{th} genone N_χ.
π_i	Initial state probability in the conventional 1-D HMM.
$\mu_j^{(g)}$	The mean of the g^{th} Gaussian component in the observation PDF of the state S_j.
$\alpha_j^{(g)}$	The variance of the g^{th} Gaussian component in the observation PDF of the state S_j.
$\alpha_t(i)$	The forward variable; $\alpha_t(i)$ is the joint probability that the partial observation sequence $O_1, O_2, \cdots, O_t$ occurs and that the state S_i is the current active state q_t in the model λ.
$\beta_t(i)$	The joint probability that the partial observation sequence $O_t, O_{t+1}, \cdots, O_T$ occurs and that the state S_i is the current active state q_t in the model λ.
$\gamma_t(i)$	The probability of the state of index i to be the active state at the time t under the conditions of given observation sequence and model parameters.

$\delta_t(i)$	The observation likelihood maximized over the past state sequence terminating with S_i at time t.
$\psi_t(j)$	The index of the best predecessor state of the current state S_j at time t.
$\xi_t(i,j)$	The probability of S_i to be active states at time t, and state S_j occurring at time $t+1$.
$\mathbf{A}$	State transition matrix of the conventional 1-D HMM.
a_{ij}	The probabitity of state transition from state Si to state S_j.
$\mathbf{B}$	The observation probability matrix of the conventional 1-D HMM.
$b_j(O_t)$	The probability that the observation at time t has been generated by state S_j.
$b_{ij}^d(k)$	The probability of distribution of observations to every state in 2D SAHMM, $[i,j]$ is the coordinate of the states, k is the symbol of the obervation, the superscript d is the direction of the four 1-D vector around the point.
$C_{i,j}$	The probability of the state $[i,j]$ occurring in a 2D SAHMM.
C_j	The probability of the state j occurring in the model,
$c_j^(g)$	The weight of the g_{th} Gaussian component of the observation PDF of the state S_j.
$D_t(j)$	The distribution probability of states in every time slot of the observation sequence.
$D_{xy}(ij)$	The distribution probability of states in every location of the 2D observation sequence in a 2D SAHMM, x,y is the coordinate of the critical point, while i,j are the coordinate of the hidden state.
E_{fore}	The foreground pixels error rate.
E_{back}	The background pixels error rate.
E_{total}	The general pixels error rate.
L_{ij}	The probability of the neighbor states combination.
$L_t^{(n)}{}_{ij}$	The probability of the neighbor states combination at time slot t after n iterations.
$L_{ij-\tilde{i}\tilde{j}}^{(d)}$	The states combination probability, where d is the direction of the link and $1 \leq d \leq 8$; ij and $\tilde{i}\tilde{j}$ are the coordinates of the hidden states at the two neighbor locations.
$\mathbf{M}$	The number of entries in the observation codebook in the Discrete HMM or SAHMM.
$\mathbf{N}$	The number of states in the conventional HMM ot SAHMM.
O	A sequence of observations.
O_t	The t_{th} observation in a 1-D observation sequence.
$P(O,Q\|\lambda)$	The joint probability of the observation O and a state sequence Q given by the class model λ.
Q	A sequence of active states.
$Q*$	A sequence of optimal active states for a sequence of observations O.

$q_{k,l}$	The active state at the i_{th} row and j_{th} column of an image in 2D SAHMM or HMM.
q_t	The t_{th} active state in the conventional 1-D HMM.
S_i	The i_{th} state in the conventional 1-D HMM or SAHMM.
$S_{i,j}$	The state at the i_{th} row and j_{th} column in the proposed 2D SAHMM state constellation.
V_k	The k_{th} entry in the observation codebook in the Discrete HMM.

Chapter 1

Introduction

1.1 Document Image Analysis

Document image analysis refers to algorithms and techniques that are applied to images of documents to obtain a computer-readable description from pixel data. Data in a paper document are usually captured by optical scanning or camera and stored in a file of picture elements, called pixels, that are sampled in a grid pattern throughout the document. These pixels may have values: OFF (0) or ON (255) for binary images, from 0 to 255 for gray-scale images, and 3 channels of 0 to 255 colour values for colour images. In any case, it is important to understand that the image of the document contains only raw data that must be further analysed to process the information.

In the next stage some further procedures are required, which include: Thresholding to convert a grayscale or colour image to a binary image, reduction of noise to reduce extraneous data, segmentation to separate various components in the image and thinning or boundary detection to enable easier subsequent detection of pertinent features and objects of interest. Thereafter objects of interest, such as characters or pictures in the image, will be recognized and saved for further process.

For gray-scale images with information that is inherently binary such as text or graphics, binarization is usually performed first. The objective of binarization is to automatically choose a threshold that separates the foreground and background information. Considering the complexity of real world images, selection of a good threshold is always difficult.

Document image noise is caused by many sources, including degradation due to aging, photocopying, or during data capture. Many image and signal processing methods have been proposed to reduce noise. After binarization, document images are usually filtered to reduce noise. Salt and pepper noise (also called impulse and speckle noise, or just dirt) is a prevalent artifact in poor quality document images. This type of noise appears in images as isolated pixels or pixel regions of ON noise in OFF backgrounds or OFF noise (holes) within ON regions, or as rough edges on character and graphics components.

Skew detection is usually necessary to rotate the image to zero skew angle before layout analysis is performed. Structural layout analysis is performed to obtain a physical segmentation of groups of document components. Depending on the document format, segmentation can be performed to isolate words, text lines, and structural blocks (groups of text lines such as separated paragraphs or table of contents entries). Functional layout analyses use domain-dependent information consisting of layout rules of a particular page to perform labeling of the structural blocks giving some indication of the function of the block.

Segmentation occurs at two levels. At the first level, if the document contains both text and graphics, these are separated for subsequent processing by different methods. At the second level, segmentation is performed on text by locating columns, paragraphs, words, and characters; and on graphics, segmentation usually includes separating symbol and line components.

There are two main types of analysis that are applied to text in documents. One is optical character recognition (OCR) to derive the meaning of the characters and words from their bit-mapped images. The other is page-layout analysis to determine the formatting of the text, and from that to derive meaning associated with the positional and functional blocks (titles, subtitles, bodies of text, footnotes etc) in which the text is located.

Optical Character Recognition (OCR) usually lies at the core of document image analysis, whose objective is to interpret a sequence of characters taken from an alphabet. The big challenge is that characters of the alphabet are usually rich in shape. In fact, the characters can be subject to many variations in terms of fonts and handwriting styles. Despite these variations, there is perhaps a basic abstraction of the shapes that identifies any of their instantiations. Developing computer algorithms to identify the characters of the alphabet is the principal task of OCR.

1.2 Hidden Markov Model

The Hidden Markov Model(HMM) was a result of the attempt to model speech generation statistically. The main reason for this success is its wonderful ability to characterize the speech signal in a mathematically tractable way.

Typically the HMM stage is proceeded by the preprocessing (feature extraction) stages. Thus the input to the HMM is a discrete time sequence of feature vectors. The feature vectors can be supplied to the HMM either in vector quantized form, which is called Discrete HMM, or in raw continuous form, which, therefore is called Continuous HMM. HMM expertizes in handling the stochastic nature of the amplitudes of the feature vectors which are superimposed on the time stochasticity. The Hidden Markov Model is a finite set of states, each of which is associated with a (generally multidimensional) probability distribution. Transitions among the states are governed by a set of probabilities called transition probabilities. In a particular state, an outcome or observation can be generated according to the associated probability distribution. It is only the outcome (not the state) visible to an external observer, therefore states are hidden to the outside; hence the name 'Hidden Markov Model'.

After several decades of research and development, HMM becomes the predominant approach to the automatic speech recognition. At the same time the applications of HMM are expanded to every field of signal processing, such as image processing, bioinformatics, etc. In this thesis, we focus on the application of Discrete Hidden Markov Model in document image analysis.

1.3 Thesis Objectives

Though document image analysis has been intensively studied in the last decades, there are still big margins for us to further improve the performance at every stage. No system can process the document as well as human being. In this thesis we will focus on binarization and classification stages. First of all, because of the variation of lighting condition, image acquasition methods, it is always difficult to binarize different kinds of real world images with a method. Binarization is the challenge we try to solve in the thesis. At the same time though numerous classifiers have been achieved and developed by now, the most used strategy is to test those methods with some benchmark databases and compare their performances at different aspects, such as recognition rate and speed. For the real world applications, it's normal that signals have more noises and distortions than the benchmark database. Tolerance to noises should be another criterion for the classifier. Here we will proposed a system who is more tolerant to the noise and distortations. A new test

methodology is also introduced in the following chapters.

The major theme of this study focuses on how to expand the implementation of HMM in document analysis fields and how to improve their performance. First, HMM is implemented in the binarization problem and satisfactory results are reported. For such pixel characteristics based binarization methods, high computation cost is alway required, which results in unbearably slow speed. In the proposed method we strive to reduce the computation cost without degrading the performance. We also address an alternative edge based binarization method in this thesis. Because the conventional HMM is a causal system, for HMM the observations in a signal sequence are supposed to be generated in series, which is untrue in many cases. There are some drawbacks in this model, especially when it is utilized as a classifier. To improve the performance of the HMM in the field of classifiers, we propose a new noncausal HMM which is more tolerant to the noise and degradation imposed to the signals. The 1-D Self Adaptive Hidden Markov Model(SAHMM) is also successfully expanded into two dimentional, where high performance is obtained.

1.4 Thesis Organization

This thesis is organized as follows: Chapter 2 details the problems involved in document image analysis. Chapter 3 gives in depth introduction of conventional HMM; some variation of HMMs are also mentioned here. Chapter 4 introduces the HMM based binarization method, some experiement results are also provided here. In chapter 5 an edge based binarization method and its mechanism are demonstrated. Chapter 6 gives the brief review of the technologies in the procedure of OCR. Chapter 6 proposes a new noncausal Self-Adaptive HMM system. An OCR engine based on the proposed model is tested on the MNIST and comparative study of performance of proposed method and conventional HMM are carried out. In Chapter 7 a 2D SAHMM system is introduced and its implementation in the OCR engine is compared with other OCR engines in the term of recognition rate. Chapter 8 provides concluding remarks and details future work.

Chapter 2

Introduction of document analysis and optical character recognition

2.1 Introduction

Digital image analysis is continuously expanding its applications in many areas of science and industry, including medicine, microscopy, remote sensing, astronomy, defense, materials science, manufacturing, security, robotics, etc. Each of these application areas has spawned separate subfields of digital image analysis, with a large collection of specialized algorithms and concepts. Document image analysis is a small division of digital image analysis. It refers to algorithms and techniques that are applied to images of documents to obtain a computer-readable description from pixel data. The objectives of image analysis vary according to the different applications. Generally, the text components or picture components in document images should be extracted and analyzed separately. Therefore, document image analysis consists of the techniques in the fields of image processing and pattern recognition. In general the procedure of documentimage analysis can be divided into several steps:

1. Document acquisition.
2. Pre-processing.
3. Binarization.

4. Page Segmentation(Layout Analysis).

5. Character recognition or Object Recognition.

6. Post-Processing

2.2 Document Acquisition

Data of a document image are usually captured by optical scanning and stored in a file of picture elements, called pixels, which are sampled in a grid pattern throughout the document. Each of the pixels that represents an image stored inside a computer has a pixel value which describes how bright that pixel is, and/or what color it should be. In the case of binary images, the pixel value is a 1-bit number indicating either foreground or background. For a greyscale image, the pixel value is a single number that represents the brightness of the pixel. The most common pixel format is the byte image, where this number is stored as an 8-bit integer giving a range of possible values from 0 to 255. Typically zero is taken to be black, and 255 is taken to be white. Values between 0 and 255 make up the different shades of gray. To represent color images, independant red, green and blue components must be specified for each pixel (assuming an RGB colorspace), so the pixel 'value' is actually a vector of three numbers. Often the three different components are stored as three separate channels, which have to be recombined when displaying or processing.

With the development of new hardware, more approaches are available for images acquisition besides the scanners, such as cameras, video cameras, and other sensors. In such cases the resolution of images are usually smaller than the ones from scanners and some more degradation will be imposed in the images, which are caused by environment lighting, angle of the lenses, etc. Consequently, the quality of images will be degraded greatly and tasks of image processing will be much more challenging.

2.3 Preprocessing

During the image acquisition process, noises are unavoidably mixed in the raw image data. Some other degradations, such as noise caused by the distortions from the lenses or non-uniform lighting sources, also increase the difficulty of document analysis. Some preprocessing, such as image enhancement and restoration techniques, are implemented in this stage to make the image more robust which is a critical to the whole performance. Preprocessing generally consists of a series of image-to-image transformations. It does not increase our knowledge of the contents of the document image,

but it may help to extract it. Such technologies include noise removal[1], image enhancement[2], deblurring, skew detection and correction. In some cases it is necessary to reduce the resolutions of images to improve the speed of the whole process, where the low-pass filters and sub-sampling methods are usually utilized.

2.4 Binarization

Binarization is the process of separating the foreground, such as texts or some other objects of interest, from the background in an image. The binarization process will convert a grey image into an image with only two levels. The first level will indicate foreground objects, such as text, logos or others, while the complementary level will correspond to the background. The foreground is usually represented by 0 and the background by 255, which is the highest luminance for 8-bit images. The extraction of textual content from digital images, which is done in the binarization and page segmentation stages, has received the most attention.

The binarization process is very critical stage of document analysis since the quality of binarized images greatly determines the final result. For example, it is well known that the performance of OCR will be degraded greatly if the characters in the binarized image are broken, blurred or overlapping. There are many uncertain factors that affect the performance of binarization, such as complex signal-dependent noise and variable background intensity, which are caused by non-uniform illumination, shadow, smear, smudge or low contrast.

One can notice that the definition of foreground is very subjective. In general, the grey levels of pixels that belong to the object are substantially different from the grey levels of pixels in the background, so thresholding becomes straightforward and the most effective tool to extract objects from background. Histogram based thresholding methods have been intensively studied since the 1960s[105][106] and are the most applied methods. In this method, the optimal threshold of the entire image or part of the image is determined according to the characteristics of the image's histogram. Thereby a 2-D pixel classification problem is converted into an easier 1-D digital signal processing problem. If the histogram of an image is bimodal, then this method can efficiently extract the foreground from the background. However, this is untrue for many real world digital images due to their inherent complexity. Besides the histogram based methods, many alternative methods have been proposed, such as the attribute-based[111][112][113][114][115][116], neural networks based binarization[164], etc. These algorithms are reported to produce robust performance, however, they are computationally expensive. In many cases, because of the high efficiency of the histogram

based thresholding algorithms, they are combined with other methods to improve speed. Thereby, histogram based algorithms always get the attention of scholars. In this thesis we will introduce two novel binarization methods with different strategies. Our simulation results prove the efficiency of these methods.

2.5 Page Segmentation

After the discrimination of the objects from the background, at the next stage the objects in the forground will be separated from each other for further classification. Segmentation occurs at two levels. At the first level, if the document contains both text and graphics, these are separated for subsequent processing by different methods[142][17]. At the second level, segmentation is performed on text by locating columns, paragraphs, words, and characters; and on graphics, segmentation usually includes separating symbol and line components. Until now an image is typically broken down into its basic components such as an individual character or a graphic element.

Previous work on page segmentation can be broadly divided into three categories: bottom-up[18][19], top-down[20], and hybrid[21]. In a typical bottom-up approach such as the Docstrum algorithm proposed by O'Gorman[19], connected components are extracted first and then merged into words, lines, zones, and columns hierarchically based on size and spatial proximity. Bottom-up methods can handle documents with complex layouts, however, they are time consuming and sensitive to noise.

A typical top-down method, such as the X-Y cuts proposed by Nagy et al.[20], starts with the whole document and splits it recursively into columns, zones, lines, words, and characters. Top-down methods are effective for documents with regular layouts, but fail when the documents have a non-Manhattan structure.

Another problem with X-Y cuts is that the global parameters for optimal segmentation are often difficult to find if prior knowledge is not available. A hybrid method which starts from the top is proposed[21]. First, they oversegment a document into small zones using the X-Y cut algorithm[20]. Then, they use the bottom-up method which groups oversegmented small zones with the same properties into a single zone.

All of the above methods are based on the analysis of foreground (black pixels). As an alternative, white stream methods based on the analysis of background (white pixels) are presented in [22][23]. In these methods, rectangles covering white gaps (white pixels) between foreground are extracted. Foreground regions surrounded by these white rectangles are extracted as zones. A more

comprehensive survey is presented in [24].

2.6 Optical Character Recognition(OCR) or Object Recognition

2.6.1 Introduction

Optical Character Recognition(OCR) stands for the techniques designed to translate images of handwritten or machine-printed characters into machine-editable text, or to translate pictures of characters into a standard encoding scheme representing them (e.g. ASCII or Unicode). It is one of the well studies topics in pattern recognition, since it was assumed to be the easiest task in the optical pattern recogtion, which is realized untrue by scholars.

After segmentation, the images of characters or objects are ready to be sent into classifiers for the classification. At this stage, pattern recognition turns out to be the major task of the process. Pattern recognition has long been studied in relation to many different applications. In the document image analysis field the optical pattern recognition is the implementation of pattern recognition in the optical or image/video information. Character or object recognition usually is the ultimate purpose of document image analysis, at the same time it is also the most challenging step of the whole process. In this thesis, we will focus on character recognition, since the methods used for OCR can be easily expanded to other shape recognition applications. Here we will give an in-depth introduction of technologies used in OCR.

According to the variation of input methods, character recognition can be categorized into on-line character recognition and off-line character recognition. Online recognition is based on the pattern of writing the characters with the aid of hand-held computers such as Personal Digital Assistant(PDAs) or other human-machine interfaces. This method has limited number of applications because of the constraints of machine-man interface. The signals of off-line OCR are any document images with characters. In general, the performance of online OCR is higher than the offline ones. First of all, most of the noise existing in the document images will not occur in the online recognition, such as the degradations caused by the lenses, complex background or nonuniform illuminations. Secondly, some extra information, such as the sequence of strokes is available in the online character recognition, while it is very difficult for the offline OCR to extract such information. Generally, segmentation of the characters from the neighbors is easier for the online OCR. According to different objects, the OCR can be categorized into handwritten character recognition and machines printed character

recognition. Since there are fewer variations, machine printed character recognition is much more easier than handwritten character recognition,

Offline OCR can be further subcategorized into character recognition, word recognition, where only limited vocabulary are permitted. In character recognition, the characters in a string are recognized separately. Since in character recognitions, especially in the cursive character recognition, most of the characters are connected with the neighbors, character segmentation is an essential step to separate the individual characters from the string. The survey of the character segmentation algorithms can be found in paper[25]. After segmentation the individual character will be inputted into a character recognition classifer.

Word recognition is implemented in some applications such as recognition of addresses on the envelops and bank check recognition. In such cases the classifiers are made to recognize the integrated whole word instead of every character. Since the segmentation of characters is omitted, the errors caused by this stage are avoided. The major drawback of this method is that the size of the vocabulary is limited, which hampers implementing this method in many other real world applications.

In general, the process occurring in OCR can be divided into three parts: preprocessing, feature extraction, classification.

2.6.2 Preprocessing

Though it is not essential, in most cases preprocessing technologies are proven useful to make the images of the characters more robust to improve the whole performance of the OCR. Various kinds of methods are available for preprocessing. Though some grey image based OCR engines are proposed, in most of the cases, the character images are binary after binarization. Therefore, here we focus on introducing binary images based techniques, such as normalization, thinning, binary morphology.

Normalization is a term with multiple meaning. In the preprocessing of the OCR, they usually include size normalization, skew normalization, stroke width normalization, and some other nonlinear normalizations, e.t.c.

In many cases, it is essential to unify the sizes of the characters before the feature extraction, because the features extracted usually are sensitive to the sizes of the characters. Image size normalization attempts to obscure scale variations of images presented to a recognizer. It is a transformation of an input image of arbitrary size into an output image of a fixed pre-specified size, while attempting to preserve structual detail. Many size normalization methods[172][173], such as Simple-Scaling Method, Ratio-based, Multirate-based are proposed.

Though there are a few rotation invariant features available for the OCR, some features are not suitbale for every application, because of computation cost issue or other reasons. Skew normalization is used in some of preprocessings in the OCR. First of all, a skew detection algorithm is imposed to the digital image of a character and it determines the angle (possibly zero degrees) by which it was skewed. Thereafter, the character image will be rotated proper angle to remove the skew.

In the stroke width normalization, run-length method is usually used to measure the width of the strokes[174]. Erosion and dilation which will be introcuded in next session are utilized to normalized the stroke width. Stroke width normalization sounds like a trival issue, actually from the images in figure 2.1, one can find the shape(feature) from the two images are different from each other, though the structures of the strokes are the same.

Figure 2.1: Same character with different stroke width

The language of mathematical morphology is that of set theory. Sets in mathematical morphology represent the shapes that are manifested on binary or gray images. The set of all the black pixels in a binary image constitutes a complete description of the binary image. Two sets are invloved in the process of binary morphology, one is the image; the other is the kernel of the convolution, which is called a structuring element and has a defined origin. It is a nonlinear convolution-like operation between two such sets. Binary morphology is extremely important for fast, low-level image matching operations. The most common binary morphology includes dilation, erosion, opening, closing.

Dilation is the morphological transformation that combines two sets using vector addition of set elements. If A and B are sets in N-space E^N with elements $a = (a_1, a_2, ..., a_N)$ and $b = (b_1, b_2, ..., b_N)$, respectively, then the dilation of A by B is the set of all possible vector sums of pairs of elements, one coming from A and one from B.

The dilation of A by B is be noted by A$\oplus$ B and defined by

$$A \oplus B = c \in E^N | c = a + b \; for \; some \; a \in A \; and \; b \in B \tag{2.1}$$

Erosion is the morphological transformation which is opposite to dilation. If A and B are sets in N-space E^N with elements $a = (a_1, a_2, ..., a_N)$ and $b = (b_1, b_2, ..., b_N)$, respectively, then the erosion of A by B is the set

The erosion of A by B is noted by A$\ominus$ B and defined by

$$A \ominus B = c \in E^N | c + b \in A \; for \; every \; b \in B \tag{2.2}$$

The opening is defined as an erosion followed by a dilation using the same structuring element for both operations. Opening an image with a disk-structing element smooths the contours, breaks narrow isthmuses, and eliminates small islands and sharp peaks or capes.

$$A \circ B = (A \ominus B) \oplus B \tag{2.3}$$

On the contrary to opening operator, the closing is defined as an dilation followed by a erosion using the same structuring element for both operations. Closing an image with a disk-structing element smooths the contours, fuses narrow breaks and long thin gulfs, and eliminates small holes, and fills gaps on the contours.

$$A \bullet B = (A \oplus B) \ominus B \tag{2.4}$$

2.6.3 Feature Extraction

Few OCR algorithms carry out recognition by matching the raw bitmap of character images, because this method is obviously sensitive to the variation of illumination, written style, fonts, size and some other issues. Feature extraction method is usually utilized to improve the performance. Feature extraction is a dimensionality reduction methods for signals to be processed in the pattern recognition; in the case of OCR the signal is the character image. The feature is required that after the reduction of dimensionality it should be able to describe the data sufficiently. Actually for classification purpose, the feature should minimize the within-class pattern variability while enhancing the between-class pattern variability.

Enormous feature extraction methods have been studied and proposed[175][176]. A feature extraction method that proves to be successful in one application domain maybe turn out not to be very efficient in another domain. Here we only briefly introduce the most used feature extraction methods. According to the object, the feature extraction methods can be divided into grey image based and binary image base. There are various feature extraction methods available for grey scale images. It includes:

1. Template matching[177][178], where the character image itself is used as a feature vector. In this method some reference images will be saved as template, distances of the target image with the references will be measured in formula 2.5 to find the nearest template. Here x,y are the pixel indexs in the 2D image; the $Z(x,y)$ is the pixel value of the target image; $T_j(x,y)$ is the pixel value of the jth template image.

$$D_j = \sum_{i=1}^{M}\sum_{j=1}^{N}(Z(x,y) - T_j(x,y))^2 \tag{2.5}$$

2. Deformable template[179]. In this way the deformed character skeletons will be used to find the best match.

3. Unitary transformation including Karhunem-Loeve(KL) Fourier[183], Cosine, Sine and Slant transforms.

4. Zoning: In this method the image of character will be divided into grids. Any simple features, such as averaye value of the grey values in the grid, DCT, Gradient values, will be extracted based on every grid.

5. Geometric moment invariants.[180][181][182] Various kinds of moments are available, such as Hu, Bamieh,Zernike,Teague-Zernike, and pseudo-Zernike moment invariant.

The only difference of a binary character image and grey charactere image is that the pixel values in binary images are either zero or the maximum pixel value, which is usually 255. If the binarization method is good enough, the features taken from binary images should be more robust than from grey images. One can tell the feature extraction methods used for the grey images mentioned in last section are all available to the binary images, since binary image can be regarded as a special case of grey image. Some extra feature methods which are only available for binary images are:

1. Projection histograms[184], which is to measure the histogram of the runlenth of strokes in vertical, horizontal, or some other diagonal directions.

2. Contour profiles[185]: The closed outer contour curve of a character is a closed piecewise linear curve that passes through the centers of all the pixels which are 4-connected to the outside background, and no other pixels. Following the curve, the pixels are visited in counter-clockwise order, and the curve may visit an edge pixel twice at locations where the object is one-pixel wide. Each line segment is a straight line between the pixel centers of two 8-connected neighbors. Chain code is usually used to describe the contour of an object.

3. Spline curve approximation[186], where the outer character contour will be divided sline by breakpoints. And the spline curve parameter are used as feature.

4. Elliptic and other Fourier descriptors[187][188], where the contour of the character will be transformed into other domains, where the feature can be more explicitly expressed.

Selection of a feature extraction methods is the most important factor in achieving high recognition performance. Different methods yield different performance with different computation cost and memory requirements. How to find the most promising methods for a specific application is not in the scope of this thesis.

2.6.4 Classification

Numerous techniques for character recognition have been investigated based on four general approaches of classification, as suggested by Jain[26]: template matching, statistical techniques, structural techniques, and neural networks. Such approaches are neither necessary independent nor disjointed from each other. Occasionally, a technique in one approach can also be considered to be a member of other approaches.

Template matching operations determine the degree of similarity between two feature vectors in the feature space. Matching techniques can be grouped into three classes: direct matching [27], deformable templates and elastic matching [28], and relaxation matching [29].

Statistical techniques are concerned with statistical decision functions and a set of optimal criteria, which determine the probability of the observed pattern belonging to a certain class. Several popular handwriting recognition approaches belong to this domain: The k-Nearest-Neighbor (k-NN) rule is a popular non-parametric recognition method, where a posteriori probability is estimated from the frequency of nearest neighbors of the unknown pattern. Compelling recognition results for handwriting recognition have been reported using this approach[199]. The drawback of this method is the high computational cost when the classification is conducted. To surpass such a problem some researchers have proposed faster k-NN methods, where some tricky search strategies such as search tree are utilized. A comparison of fast nearest neighbor classifiers for handwriting recognition is given in [31].

The Bayesian classifier assigns a pattern to a class with the maximum a posteriori probability. Class prototypes are used in the training stage to estimate the classconditional probability density function for a feature vector[32].

The polynomial discriminant classifier assigns a pattern to a class with the maximum discriminant

value which is computed by a polynomial in the components of a feature vector. The class models are implicitly represented by the coefficients in the polynomial[33].

Hidden Markov Model (HMM) is a doubly stochastic process, with an underlying stochastic process that is not observable, but can be observed through another stochastic process that produces the sequence of observations. Because of its two layer architecture, it is suitable to process highly variable spatiotemporal data sequence. Its application has been expanded to almost every field of image processing and machine vision[51][52][80]. Through our study, we have found there are some drawbacks in the conventional HMMs. In this thesis we will focus on some modifications of HMM based classifier to overcome some of the shortcomings.

Support Vector Machine (SVM) is based on the statistical learning theory[34] and quadratic programming optimization. A SVM is basically a binary classifier and multiple SVMs can be combined to form a system for multi-class classification. In the past few years, SVM has received increasing attention in the community of machine learning due to its excellent generalization performance. More recently, some SVM classification systems have been developed for handwritten digit recognition, and some promising results have been reported in [35][36][37].

In structural techniques the characters are represented as unions of structural primitives. It is assumed that the character primitives extracted from handwritten are quantifiable, and one can find the relationship among them. Basically, structural methods can be categorized into two classes: grammatical methods[38] and graphical methods[39].

A Neural Network (NN) is defined as a computing structure consisting of a massively parallel interconnection of adaptative neural processors. The main advantages of neural networks lies in the ability to be trained automatically from examples, good performance with noisy data, possible parallel implementation, and efficient tools for learning large databases. NNs have been widely used in this field and promising results have been achieved, especially in handwriting digit recognition. The most widely studied and used neural network is the Multi-Layer Perceptron (MLP) [40]. Such an architecture trained with back-propagation [41] is among the most popular and versatile forms of neural network classifiers and is also among the most frequently used traditional classifiers for handwriting recognition. See [42] for a review. Other architectures include Convolutional Network (CN) [43], Self-Organizing Maps (SOM)[44], Radial Basis Function (RBF)[40], Space Displacement Neural Network (SDNN)[45], Time Delay Neural Network (TDNN)[46], and Hopfield Neural Network (HNN)[47].

The above review indicates that there are many recognition techniques available for handwriting recognition systems. All of them have their own advantages and drawbacks. In the recent years,

many researchers have combined such techniques in order to improve the recognition results. Various classifier combination schemes have been devised and it has been experimentally demonstrated that some of them consistently outperform a single best classifier[48].

Object recognition has the same strategy as character recognition except the object can be more than characters, and some steps of the recognition process, such as feature extraction, have to been changed. This theme is out of the scope of this thesis.

2.7 Post-Processing

Recently, the majority of research in handwritten word recognition has integrated the lexicon as constraint to build lexicon-driven strategies with character recognition opposite to handwritten word recognition. The lexicon is a list of possible words that could possibly occur in an image. This lexicon is usually determined by the application. It aims at decreasing the complexity of the problem since the ambiguity makes many characters unidentifiable without referring to context. One can notice that such a method is much more flexible than word recognition. Generally the string matching algorithms between candidate words and a lexicon are used to rank the lexicon, often using a variant of the Koch et al[49] to combine contextual information for recognition of handwritten words extracted from real incoming mail documents. The word recognition process is based on three different sources of information: outputs of a character classifier, contextual information extracted from word shapes and some a priori knowledge. The experiments have shown the benefit of the additional information on word recognition rates.

Koerich et al[210] proposed a fast two level HMM decoding algorithm to deal with large vocabulary handwriting. The authors propose a nonheuristic, fast decoding algorithm which is based on a hidden Markov model representation of characters. The decoding algorithm breaks up the computation of word likelihoods into two levels: state level and character level. Given an observation sequence, the two level decoding enables the reuse of character likelihoods to decode all words in the lexicon, avoiding repeated computation of state sequences. In an 80,000 word recognition task, the proposed decoding algorithm is about 15 times faster than a conventional Viterbi algorithm, while maintaining the same recognition accuracy.

2.8 Conclusion

We presented a brief summary of basic building blocks that comprise a document analysis system. Some mainstream solutions for every stage of the procedure are numerated here. Though every step is critical for the whole process, until now the bottlenecks for the document image analysis are binarization and pattern recognition. First of all, because of the infinite variation of real world images, it is nearly impossible to find a real universal binarization method to sufficiently segment any kinds of document images. In the follwoing chapters we will introduce two proposed methods which are capable of handling many kinds of degraded images. Though enormous methods have been proposed for OCR, this problems remains open for better solutions. Some methods[43] report high performance with unbelievable computational burden, and hugh memory storage requirement. Some[43] can efficiently recognize characters in good conditions, while failing when there is slight noise mixed with images. Classification is the major subject of OCR and different of classifiers are general introduced in this chapter. We will also present improved HMM based classifiers in the thesis.

Chapter 3

Review of Hidden Markov Model

3.1 Hidden Markov Model

The 1-D Hidden Markov Models(HMM) were proposed in the 1960s by Baum et al. [93][94][95][96]. HMM has the capability of capturing statistical properties of a wide spectrum of nondeterministic signals, or pseudo-properties from real random signals. Because of its efficiency and flexibility, in recent decades, it has become the predominant approach to automatic speech recognition[98], gesture and body motion recognition[51], optical character recognition[80][205], machine translation[52], bioinformatics and genomics[53].

A Markov chain is a collection of random variables X_t (where the index t runs through 0, 1, ...) with the property that the probability of variable X_t occurring depends on its history states(past states). In most cases the present variable is only directly effected by the last one, thereby it is called first order Markov chain. It can be formulated as below:

$$P(X_t = j|X_0 = i_0, X_1 = i_1, ..., X_{t-1} = i_{t-1}) = P(X_t = j|X_{t-1} = i_{t-1}) \tag{3.1}$$

In a Markov model, the state is directly visible to the observer, therefore, the state transition probabilities are the only parameters. In a HMM, the state is not directly visible, however, observations which can be influenced by the state are visible. Each state has a probability distribution over the possible output observations. So the sequence of observations generated by an HMM give some information about the sequence of hidden states. Since its two layers architecture makes it capable

of handling more complex signals, HMMs are more powerful and much more broadly utilized in real world applications than Markov models.

3.1.1 Architecture of Hidden Markov Model

Different from the Markov chain, the architecture of HMM can be divided into two layers, observations and states, which are illustrated in Figure 3.1. Observations are the data (signals) collected for modeling the HMM, which are visible to external observers; while states are hidden behind the observations and the relationship of hidden states are the same as Markov chain, hence, it is called Hidden Markov Model. Here, another important notation, that is 'time' in the HMM has to be mentioned. Time here is used to annotate the progressive dimension of the signal. Without loss of generality, under the assumption, it is a temporal signal. For any other signal, the time dimension is replaced with the proper alternative dimension. At any given time slot one state is active, which generates the observation corresponding to that instant according to the probability function distribution of the emitting state.

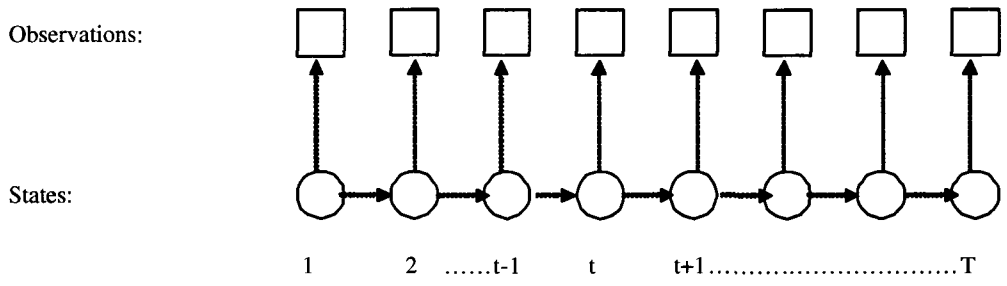

Figure 3.1: Architecture of Hidden Markov Model

In general, a HMM is sufficiently defined by the three parameters: transition matrix A, observation probability matrix B, and initial model probability π. The number of parameters should be predefined as the number of states of the model, N, and the number of observation symbols in the alphabet, M. It should be mentioned that there is not an universal method to determine the 'optimal' N and M for a specific application, though in many cases empirical numbers are very helpful to choose the optimal values.

The possible state transitions are denoted by paths as shown in Figure 3.1. Each path is associated with a transition probability, a_{ij}, which expresses the probability of state S_j occurring after

state S_i in the model. Similar to a Markov chain, the probability for a state S_j to be the active state q_t at the time t is dependent only on q_{t-1}, the active state at time $t-1$, therefore, HMM is also a causal system. A state transition probabilities is formulated as:

$$a_{ij} = p\{q_{t+1} = j | q_t = i\},\ 1 \leq i,\ j \leq N \tag{3.2}$$

where q_t denotes the current state and q_{t+1} denotes the next state.

The transition probabilities among the states in the model are usually expressed in a matrix form. The matrix that contains all transition probabilities of a model is called transition matrix, A.

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1(N-1)} & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2(N-1)} & a_{2N} \\ & & \ddots \quad \ddots & \vdots & \vdots \\ a_{(N-1)1} & a_{(N-1)2} & \cdots & a_{(N-1)(N-1)} & a_{(N-1)N} \\ a_{N1} & a_{N2} & \cdots & a_{N(N-1)} & a_{NN} \end{pmatrix}$$

HMMs can be categorizied into Discrete HMM and Continuous HMM according to the observation probability distribution. The simplest type is the discrete HMM whose states possess discrete probability distributions as functions of discrete observations. In this case, the observations are quantized to discrete values using a predefined finite-length codebook and mapped to its M entries. The probability distribution of observations in each of the states is $B = \{b_j(k)\}$.

$$b_j(k) = p\{O_t = v_k | q_t = j\},\ 1 \leq j \leq N,\ 1 \leq k \leq M \tag{3.3}$$

where q_t denotes the current state; O_t denotes the observation in state t; v_k means that observation is kth symbol in the codebook.

From the above equation one can find that the present state should be deduced by the former one. However, the first time slot is the start point of the whole deduction and there are no states before it, so the probabilities of states in time slot 1 can not be derived from the former states anymore. This problem can be solved by the definition of a new parameter, the initial state distribution $\pi = \{\pi_i\}$, which is the probability of the states occurring in the first time slot. It is formulated as:

$$\pi_i = P[q_1 = S_i],\ 1 \leq i \leq N \tag{3.4}$$

One compact notation λ is usually used to combine all of the parameters mentioned above,

$$\lambda = (A, B, \pi) \tag{3.5}$$

to indicate the complete parameter set of the model.

3.2 The problems of Hidden Markov Model

For a given sequence of observation $O = o_1, o_2, .., o_T$, the relationship of the sequence with a HMM is subject to a certain measure, e.g. likelihood. The likelihood of a process model to generate the observation sequence O indicates to what extent the model fits the modeled process. Hence, the evaluation of the likelihood measure becomes an essential part of the model learning and evaluation. In general, HMM learning is an adaptive process to maximize the likelihood of the training data generated by a HMM. Determining the optimal state sequence by which the model generates the observation is referred to as decoding process. For any given HMM, three problems should be solved before it can be utilized it in any applications, which are:

1. Evaluation Problem.
2. Decoding Problem.
3. Learning Problem.

In the following sections the details about the three problems will be addressed.

3.2.1 Evaluation Problem:

This problem is to estimate the probability of the observation sequence O to be generated by a given model $\lambda, P(O|\lambda)$. Given a model λ, observation sequence O can be generated by any state sequence $Q = q_1, q_2, ..., q_T$ of all the possible state sequences; q_t is the active state in time slot t and T is the length of the sequence. The probability $P(Q|\lambda)$ which is a state sequence Q occurring in a given model λ depends only on the transition matrix, A, and the initial model probability, π, and can be formulated in equation 3.6.

$$P(Q|\lambda) = \pi a_{q_1 q_2} a_{q_2 q_3}, ..., a_{q_{T-1} q_T} \tag{3.6}$$

The probability of the observation at the time t, O_t, to be generated by a given active state at the same time instant t, q_t corresponds to the parameter $b_{q_t}(O_t)$. This probability depends on the observation probability distribution, therefore, the probability of the observation sequence produced by a state sequence and the model can be expressed as:

$$P(O|Q, \lambda) = \prod_{t=1}^{T} P(O_t|q_t, \lambda) = b_{q_1}(O_1) b_{q_2}(O_2) ... b_{q_T}(O_T) \tag{3.7}$$

Hence the probability of the observation sequence O generated by model λ can be:

$$P(O|\lambda) = \sum_{allQ} P(O|Q,\lambda)P(Q|\lambda),\ Q = q_1, q_2, ..., q_T, O = O_1, O_2, .., O_T \tag{3.8}$$

From equation 3.8, one can tell that it is required to add the product over all possible state sequences to calculate the probability $P(O|\lambda)$. The number of available state sequences is N^T, which grows exponentially with the length of the observations sequence. In most of the cases the computational cost will be prohibitive and it is not practically feasible to calculate the likelihood in this way. A much more efficient method, called the Forward-Backward algorithm[93], was proposed. A recursive strategy is utilized in this method and it will be introduced in the next section.

Forward-Backward Algorithm

Before demonstrating the details of the Forward-Backward algorithm, two new auxiliary variables should be introduced: $\alpha_t(i)$, $\beta_t(i)$. $\alpha_t(i)$ is the joint probability that the partial observation sequence $O_1, O_2, \cdots, O_t$ occurs and that the state S_i is the current active state q_t in the model λ.

$$\alpha_t(i) = P(O_1, O_2, ..., O_t, q_t = S_i|\lambda) \tag{3.9}$$

The backward variable $\beta_t(i)$ is defined as the joint probability that the partial observation sequence $O_t, O_{t+1}, \cdots, O_T$ occurs and that the state S_i is the current active state q_t in the model λ.

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, ..., O_T|q_t = S_i, \lambda) \tag{3.10}$$

Forward Phase:

In the forward phase, the forward variable $\alpha_t(i)$ can be deduced in 3 steps:

Initialization:

$$\alpha_1(i) = \pi_i b_i(O_1),\ 1 \leq i \leq N \tag{3.11}$$

Induction:

$$\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i)a_{ij}]b_j(O_{t+1}),\ 1 \leq i \leq N,\ 1 \leq t \leq T-1 \tag{3.12}$$

Termination:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{3.13}$$

The process of forward phase can be illustrated in Fig. 3.2.

Backward Phase:

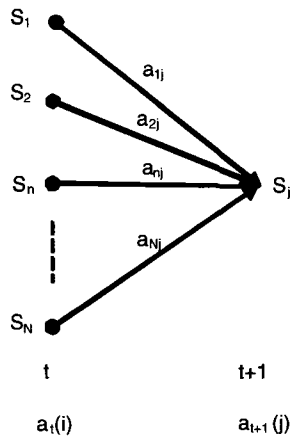

Figure 3.2: Illustration of the computation of the forward variable $\alpha_{t+1}(j)$

In a similar manner, the backward variable $\beta_t(i)$ is recursively computed as follows:

Initialization:

$$\beta_T(i) = 1,\ 1 \leq i \leq N \tag{3.14}$$

Induction:

$$\beta_t(i) = \sum_{i=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j),\ 1 \leq i \leq N \tag{3.15}$$

The β is computed in a lattice structure similar to that of Fig. 3.2, except the propagation direction is different. From equation 3.13, the target of evaluation problem $P(O|\lambda)$ can be easily obtained. Thereby, the forward phase of the forward-backward algorithm is enough to solve the evaluation problem, while backward phase helps in solving the decoding problem as shown in next section.

3.2.2 Decoding problem:

For a given observation sequence O, the decoding problem is to find the optimal state sequence, $Q*$, in the model λ. Due to different optimal criteria, various definitions of the optimal state sequence can be given. Though there are many alternative methods available, forward-backward algorithm is the one most used. This can be achieved by using the forward-backward algorithm to find the state that is most likely to be the active state at each time instant t, q_t, as follows:

First, we define a temporal variable $\gamma_t(i)$:

$$\gamma_t(i) = P(q_t = S_i|O, \gamma) \tag{3.16}$$

This is the probability of the state of index i to be the active state at the time t under the conditions of given observation sequence and model parameters.

The $\gamma_t(i)$ can be computed with the forward-backward algorithm as:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\gamma)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{T} \alpha_t(i)\beta_t(i)} \tag{3.17}$$

With the aid of equation 3.17, q_t is the state that yields maximum $\gamma_t(i)$ value.

$$q_t = \underset{1 \le i \le N}{\operatorname{argmax}}[\alpha_t(i)], \ 1 \le t \le T \tag{3.18}$$

Through equation 3.18, it is easy to find the optimal state sequence. This method have two drawbacks:

First, to obtain an optimal state sequence by forward-backward algorithm, approximatelly N^2T times multiplications and N^2T times additions are needed. The computation complexity is still comparably high. Second, even every obtained state in every time slot is optimal, there is still potential risk that in some cases some probabilities of transitions between two neighbor 'optimal' states are 0, which means that this state sequence is invalid and the result is wrong. This is because that optimal state is calculated independantly and the probability of occurrence of sequences of states is not regarded. The Viterbi algorithm[79] is proposed to solve the above two problems.

Let the state score $\delta_t(i)$ be the observation likelihood maximized over the past state sequence terminating with S_i at time t. Thus, $\delta_t(i)$ can be defined as:

$$\delta_t(i) = \max_{q_1 \cdots q_{t-1}} P[q_1, q_2, ..., q_t = i, O_1 O_2 ... O_t | \lambda] \tag{3.19}$$

Then $\delta_{t+1}(i)$ can be computed by induction as follows:

$$\delta_{t+1}(j) = [\max_i \gamma_t(i) a_{ij}] b_j(O_{t+1}) \tag{3.20}$$

Equation 3.20 is the core of the Viterbi algorithm which chooses the state S_i that results in the highest score when a transition occurs from S_i at time $t-1$ to S_j at time t. A pointer should be kept to such a state so the whole path, i.e., active state sequence, can be retrieved later on. A predecessor parameter $\psi_t(j)$ is defined in order to store the index of the best predecessor state of the current state S_j at time t, i.e.:

$$\psi_{t+1}(j) = \underset{1 \le i \le N}{\operatorname{argmax}}[\gamma_t(i) a_{ij}] \tag{3.21}$$

The algorithm can be scripted as follows:

Initialization:

$$\gamma_1(i) = \pi_i b_i(O_1), \ 1 \le i \le N \tag{3.22}$$

Recursion:

$$\gamma_t(j) = \max_{1 \le i \le N} [\gamma_{t-1}(i) a_{ij}] b_j(O_t),\ 1 \le i \le N\ 2 \le t \le T \tag{3.23}$$

$$\psi_t(j) = \operatorname*{argmax}_{1 \le i \le N} [\gamma_{t-1}(i) a_{ij}],\ 1 \le i \le N\ 2 \le t \le T \tag{3.24}$$

Termination:

$$P* = \max_{1 \le i \le N} [\gamma_T(i)] \tag{3.25}$$

$$q_T* = \operatorname*{argmax}_{1 \le i \le N} [\gamma_T(i)] \tag{3.26}$$

Backtracking:

$$q_t* = \psi_{t+1}(q_{t+1}*), t = T-1, T-2, ...1. \tag{3.27}$$

where Q* is the optimal state sequence.

After the accomplishment of the search for the best state sequence path, the highest likelibood of the observation generated by the state sequence is usually used as approximate solution to the evaluation problem, which can be formulated in equation 3.28. This approximation is mostly used in two dimensional(2D) HMM systems[100][61]. In many cases, such as in bioinformatics[53], the search of the optimal path is the major goal of the whole process.

$$P(O|\lambda) = \max_{All\lambda} P(O, Q|\lambda) = P(O, Q*|\lambda) \tag{3.28}$$

3.2.3 Learning problem:

In the problems of the HMMs, the most challenging and critical one is the learning algorithm, where parameters of the model that maximize the likelihood of the training set is determined. The training set are a set of observation sequences, and the summary of the likelihoods of all of the sequences to a model is usually calculated as the criterion. There is no known way to analytically determine the optimal paramters of a model to produce the observation sequence, however, a local maximum can be reached. Because of its guaranteed convergence, Baum Welch algorithm[97] has become the dominant HMM training method. A detailed introduction of the iterative procedure is presented here.

In this method an auxiliary parameter ξ is introduced, which is the probability of S_i to be active states at time t, and state S_j occurring at time $t+1$.

$$\xi_t(i, j) = P(q_t = S_i, q_{t+1} = S_j|O, \lambda) \tag{3.29}$$

It can be computed in this way:

$$\xi_t(i,j) = \frac{\lambda_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{P(O|\lambda)} = \frac{\lambda_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)} \tag{3.30}$$

Another parameter $\gamma_t(i)$ is added here to describe the probability of being in state S_i at time t, given the observation sequence and the model; hence we can relate $\gamma_t(i)$ to $\xi_t(i,j)$ by summing over j, giving:

$$\gamma_t(i) = \sum_{j=1}^{N}\xi_t(i,j) \tag{3.31}$$

If $\gamma_t(i)$ is summed over the time index t, a quantity of the expected number of transitions made from states S_i can be obtained. At the same time, summation of $\xi_t(i,j)$ over t can be interpreted as the expected number of transitions from state S_i to state S_j. That is:

$$\sum_{t=1}^{T-1}\gamma_t(i) = expected\ number\ of\ transitions from S_i \tag{3.32}$$

$$\sum_{t=1}^{T-1}\xi_t(i,j) = expected\ number\ of\ transitions\ from\ S_i\ to\ S_j \tag{3.33}$$

With the above formulas we can give a method for re-estimation of the parameters of a HMM. Hence the estimated π, A and B are

$$\overline{\pi} = expected\ frequency(number\ of\ times)\ in\ state\ S_i\ at\ time\ (t=1) = \gamma_1(i) \tag{3.34}$$

$$\overline{a_{ij}} = \frac{expected\ number\ of\ transitions\ from\ state\ S_i\ to\ state\ S_j}{expected\ number\ of\ transitions\ from\ state\ S_i} = \frac{\sum_{t=1}^{T-1}\lambda_t(i,j)}{\sum_{t=1}^{T-1}\xi_t(i)} \tag{3.35}$$

$$\overline{b_j(k)} = \frac{expected\ number\ of\ times\ in\ state\ j\ and\ observing\ symbol\ v_k}{expected\ number\ of\ times\ in\ state\ j} \tag{3.36}$$

In this way, the updated parameter $\overline{\lambda}$ can be obtained easily. If we iteratively use $\overline{\lambda}$ as estimated parameters and repeat the reestimation calculation, we can then improve the probability of O being observed from the model, which is proved in [97]. The training stop until some criteria are met, such as the $P(O|\lambda)$ is bigger than the predefined threshold or the time of iteration is bigger than a threshold.

3.2.4 Phases of Hidden Markov Model

Hidden Markov Models are extensively implemented in various fields, such as pattern recognition, data mining, data modeling, etc. When used as a recognizer in pattern recognition field, which is the major concern of this thesis, HMM works in two phases: training phase and evaluation phase:

The Training Phase

Generally, the training phase is to adjust the HMM parameters λ (which are presented in the equations 3.1.1, 3.3, 3.4), so that the given set of observations O(called the *training set*) are represented by the model in the best way for the intended application. This problem can be solved with the Baum-Welch algorithm as mentioned in the last section.

When HMM implemented as a classifier, a multiple classes problem should be handled. Assuming there are total I classes to be discriminated, first of all, I HMMs should be established and initialized. Training set labeled with different classes will be sent to corresponding models for training. Various advanced training methods have been studied recently. These methods not only maximize the likelihood that training data belong to the corresponding classes, the discrimination between the models of the classifiers is also increased. These algorithms include: Maximum Mutual Information (MMI)[54], Maximum A Posteriori (MAP)[55], Minimum Classification Error[56], Optimizing the model structure using Bayesian model merging[57], Model merging and splitting according to an a priori knowledge[58], and model selection based on Discriminative Information Criterion (DIC)[59]. In depth discussion of these methods are out of the range of this thesis.

The Evaluation Phase

Assuming there are I classes in a recognition system, the parameters of every model $\lambda_i = (A_i, B_i, \pi_i)$ have been obtained in the training phase. In this evalution phase given a sequence of observations $O = o_1, o_2, \ldots, o_t$, our aim is to find the maximum $P\{\lambda_i|O\}$.

$$\lambda = \underset{1\leq i\leq I}{\operatorname{argmax}} P(\lambda_i|O) \tag{3.37}$$

Therefore, the *Bays' rule* is used to obtain *Maximum A Postriori Probability*.

$$\lambda = \underset{1\leq i\leq I}{\operatorname{argmax}} \frac{P(O|\lambda_i)P(\lambda_i)}{P(O)} \tag{3.38}$$

Generally, $P(\lambda)$ is regarded as equal for all λs, since all of the classes have an equal chance to occur in a system. Obviously for a specific observation sequence O, the $P(O)$ should be the same for every model. Thereby, the equation 3.37 can be transformed into:

$$\lambda = \underset{1\leq i\leq I}{\operatorname{argmax}} P(\lambda_i|O) = \underset{1\leq i\leq I}{\operatorname{argmax}} P(O|\lambda_i) \tag{3.39}$$

The forward algorithm[95] is usually utilized, which is described in this section 3.2.1. Through the equations 3.11 3.12 3.13, the probability of the observation sequence belonging to every class

$P_i\{O|\lambda_i\}$ can be easily calculated. The computational burden can be:

$$C*_{HL} = (N(N+1)(T-1)+N) \times I \ Multiplication \tag{3.40}$$

$$C*_{HL} = N(N-1)(T-1)I \ Addition \tag{3.41}$$

3.3 Variations of Hidden Markov Model

In the above sections we have introduced some fundamental knowledge of HMM. With the development of HMM, various HMMs are proposed to optimize their implementations in different applications. HMMs can be categorized into different types in different aspects. In the following we will address sevaral of the most common variations of HMM.

3.3.1 Ergodic and Left-Right

According to the characteristics of transition matrix A, HMMs are divided into two types, one of which is ergodic HMM and the other is left to right HMM. An HMM is said to be ergodic if every state is reachable from any other state in a finite number of transitions. The transition of the hidden states in a fully-connected HMM is an example of the ergodic HMM in which every state is reachable from any other state directly. The left to right model is characterized by its transition matrix that has a zero lower-left triangle. The left to right type of HMM has the desirable property that it can readily model signals whose properties change over time-e.g., speech. The fundamental property of all left to right HMMs is that the state transition coefficients have the property:

$$a_{ij} = 0 \ j < i \tag{3.42}$$

A is characterized as below:

$$A = \begin{pmatrix} a_{11} & a_{12} & & \cdots & a_{1(N-1)} & a_{1N} \\ 0 & a_{22} & & \cdots & a_{2(N-1)} & a_{2N} \\ & & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & & \cdots & a_{(N-1)(N-1)} & a_{(N-1)N} \\ 0 & 0 & & \cdots & 0 & a_{NN} \end{pmatrix}$$

One can notice that the initial state probabilities have the property:

$$\pi_i = \begin{cases} 0 & i \neq 1 \\ l & i = 1 \end{cases}$$

In many cases, such as speech recognition and character recognition, additional constraints are imposed about the state transition coefficients to ensure that large states transition do not occur, which means:

$$a_{ij} = 0 \; j < i \text{ or } j > i + \Delta \tag{3.43}$$

For instance, when $\Delta = 1$:

$$A = \begin{pmatrix} a_{11} & a_{12} & & \dots & 0 & 0 \\ 0 & a_{22} & & \dots & 0 & 0 \\ & & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & & \dots & a_{N-1N-1} & a_{N-1N} \\ 0 & 0 & & \dots & 0 & a_{NN} \end{pmatrix}$$

Such constrains are try to make the models more precise in describing the characteristics of real world signals, such as speech or character. In such cases, the sequence of the segments (states) in a signal should rarely be changed, and it is possible that limited number of the segments(states) are skipped.

3.3.2 Discrete and Continuous

HMMs are also classified according to the state probability distribution type. We have introduced the discrete HMM whose states possess discrete probability distributions as functions of discrete observations. In this case, the observations are quantized to discrete values using a predefined finite-length codebook and quantized to its M entries. The probability of a state S_j generating an observation, O_t, is the probability of it generating the associate codebook entry V_k whose index is k. This probability is written as:

$$b_j(O_t) = P\{V_k | q_t = S_j\}, \; 1 \leq j \leq N, \; 1 \leq k \leq M \tag{3.44}$$

Another HMM type is the continuous probability distribution HMM whose states have continuous probability density distributions which, in turn, are functions of continuous observation parameters. In most cases, the general continuous probability function is approximated by a weighted sum of finite numbers of Gaussian distributions. Thus, the probability of a state S_j to generate an observation O_t, is:

$$b_j(O_t) = \sum_{g=1}^{G} C_j^{(g)} G[O_t, \mu_j^{(g)}, \sigma_j^g], \ 1 \leq j \leq N \tag{3.45}$$

In this thesis we will only focus on the implementation and improvement of Discrete Hidden Markov Model. It should be mentioned that all of the HMM based methods introduced later can be converted into corresponding continuous HMM with similar performance.

3.3.3 Other HMMs

Besides the categories mentioned above, other variation of HMMs proposed by scholars. For example, 2D HMM[100], pseudo 2D HMM[61], autoregressive HMMs[79], Null transitions and tied states[79], factorial HMMs[62] e.t.c. Every model has its unique advantage and specific applications. Here we will not describe the details of the different models further, as they are not focus of this thesis.

3.4 Conclusion

In this chapter we present the basic theory of hidden Markov models from the simple concepts to complex variations of HMMs. The architecture and the mechanism of the HMM are discussed in detail. Three basic problems of HMM, including the evaluation problem, decoding problem and learning problem, are discussed here along with the formulated solutions. How to implement the HMM in the pattern recognition field is highlight of this thesis. In the following chapters we will introduce how to expand the application of HMM into binarization and new HMM architecture is proposed to solve the drawbacks of the conventional methods.

Chapter 4

HMM-Based Binarization Method

4.1 Introduction of binarization algorithms

Separating the foreground from the background of an image is a critical process in image analysis. Its purpose is to acquire some useful information in the image for further processing. In many cases for the images to be processed, the gray levels of pixels of the object are substantially different from the gray levels of the pixels belonging to the background. Therefore, in some cases it is possible to discriminate the objects from the background with a simple and effective thresholding method. Such methods are broadly implemented in document image analysis such as character extraction, form extraction, as well as some applications for medical images, for example endoscopic images, laser scanning and confocal microscopy. It should be mentioned that the threshold base method is only one of the feasible binarization methods.

After this operation an image will be divided into two states: one state will indicate the foreground objects, while the complementary part will correspond to the background, which will be ignored. Depending on the application, the foreground can be represented by gray-level 0, that is, black as text, and the background by the highest luminance for document paper, that is 255 in 8-bit images, or conversely the foreground by white and the background by black. Hence this process is usually called binarization. Various factors, such as nonstationary and signal dependent noises, nonuniform illumination, ambiguity of gray levels within the object and its background, low contrast, and unknown shape and orientation of the objects complicate the binarizing operation.

Many methods have been reported about the binarization. The survey of relevant studies can be found in [103][104]. We will briefly introduce them in the following sections.

4.2 Survey of binarization techniques

The binarization methods reported in the literatures can generally be categorized into four different types, which are:

1. Histogram based methods[105][106].
2. Clustering-based methods[107][109].
3. Object attribute-based methods[111][112].
4. Discrimination based on local pixel's characteristics[164][162].

Histogram based methods are further divided into two types: histogram entropy based algorithms[126] and histogram shape based algorithms[135]. Histogram entropy based algorithms consider certain measures of the entropy of the original image and that of the binarized image. Various types of entropy based binarization have been proposed, which include entropic thresholding[126][127], fusion of three different entropies[128], co-occurrence matrix for second order entropies[129], normalized entropy[131], utilization of information measure[132], entropic thresholding block source model[133], and minimum cross entropy[130][134]. It should be noted that although all of the entropies addressed above cannot be mathematically proven to be efficient for the real world images, in many cases they are still available.

The shape of the histogram can be used to determine the threshold level for the binarization of the images. These methods use peak detection[135], valley-seeking threshold selection[136], histogram concavity analysis[137], histogram shape analysis[138], valley detection using wavelet transform[139], histogram modification by enhancing the concavity[140], and histogram modification via partial differential equations[141]. The bimodality of the histogram of the image is required for such kind of methods, which is untrue for most of the real world images. Therefore the methods have limited applications too.

The above methods are regarded as global thresholding methods, which yield good performance when the histogram of the image is clearly bimodal. Unfortunately, this attribute can be missing for images with nonuniform backgrounds or when images are degraded by noise. For these types of images, local histogram analysis methods have been proposed[142]. Local histogram methods

divide an image into different zones through layout analysis so that the pixels in one zone are homogeneous. These methods are known to outperform the global methods. The major drawbacks of the local approaches are that it is difficult to determine the correct window size and the block effect. Some additional algorithms with higher complexity have also been proposed, such as adaptive[143], iterative[144][145] and multi-level thresholding methods [146][147].

In the clustering-based methods, the gray-level samples are clustered in two parts as background and foreground, or alternatively are modeled as a mixture of two Gaussians iterative thresholds. The Otsu method[107] is the most referenced thresholding method. In this method the weighted sum of within-class variances of the foreground and background pixels should be minimized to establish an optimum threshold. The minimization of within-class variances is equivalent to the maximization of between-class scatter. This method gives satisfactory results when the number of pixels in background and foreground are similar. Additional methods include iterative thresholding[147], minimum error thresholding[109][110][148][149] and fuzzy clustering thresholding[150]. In [123] the cluster is based on the information contained in a small window around each pixel.

Object attribute-based methods search for a measure of similarity between the gray-level and the binarized image. The threshold value is based on some attribute quality or similarity measure between the original image and the binarized version of the image. These attributes can take the form of edge matching[111], shape compactness[151][152] or gray-level moments[153][154][155]. Other algorithms directly evaluate the resemblance of the original gray-level image to binary image with fuzzy measure[157][158][159] or resemblance of the cumulative probability distributions[160], or in terms of the quantity of information revealed as a result of segmentation[161].

For local pixels adaptive algorithms, a threshold is calculated at each pixel, which depends on some local statistics like range, variance, or surface-fitting parameters of the neighboring pixels. A surface fitted to the gray-level landscape is used as a local threshold, as in Yanowitz and Bruckstein[162] and Shen and Ip[163]. Conventional classifiers such as Neural Networks[164] are introduced to discriminate the pixels into background and foreground according to the characteristics around every pixel. The major drawback of such a method is that binarization is based on the feature around every pixel in an image, therefore the computational cost is much higher than others.

It should be mentioned that all of the algorithms described above are based on the hypothesis that the foreground is much darker than the background, which is true in most of the cases. Though much effort has been devoted to the binarization, binarization from noisy document images is still a big challenge because of the complexity of real world images.

4.3 The Proposed Binarization Method

Here we will introduce a new HMM based binarization method, which belongs to local pixel characteristics based method. Since OCR oriented document analysis technique is the major concern of this thesis. The binarization method focuses on extracting the characters from background, even when the background is noisy, which is difficult for other binarization methods.

This method is called pixel's characteristics based method. After features are extracted from every pixel's neighborhood, the feature will be inputted into a classifier to clarify the pixel belonging to foreground or background. Consequently, the computational cost will be extremely high in comparison with histogram method, therefore how to reduce the computational burden is the major concern of this algorithm.

The proposed algorithm is composed of two stages. In the first stage, a coarse global thresholding method is used to discriminate the brighter part of the whole image from the foreground pixels which have lower values. Thus part of the background pixels are eliminated. Thereafter, the remaining pixels are supposed to be the mixture of all of the foreground and part of the background. In the second stage, the remaining pixels are applied to the HMM based pixel classifier to obtain the attribute of each pixel. In doing so, only part of the whole image pixels are needed for further testing, so the whole processing time will be minimized without sacrificing quality.

4.3.1 The first stage

Since the first stage of the proposed technique is regarded as a pre-processing of the binarization, the speed issue is the major concern in this stage instead of accuracy. The only requirement to this stage is that the threshold should be high enough to avoid misclassifying some foreground into background. The mean value can be regarded as the preliminary threshold value of the whole process, and calculated for 8 bits gray-level images as:

$$Mean = \frac{\sum_{i=0}^{255} ih(i)}{\sum_{i=0}^{255} h(i)} \tag{4.1}$$

where $h(i)$is the number of pixels in the image with grey-level i where

$$0 \leq i \leq 255 \tag{4.2}$$

Since this method does not take into account histogram shape, the results are obviously suboptimal. For document images, the number of pixels in the objects of the foreground is usually much

smaller than the number of pixels in the background. Thus, the mean value will result in under thresholding, which is required in this case.

4.3.2 The Second Stage

Feature Extraction

Instead of only considering the pixel value as the feature in the conventional histogram based binarization method, in the pixel classification strategy, binarization is based on the features around each pixel, such as range, variance, or surface-fitting parameters of the pixel neighborhood. A proper feature is critical for the performance of classification algorithms. Selected features utilized by other researchers [166][167] for classification include intensities, statistical features, feature from gradient and compass operators, local contrast feature and moment.

In this chapter we will propose a novel feature extraction method suitable for HMM based image binarization. Inside a window of N by N, we take four feature vectors V_1, V_2, V_3, V_4 around every candidate pixel from four directions: horizontal, left up to right down, vertical, right up to left down, around each of which contains N elements. Various sizes of windows have been tested. Our simulation results show smaller window sizes yield smoother strokes with less computation cost. However, it does not eliminate some specks in the background and makes the method vulnerable to noise. For the value of $N = 7$, the window is shown in Table 4.1.

In the first vector, the 7 elements are obtained as below:

$$\begin{aligned}
v_1(0) &= (P_{(0,-3)} + P_{(0,-2)} + P_{(0,-1)})/3 - P_{(0,0)} \\
v_1(1) &= (P_{(0,-2)} + P_{(0,-1)})/2 - P_{(0,0)} \\
v_1(2) &= P_{(0,-1)} - P_{(0,0)} \\
v_1(3) &= P_{(0,0)} \\
v_1(4) &= P_{(0,1)} - P_{(0,0)} \\
v_1(5) &= (P_{(0,1)} + P_{(0,2)})/2 - P_{(0,0)} \\
v_1(6) &= (P_{(0,1)} + P_{(0,2)} + P_{(0,3)})/3 - P_{(0,0)}
\end{aligned}$$

Similar features using the other 3 directions can be extracted as follows. Except the center element in which the original pixel value is used, other elements are chosen as the difference of the average value of the sliding window to the central pixel. The general expressions are shown as in equations (4.3-4.6), where j begins from 1 when $i > 3$, and j begins from -1 when $i < 3$. By taking

Table 4.1: Demonstration of feature extraction

P(-3,-3)	P(-3,-2)	P(-3,-1)	P(-3,0)	P(-3,1)	P(-3,2)	P(-3,3)
P(-2,-3)	P(-2,-2)	P(-2,-1)	P(-2,0)	P(-2,1)	P(-2,2)	P(-2,3)
P(-1,-3)	P(-1,-2)	P(-1,-1)	P(-1,0)	P(-1,1)	P(-1,2)	P(-1,3)
P(0,-3)	P(0,-2)	P(0,-1)	P(0,0)	P(0,1)	P(0,2)	P(0,3)
P(1,-3)	P(1,-2)	P(1,-1)	P(1,0)	P(1,1)	P(1,2)	P(1,3)
P(2,-3)	P(2,-2)	P(2,-1)	P(2,0)	P(2,1)	P(2,2)	P(2,3)
P(3,-3)	P(3,-2)	P(3,-1)	P(3,0)	P(3,1)	P(3,2)	P(3,3)

this approach the computation cost is reduced in comparison with the conventional methods such as variance[124] and moments[143].

$$v_1(i) = \sum_{j=1or-1}^{|i-3|} \frac{P_{(0,j)}}{|i-3|} - P_{(0,0)}, \quad -3 \leq i \leq 3, \ i \neq 3 \tag{4.3}$$

$$v_2(i) = \sum_{j=1or-1}^{|i-3|} \frac{P_{(j,j)}}{|i-3|} - P_{(0,0)}, \quad -3 \leq i \leq 3, \ i \neq 3 \tag{4.4}$$

$$v_3(i) = \sum_{j=1or-1}^{|i-3|} \frac{P_{(j,0)}}{|i-3|} - P_{(0,0)}, \quad -3 \leq i \leq 3, \ i \neq 3 \tag{4.5}$$

$$v_4(i) = \sum_{j=1or-1}^{|i-3|} \frac{P_{(j,-j)}}{|i-3|} - P_{(0,0)}, \quad -3 \leq i \leq 3, \ i \neq 3 \tag{4.6}$$

To make the features more independant of the illumination variation, we can find the range of candidate pixel values after the thresholding to be between a minimum value in the given image $Mini$ to the mean value of the image pixels $Mean$. In this case:

$$v_i(3) \in [Mini, Mean], \ i = [1, 4] \tag{4.7}$$

An enhancement process is proposed as follow:

$$v_i^{'}(3) = (v_i(3) - Mini)^2 \times \frac{255}{(Mean - Mini)^2} \tag{4.8}$$

Noting that 255 is the largest pixel value.

Through the observation of Fig. 4.1-Fig. 4.3, one can deduce that the 3D surface around pixels in a stroke shown in Fig. 4.3 is completely different from pixels in the background (see Fig. 4.1

Fig. 4.2). Since the features from the foreground and background are different from each other, it is possible for the HMM based recognizer to discriminate between them.

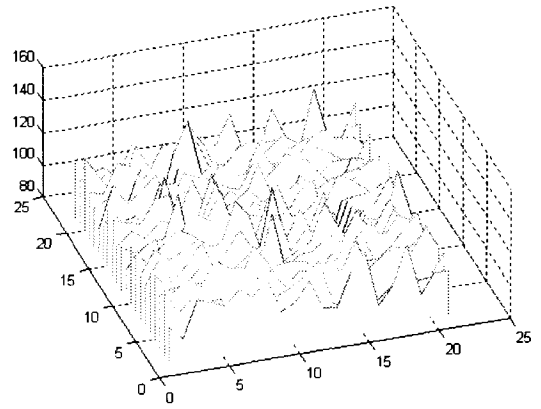

Figure 4.1: Pixel values in a noisy background

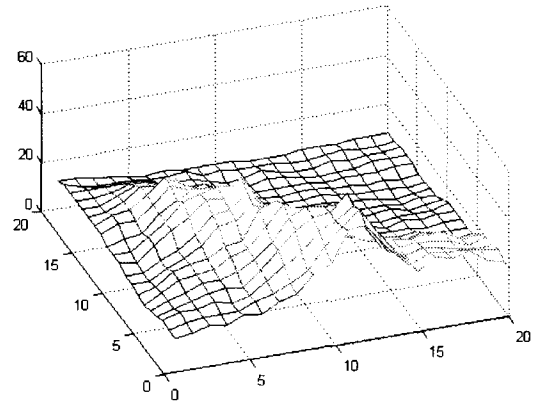

Figure 4.2: Pixel values in a dark background

Hidden Markov Model based binarization

After taking the features around the pixels, we can easily put the features into HMM based classifiers to check the attribute of every pixel through the classification result . The HMM based recognizer works in two phases, training phase and evaluation phase.

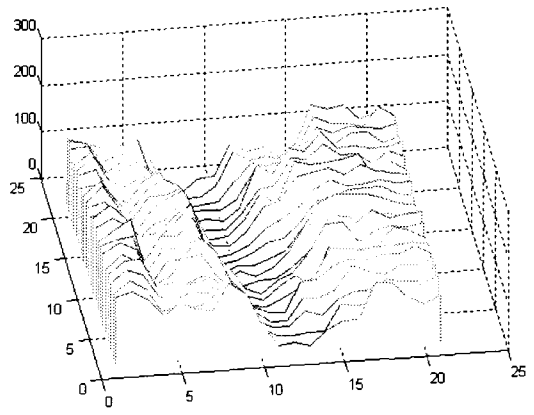

Figure 4.3: Pixel values in one segment of a stroke of a character.

The Training Phase

In the previous section we have explained how features from a pixel should be computed using equations (4.3-4.6). We have scanned images from various sources such as newspaper and magazines to setup a database. Twenty images from the database are randomly selected. From these images 40,000 pixels are selected and their feature vectors are extracted for training. Half of the pixels are in strokes of the characters in the foreground, while the other half are from various backgrounds. Each pixel is assigned to four feature vectors, therefore 160,000 vectors are generated to form a code book.

The K-Mean algorithm[165] is used to segment this vector space into 10 partitions represented by a set of cluster centers. Then the 4 feature vectors for each pixel can be quantized into one observation sequence with 4 elements according to their distances to the 10 cluster centers. All of the observation sequences will be inputted into the HMM to estimate the parameters of the HMM with the Baum-Welch algorithm[93]. Here we select the number of observations M to be 10; the number of hidden states N is 4 and the length of the sequence is 4 as mentioned before.

For the conventional HMM, during the evaluation phase the observation sequence for every pixel should be inputted to the HMM to calculate the probability of the sequence belonging to a class. In this application the length of the sequence is only 4 and the number of observations is 10, therefore the total number of the possible observations is 10000. The observation sequences vary from [0 0 0 0] to [9 9 9 9]. At the same time there are only two classes to be identified - background and foreground. Therefore it is possible to implement some strategy to speed up the whole process,

which is explained as below:

All of the observation sequences from [0 0 0 0] to [9 9 9 9] are inputted into the trained HMM engine and the classification results will be saved in a reference set with 0 representing the background and 1 the foreground. The size of the reference set is only 10K bits, which is affordable for most of the applications.

The Evaluation Phase

To binarize an inputted image, first of all features of every pixel will be extracted. Then the feature vectors are quantized into 10 observations through comparison of their distances to the centers of the 10 clusters determined in the training stage. Instead of calculating the probability for each observation sequence, the recognition result can be looked up in the reference set to obtain the recognition result with an efficient computational cost. Without the look up table, from equations 3.40 3.41, one can find the computation cost for each pixel's classification is 64 multiplications and 36 additions. Assuming the size of a processed image is 1024 by 768 and half of the pixels have been eliminated in the first stage (coarse thresholding), the total HMM based classification in this stage would need 25,165,824 multiplications and 14,155,776 additions.

With the aid of the Look Up Table(LUT) saved in the reference set, all of the calculations can be skipped. Through this method, only feature extraction and quantization are required in the evaluation phase, while the classification time is ignorable, since they can be looked up in the reference set. The function of the HMM is to set up the reference set through limited number of training samples.

4.4 Simulation Results

To test the performance of the proposed binarization technique, a comparative study is carried out. In our experiment, Otsu global thresholding[107], local adaptive thresholding[108] and Kittler thresholding algorithms[110] are utilized as benchmarks for this comparison, because these three methods are reported to have stable and good performances at various applications[103][121].

Our simulation results show the HMM based segmentation method can provide the satisfying results in all of the tested images while others fail to provide acceptable results for some of the images in our database. To test the robustness of the proposed algorithm in presence of noise sources we have conducted the same comparative studies with the three binarization techniques.

4.4.1 Database used

To test the efficiency of the proposed algorithm, three kinds of noisy images were tested. The first picture in Fig. 4.4 is a historical document image with blurred strokes, complex signal-dependent noises and low contrast. The histogram is shown in Fig. 4.5.

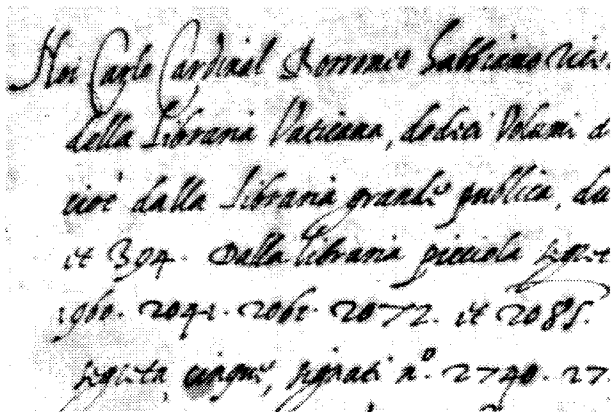

Figure 4.4: An original historical document image with low contrast and signal-dependent noise

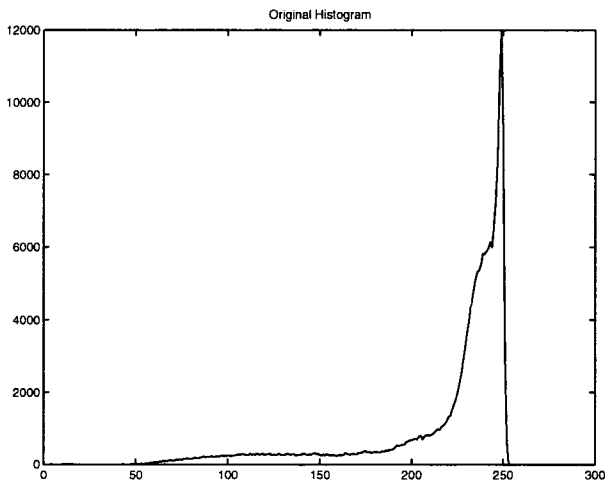

Figure 4.5: The histogram of a gray historical document image with low contrast and signal-dependent noise

The other image shown in Fig. 4.6 has inhomogeneous and complex background. From the histogram shown in Fig. 4.7, one can note the lack of bimodality which is a fundamental requirement for a global thresholding algorithms.

Figure 4.6: An original document image with an inhomogeneous background

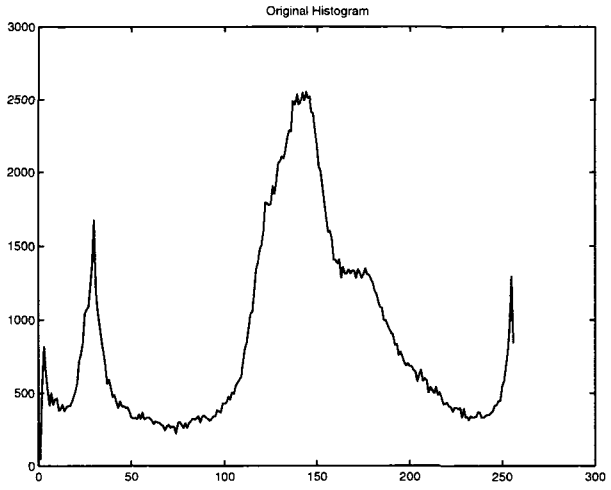

Figure 4.7: The histogram of a gray document image with an inhomogeneous background

The picture in Fig. 4.8 is an image taken by camera with varying illumination in different parts of the image. We can find the distribution of the histogram in Fig. 4.9, which is obviously not bimodal.

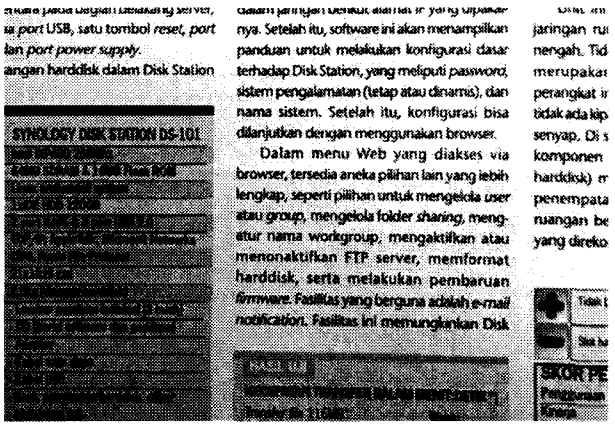

Figure 4.8: An original document image under bad illuminating condition

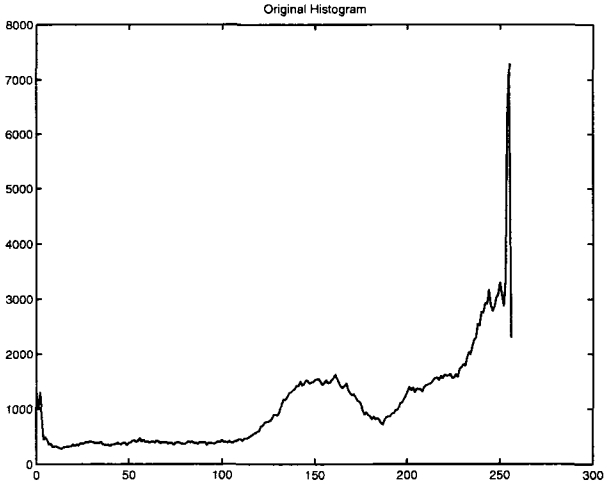

Figure 4.9: The histogram of a gray document image under bad illuminating condition

4.4.2 Comparison of the binarization results

For the image in Fig. 4.4, the Kittler algorithm completely failed. The binary images obtained with the Otsu and Local thresholding algorithm are shown in Fig. 4.10, Fig. 4.11, which are too noisy for any character recognition scheme. From Fig. 4.12 we can see that the binary image from HMM based binarization algorithm is clearer.

The binarization results of the image in Fig. 4.6, which has a complex background, is shown in Fig. 4.13, Fig. 4.14, Fig. 4.15 and Fig. 4.16. We can observe that the Kittler algorithm and HMM based binarization have identical performance, while others yield unsatisfactory results. Actually

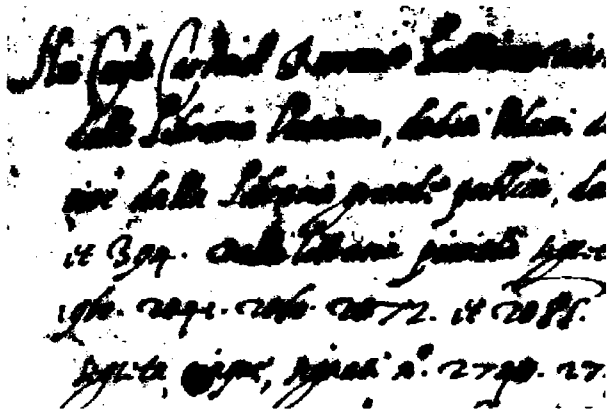

Figure 4.10: Binary document image extracted with the Otsu algorithm from the original image of Fig. 4.4

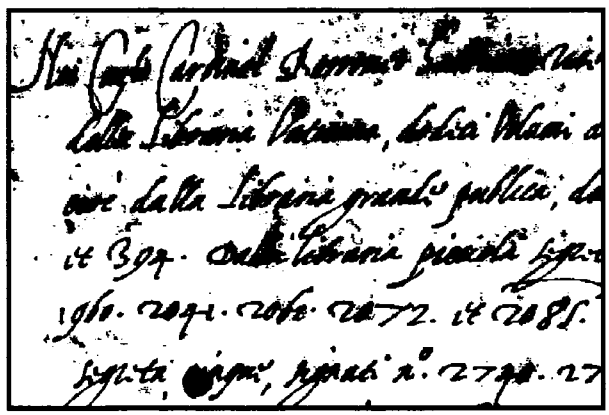

Figure 4.11: Binary document image extracted with the local thresholding algorithm from the original image of Fig. 4.4

the Kittle always over-thresholds the images, therefore it gets comparable good result in such images with noisy background.

The binarization results of image in Fig. 4.8, which has non-uniform illumination is shown in Fig. 4.17, Fig. 4.18, Fig. 4.19 and Fig. 4.20. One can clearly observe that the proposed algorithm performed better than others.

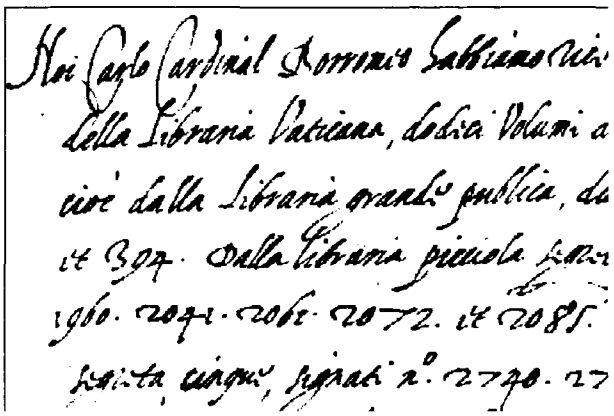

Figure 4.12: Binary document image extracted with the HMM based thresholding algorithm from the original image of Fig. 4.4

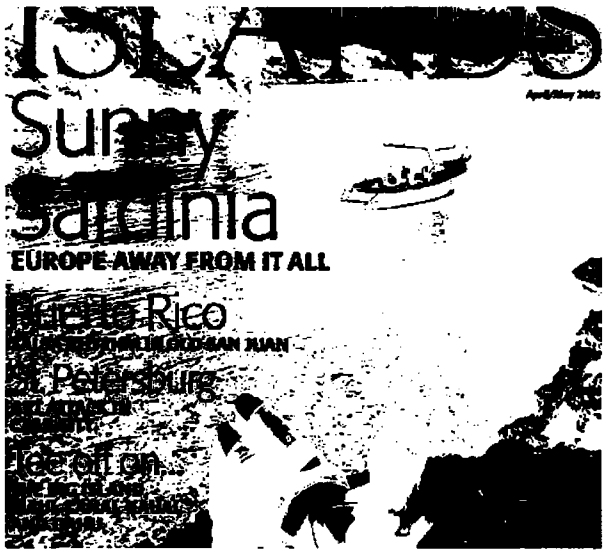

Figure 4.13: Binary document image extracted with the Otsu algorithm from the original image of Fig. 4.6

4.4.3 Quantitative Study

A handful of binarization performance criteria[168][169][170] have been proposed, where OCR based criteron is one of the most acceptable methods. To further prove the efficiency of the proposed binarization method, we randomly selected 42 images from the database of ICDAR[171] for our test. It should be mentioned that all of the methods used here work on gray images and the foreground

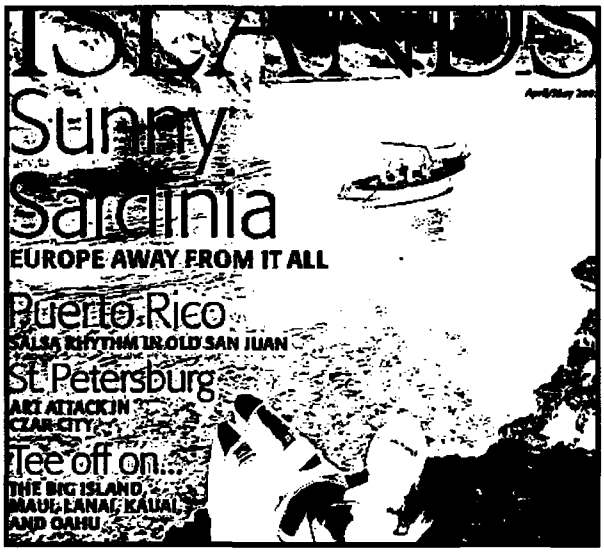

Figure 4.14: Binary document image extracted with Local thresholding from the original image of Fig. 4.6

Figure 4.15: Binary document image extracted with the Kittler algorithm from the original image of Fig. 4.6

pixel values are considered to be darker than the background, thus color images are converted into gray images before binarization.

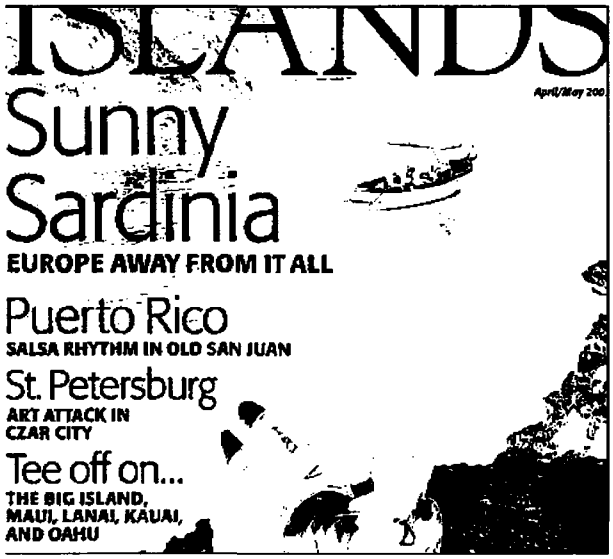

Figure 4.16: Binary document image extracted with the HMM based thrsholding algorithm from the original image of Fig. 4.6

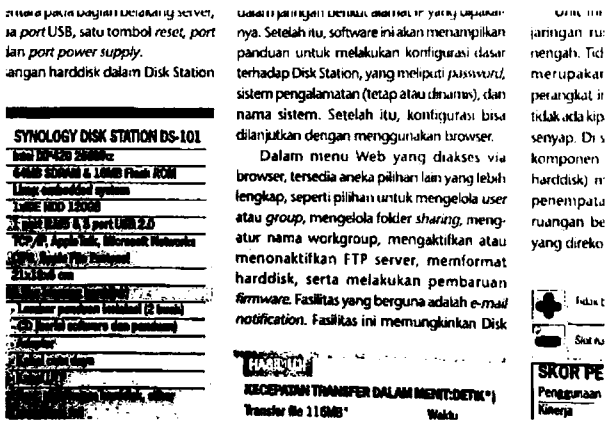

Figure 4.17: Binary document image extracted with the Otsu algorithm from the original image of Fig. 4.8

After binarization the binary images are sent to the commercial OCR software Readiris 10.04 Professional, the recognition results can be regarded as a reliable criterion to evaluate the performances of different binarization methods. There are a total of 1078 characters in the tested images. The OCR test results are reported in table 4.2.

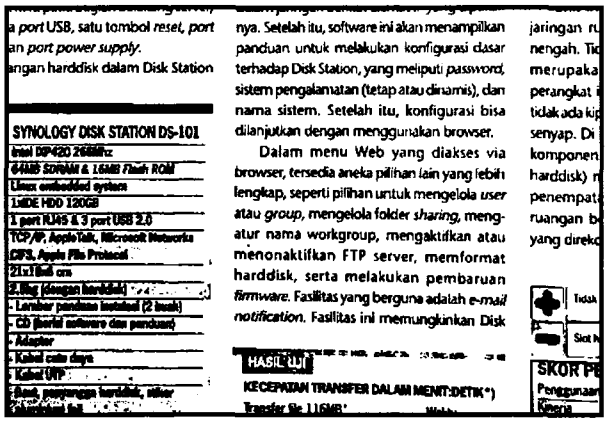

Figure 4.18: Binary document image extracted with the Local thresholding algorithm from the original image of Fig. 4.8

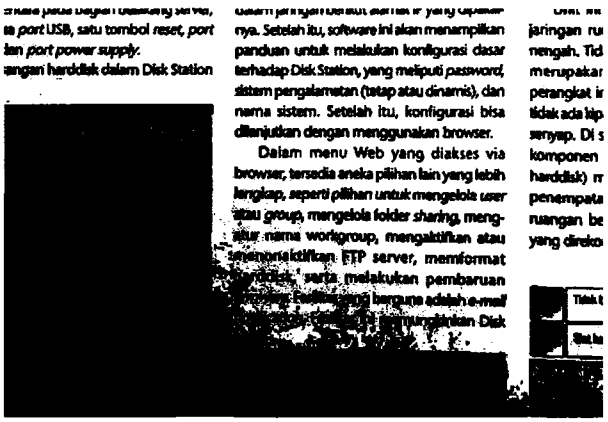

Figure 4.19: Binary document image extracted with the Kittler algorithm from the original image of Fig. 4.8

4.5 Conclusion

For the local pixel's characteristics based binarization method, the major drawback is that the computation cost is extremely high, which hampers the implementation of these methods in real world applications. In this proposed algorithm, two critical strategies are utilized to minimize the processing time. After the preprocessing in the first stage, most of the background pixels are identified and it is unnecessary to send them to a classifier. In the classification state, after feature extraction and quantization, the feature sequence of every tested pixel can be looked up in the

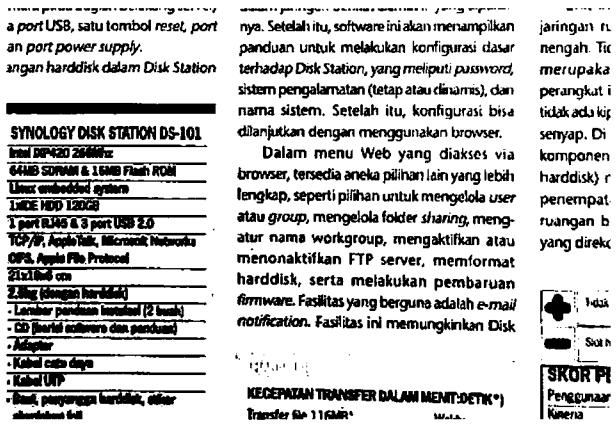

Figure 4.20: Binary document image extracted with the HMM based thresholding algorithm from the original image of Fig. 4.8

Table 4.2: OCR results from different binarized images

Method	HMM based binarization	Kittle[110]	Local[108]	Otsu[107]
OCR recognition rate	77%	25%	48%	53%

reference set, which is established in the training stage. It only takes approximately 2 seconds to binarize an image with the size of 1280 by 768, though the processing time varies from image to image according to the variation of the number of pixels eliminated in the first stage. This method is feasible for most of the real world applications because of its low computatio cost and accuracy. The size of the reference set is only 10k bits, which is also acceptable to real world applications too.

In terms of performance, the test results conducted on number of noisy images yielded satisfactory results. A comparative study of the proposed technique with three different binarization scheme was also carried out. The result of the extracted characters from all binarization techniques were applied to a commercial OCR engine and shows that the proposed method outperforms some of the referenced methods. This can be attributed to the fact that in this method local features around every pixel are selected for binarization. As can be seen, the characteristics of a neighborhood around a pixel in the background illustrated in Fig. 4.1 and Fig. 4.2 are completely different from the pixels in strokes shown in Fig. 4.3. This makes it easy to distinguish between background and pixels in characters. We also notice when the contrast of an image is too low, our method fail to obtain the characters. In some cases the touching characters can not be recognized by the OCR engine. As a result, a recognition rate of 77% is obtained using the proposed technique while the closest

performance was that of Otsu's which yielded 53% correct rate. All programs for comparison of different methods with our proposed HMM based binarization method were written with MATLAB. Since no optimization of the processing time was carried out, we did not include computer time for each technique in the table 4.2.

Chapter 5

Edge-Based Binarization Method

5.1 Proposed Methodology

In the last chapter we introduced a classification based binarization method whose high efficiency has been demonstrated. In many cases simple histogram based (thresholding) methods are required, because of its low computation cost and zero memory storage requirement. However such methods are based on the hypothesis that the histogram of the handled image is bimodal, which is untrue for most of the real world images. Therefore, it is comparably difficult to choose a proper threshold to separate the foreground from a complex background. Through observation, one can notice that no matter how complex the backgrounds are, the values of pixels around the boundary of the foreground and background change abruptly. Our simulations show that any simple edge detector can easily distinguish the boundary of the objects from an inhomogeneous background. Although in some conditions the boundaries are not complete and some edges are derived from variable backgrounds, it has little negative effect to our application since dominant edges are still from the boundary of the foreground objects. Since the edges are the transitions from the foreground to background, the neighbor pixels around every edge are the mixture of the foreground and background. With a simple process we can locate the foreground and background pixels around the edges. Through the histograms of the foreground pixels and background pixels in an image, an optimal threshold to the entire image can be obtained. Later, in this way we can find that we have a much better chance to get a bimodal histogram no matter how complex the background is. In some sense this algorithm

Table 5.1: Horizontal orientational kernel of Prewitt filter

-1	-1	-1
0	0	0
1	1	1

Table 5.2: Vertical orientational kernel of Prewitt filter

-1	0	1
-1	0	1
-1	0	1

can be regarded as histogram enhancement technique.

5.2 The Proposed Edge Based Binarization Method

According to the above analysis, we here propose an edge based binarization method. In this method edges in an image are extracted; then pixels around boundaries of the objects will be taken to select pixels representing the foreground and background pixels. The histogram of the representative pixels will be processed to find the proper threshold. The outline of the process is as shown below:

1. Edge detection.
2. Foreground and background pixels localization.
3. Analysis of histograms of foreground and background.

5.2.1 Step 1: Edge detection

In this algorithm the edge detection is performed base on Prewitt detector[156], whose kernels as shown in table 5.1 and table 5.2. Since all of the nonzero coefficients in the kernels are 1s, no multiplications are required for this process. Consequently, this edge detection has the high efficiency.

Therefore, the algorithm is summarized by the following notation; let $I[i,j]$ denotes the image. The gradient of the image $I[i,j]$ can be computed using the first-difference approximations to produce two arrays $P[i,j]$ and $Q[i,j]$ for the x and y partial derivatives:

$$P[i,j] = \frac{\sum_{k=-1}^{1} I[i+k, j-1] - \sum_{k=-1}^{1} I[i+k, j+1])}{6} \tag{5.1}$$

$$Q[i,j] = \frac{\sum_{k=-1}^{1} I[i-1, j+k] - \sum_{k=-1}^{1} I[i+1, j+k])}{6} \tag{5.2}$$

The magnitude of the gradient can be computed from the standard formulas for rectangular-to-polar conversion.

$$M[i,j] = \sqrt{P[i,j]^2 + Q[i,j]^2} \tag{5.3}$$

Thereby, the problem of finding locations in the image array where there is rapid change has merely been transformed into the problem of finding locations in the magnitude array $M[i,j]$, which are local maxima. A proper threshold is essential to determine the edges in an image according to their magnitude. Many thresholding methods in edge detection have been discussed in [156]. Here the simplest root mean square(RMS) value is utilized as shown in equation 5.4.

$$T_{edge} = \sqrt{\frac{4 * \sum_{i=1}^{width} \sum_{j=1}^{height} M(i,j)}{width * height}} \tag{5.4}$$

In spite of its low contrast and stroke-dependent noises, from the original image shown in Fig. 5.1, we can explicitly abstract the edges from the blurred image as shown in Fig. 5.2.

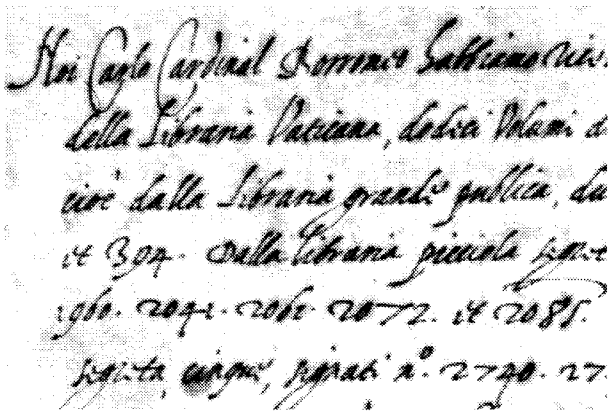

Figure 5.1: An original historical document image with low contrast and signal-dependent noise

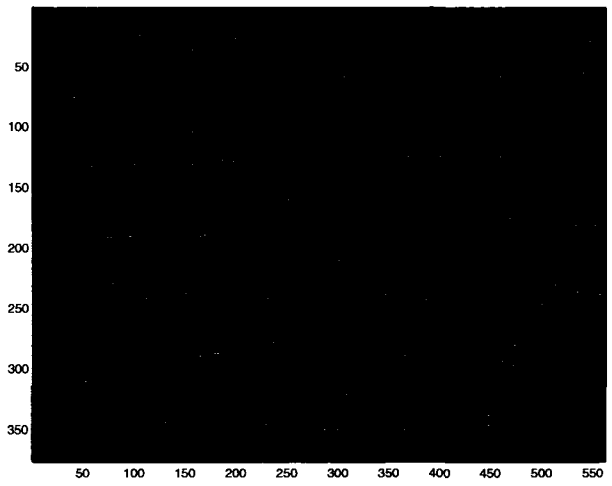

Figure 5.2: Edges detected by the Prewitt detector from Fig. 5.1

Table 5.3: Horizontal orientational kernel

I(i-1,j-1)	I(i-1,j)	I(i-1,j+1)
I(i,j-1)	I(i,j)	I(i,j+1)
I(i+1,j-1)	I(i+1,j)	I(i+1,j+1)

5.2.2 Step 2: Foreground and background pixels localization

The edges found in the last stage are supposed to be the boundaries between the foreground and background, therefore the pixels around the edges belong to the foreground and background separately while some have amphibolous attributes. The purpose of this stage is to identify the pixels to represent the foreground and background. For an edge pixel $I(i,j)$, the 8 neighbor pixels around the edge pixel can be shown as below in table 5.3.

Theoretically, the pixels along the direction of the edge separately belong to the foreground and background. Here we propose a novel method to identify the background and foreground around an edge:

After edge detection and thresholding, we can find edge pixels $I(i,j)$ where the $M(i,j)$ are bigger than the threshold T_{edge}. Here, another 2 corner detection approximation masks shown in table 5.4 5.5 are used. Thereby we can get the gradient magnitude of neighbor pixels around the edge pixels along 4 directions.

We can find the maximum gradient magnitude in the four directions around the edge $I(i,j)$.

Table 5.4: Corner kernel 1

-1	-1	0
-1	0	1
0	1	1

Table 5.5: Corner kernel 2

0	-1	-1
1	0	-1
1	1	0

We will then select the minimum value in this edge or corner with lower values as the foreground and the maximum value in the edge or corner with higher values as background. In this way we can sample a foreground and a background pixel around every edge in an image. For the image in the Fig. 4.4, through such a process, we can get the foreground and background pixels as shown in Fig. 5.3 and Fig. 5.4.

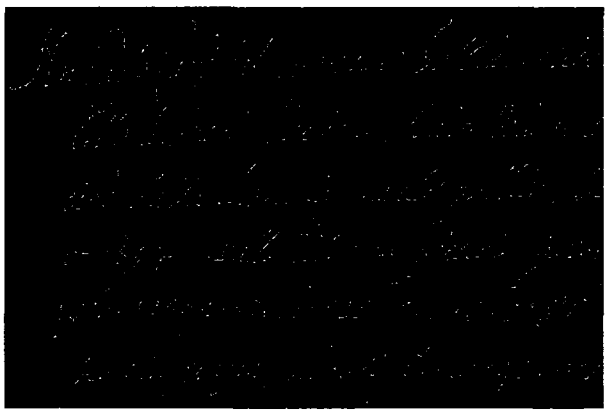

Figure 5.3: The foreground pixels extracted from the Fig. 4.4

Through our observation we can find, no matter how complex the background is, only the pixels around the boundary of foreground, which can dramatically improve the performance of the histogram based thresholding.

Figure 5.4: The background pixels extracted from the Fig. 4.4

5.2.3 Step 3: Analysis of histogram of selected pixels

As mentioned earlier, the major drawback of the histogram based thresholding is that the histograms of the images are required to be bimodal; Otherwise the result will be degraded or it is impossible to obtain a proper threshold. Considering the variance of the real world images, such a requirement heavily constrains the implementation of these methods, though they are still the mainstream techniques of binarization for document analysis. For example the histogram of the Fig. 4.4 is shown as Fig. 4.5. It is impossible to extract the characters from the image through conventional histogram based methods. However from the histograms of the selected pixels from the image as shown in the Fig. 5.5, it is easy for us to calculate the optimal threshold to separate the objects from the background. Because most of the unimportant pixels in the background are skipped, our simulation results prove that in this way we can always get the bimodal histograms from any kinds of noisy images, even though the background is very complex, it has little effect to the extraction of the foregrounds. In this way the histogram of the processed image is enhanced. Because the major inhomogeneities in an image are the relationship between the foreground and background, the dominant edge pixels are the boundary pixels from foreground.

The principle of obtaining optimal threshold is to minimize the errors of misrecognizing the foreground into background and vice versa. After we count the two histograms of the background and foreground as H_{fore} and H_{back}, for every threshold value varying from 0 to 255 there are foreground error rate E_{fore} and background error rate E_{back}, which can be calculated as below:

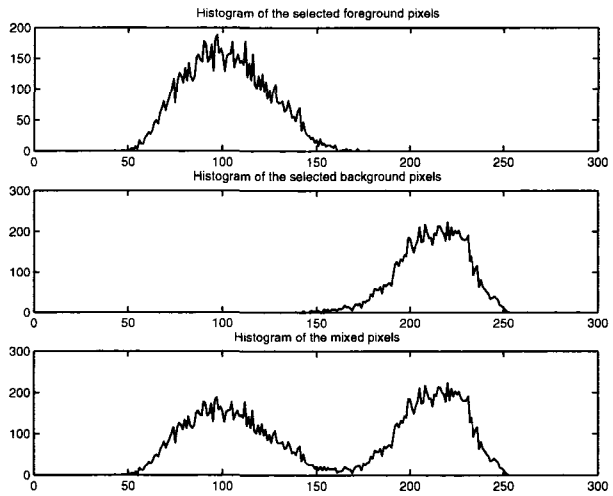

Figure 5.5: The histogram of the selected pixels in the Fig. 4.4

$$E_{back}(j) = \sum_{i=0}^{j} H_{back}(i) \tag{5.5}$$

$$E_{fore}(j) = \sum_{i=255}^{j} H_{fore}(i) \tag{5.6}$$

Then the general error E_{total} can be calculate as:

$$E_{total}(i) = E_{fore}(i) + E_{back}(i) \tag{5.7}$$

The errors can be illustrated as in Fig 5.6. Obviously the optimal threshold is the value which corresponds to the minimum errors.

Since these noises or inhomogeneous backgrounds can be skipped at the edge detection stage, we can explicitly separate the pixels in the foreground from the background without the interference of these pixels with this method. Even though in some cases some edges in the fast changing background will be misrecognized as the boundaries of the foreground at the edge detection stage, the number of edges from the foreground are still big enough to suppress the negative effect of the fake edges. The binarization result for the Fig. 4.4 is shown in Fig. 5.7.

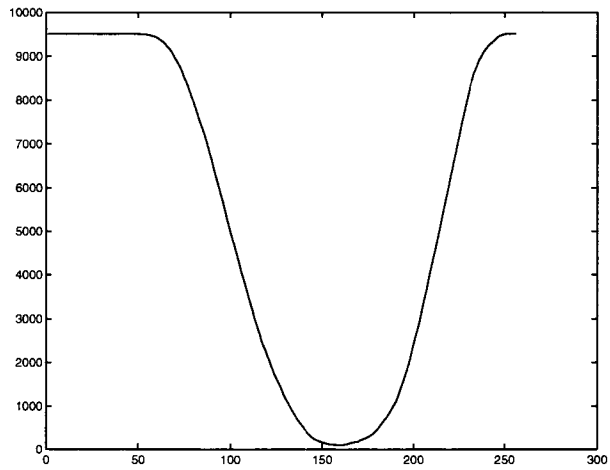

Figure 5.6: The number of errors corresponding to different thresholds

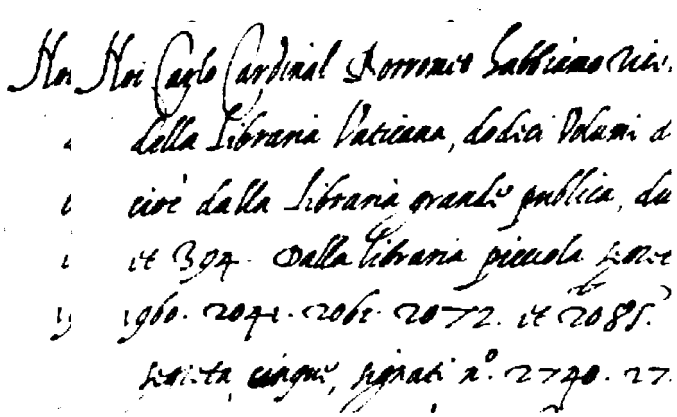

Figure 5.7: The binary document image extracted with proposed global threshold from the image shown in the Fig. 4.4

5.2.4 Local thresholding

Test results show this method succeeds in most of the images, some of which even have low contrast and inhomogeneous background, which are hard for most of the other histogram based methods. However, for the global histogram based methods there is an inherent drawback that for the image with nonuniform illumination such as image shown in Fig. 5.8, one single threshold cannot binarize the whole image properly. The binarization result is shown in Fig. 5.9. Obviously, because of the deficiency of the global method, it is not good enough to handle such problems.

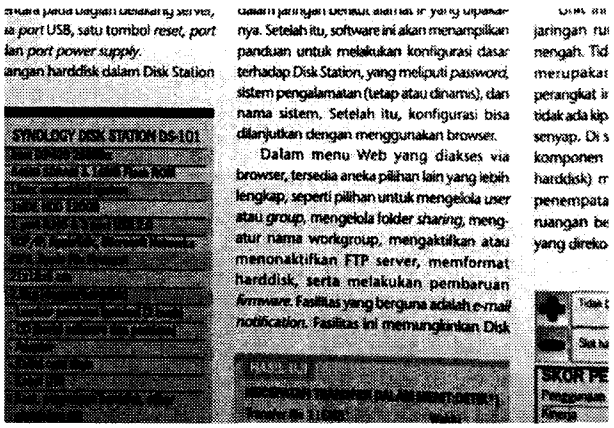

Figure 5.8: An original document image under bad illuminating condition

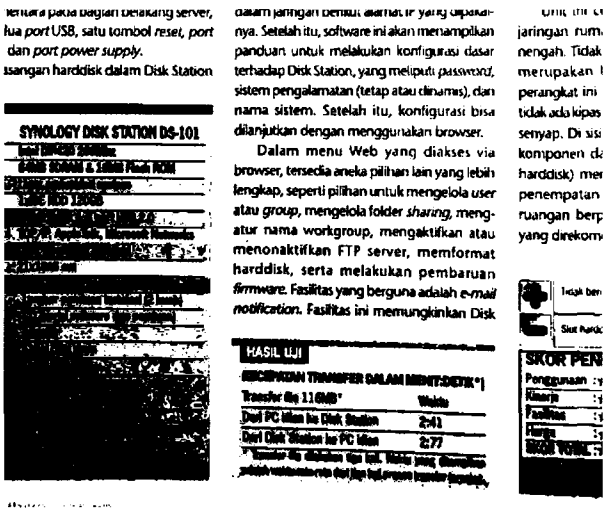

Figure 5.9: The binary document image extracted with proposed global threshold from the image shown in the Fig. 5.8

To improve the performance the local histogram analysis is implemented here. First, an image will be divided into different non-overlapped zones. The threshold is obtained for every zone. Since, for every zone, only the vector of histogram which has 256 elements is to be processed. The extra processing time is short enough for most of applications. Here we set the number of zones as eight by eight and the result is shown in the Fig. 5.10. It should be noted that the division of the images to 8×8 can not be generalized as different applications may require different number of subimages. With the application we had in hand the best result was obtained by divided into 8×8 subimages.

Thereby, the nonuniform illumination problem can be easily overcome. If the illumination is uniform, there are little differences between the global threshold and the local thresholds. In the following sections, the implemented edge based algorithm means the local thresholding method.

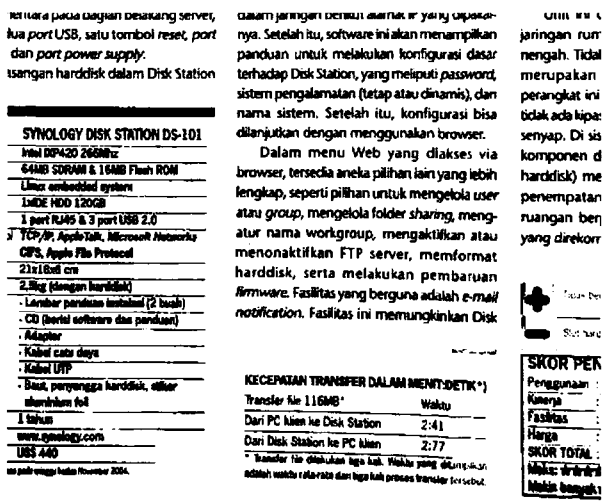

Figure 5.10: The binary document image extracted with proposed local threshold from the image shown in the Fig. 5.8

5.3 Simulation Results and conclusion

We have tested Otsu global thresholding[107], local adaptive thresholding[108], Kittler thresholding algorithms[110] and our proposed edge based thresholding for a number of document images with poor quality caused by bad illumination, non-stationary backgrounds, and signal dependent noises. Our simulation results show that only the edge based segmentation method can provide the satisfactory results in all of the images, while others fail in some of the images. To test the feasibility of proposed algorithms in different noisy images, we would like to demonstrate the binarization results of the three reference algorithms and edge based algorithms.

5.3.1 Test results

To test the feasibility of the proposed algorithm in various applications, the three noisy images shown in figure Fig. 5.1, Fig. 5.8, Fig. 5.11 are considered, which are also used in last chapter.

The histograms of the selected pixels in the three images in Fig. 5.1, Fig. 5.8, Fig. 5.11, can

Figure 5.11: An original document image with an inhomogeneous background

be found in Fig. 5.5, Fig. 5.12, Fig. 5.13 respectively. All of the histograms of the selected pixels in an image from the images are bimodal. The enhanced histograms can be utilized to get the proper threshold. These histograms are demonstrated here to show the bimodality of the selected pixels in tested images. Actually, all of the histograms used in the binarization are the histogram of independant zones in an image as described in 5.2.4, instead of the whole image.

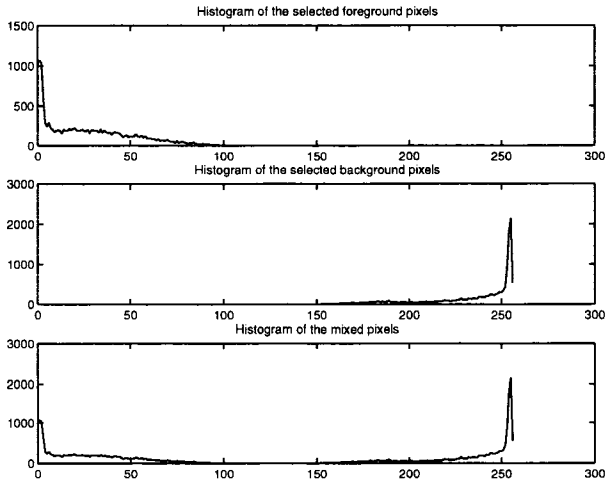

Figure 5.12: The histogram of the selected pixels of Fig. 5.8

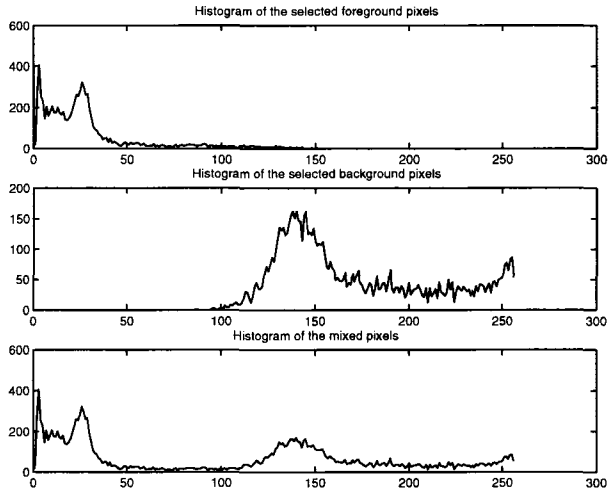

Figure 5.13: The histogram of the selected pixels of Fig. 5.11

5.3.2 Comparison of the binarized images

For the image of an ancient document as shown in Fig. 5.1, from Fig. 5.7 we can tell that the binary image from edge based binarization algorithm is clear and robust. Except the Kittler and Otsu algorithms completely fail, the binary image obtained with Local thrsholding algorithm as shown in Fig. 5.14 is not acceptable to any existing OCR engine.

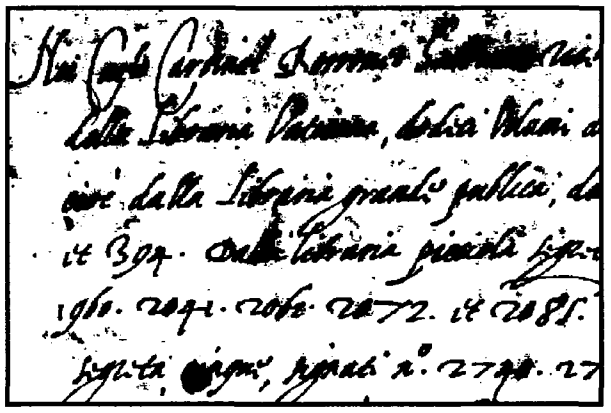

Figure 5.14: Binary document image extracted with the local thresholding algorithm from the original image of Fig. 5.1

The binarization results of the image in Fig. 5.8 which has non-uniform illumination is shown

in Fig. 5.15, Fig. 5.16, Fig. 5.17 and Fig. 5.10. We can find that the binary image from edge based algorithm is more clear than the others.

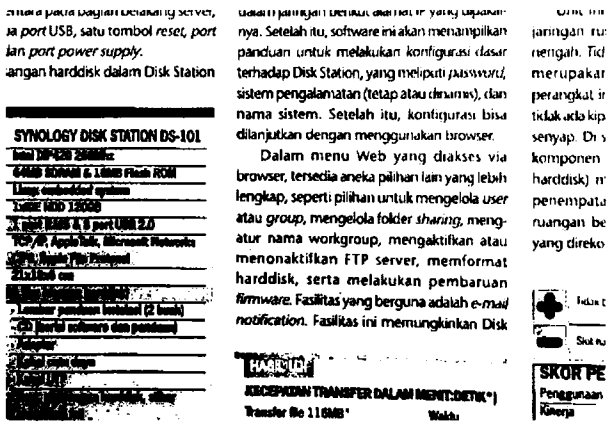

Figure 5.15: Binary document image extracted with the Otsu algorithm from the original image of Fig. 5.8

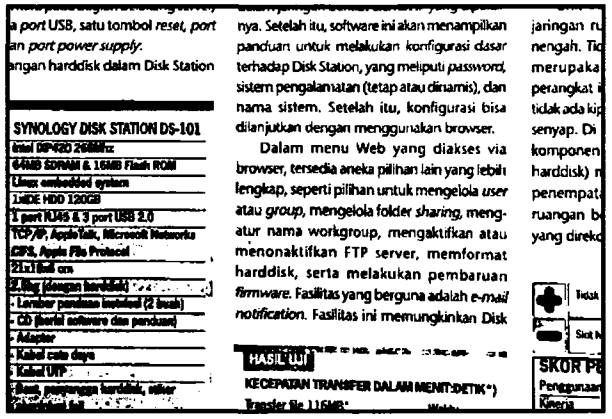

Figure 5.16: Binary document image extracted with the Local thresholding algorithm from the original image of Fig. 5.8

The binarization results of the image in Fig. 5.11, which has complex background is shown in Fig. 5.19, Fig. 5.20, Fig. 5.21, Fig. 5.22. We can find except the Kittler algorithm has identical performance as the edge based binarization, and the other algorithms partially fail.

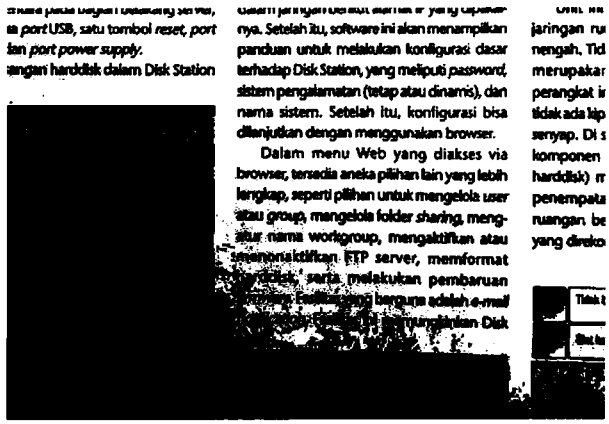

Figure 5.17: Binary document image extracted with the Kittler algorithm from the original image of Fig. 5.8

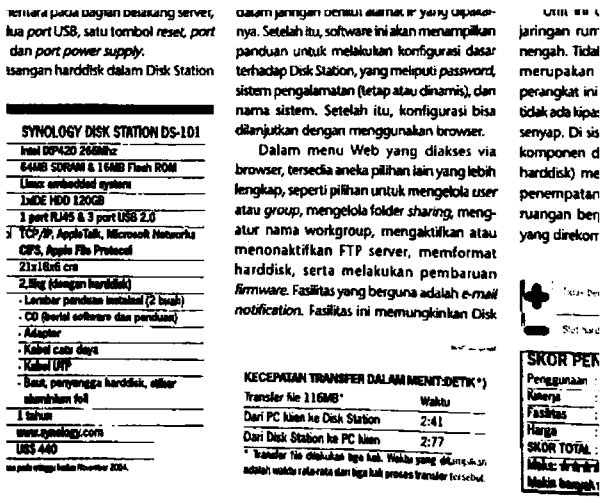

Figure 5.18: Binary document image extracted with the edge based thresholding algorithm from the original image of Fig. 5.8

5.3.3 Quantitative Study

To further prove the efficiency of the proposed binarization method, we carry out the same experiment as last chapter. 42 images are randomly selected from the database of ICDAR[171]. After binarization the binary images are sent to the commercial OCR software Readiris 10.04 Professional, the recognition results can be regarded as a reliable criterion to evaluate the performances of differ-

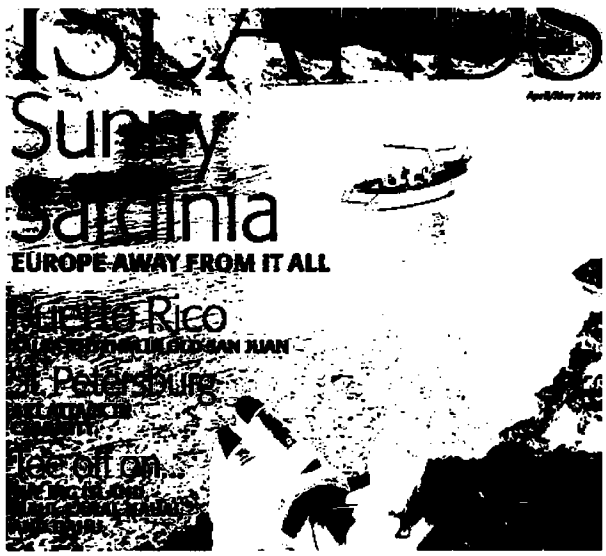

Figure 5.19: Binary document image extracted with the Otsu algorithm from the original image of Fig. 5.11

Figure 5.20: Binary document image extracted with Local thresholding from the original image of Fig. 5.11

ent binarization methods. There are a total of 1078 characters in the tested images. The OCR test results are reported in table 5.6. One can find the recognition rate based on the binarized image

Figure 5.21: Binary document image extracted with the Kittler algorithm from the original image of Fig. 5.11

Figure 5.22: Binary document image extracted with the Edge based thrsholding algorithm from the original image of Fig. 5.11

from proposed method is much higher than the reference ones.

Table 5.6: OCR results from different binarized images

Method	Edge based binarization	Kittle[110]	Local[108]	Otsu[107]
OCR recognition rate	67.3%	25%	48%	53%

5.3.4 Conclusion

In this chapter, we have presented a novel thresholding method based on edge information to binarize seriously degraded and very poor quality gray-scale document image. Our method can threshold gray-scale document images with complex signal-dependent noise, variable background intensity caused by nonuniform illumination, shadow, smear or smudge and very low contrast without obvious loss of useful information. From the tests of the various styles of noisy images, we can say that the proposed algorithm provides robust results for all of the tested cases. The major reason for the robustness of the proposed algorithm is that our method extracts the threshold based on the pixels around the edges. In this way the tremendous amount of useless information contained in the background can be ignored. From the above simulation it can be found that the histograms of the selected pixels are real bimodal. Its computational efficiency is much higher than other algorithms such as connectivity-based thresholding or local adaptive methods [153][162] [163]. The quantative test shows its recognition rate is 67.3%, which is 14.3% higher than the best reference binarization methods. In comparison with other histogram based algorithms, only one more edge detection process is added, which is fast enough for most of the real world applications.

Chapter 6

1-D Self-Adaptive HMM and its application to OCR

6.1 Conventional HMM and its drawback

Chapter 3 has provided the detailed introduction to HMM and its applications in different fields. Some variants of standard HMMs are conceived as extensions of the internal structure of the model. However, the data interface towards the external world essentially remains the same and the basic data objects being processed are still single sequences. One can notice that there are some drawbacks in the theory of the conventional HMMs.

First of all, the hidden Markov model is a causal system. The probability of state at time $(t+1)$ is derived from state t and affected by the present observation $(t+1)$. For classifiers based on Hidden Markov Models, all of the observation elements are parallelly inputted into the model. The hypothesis that the hidden state is only derived from the former one is untrue. If state t could be determined by the states and observations at t-1 and t+1, the model should be more accurate than the unidirectional Markov chain.

Secondly, HMM is a one path chain, which means that with the presence of noise the whole propagation path of the hidden states after the noise will be changed dramatically. For example, there are a sequence of observations, V_i, as shown in figure 6.1, and according to equations 3.22 3.23 3.24, the optimal path $S_1S_2S_2S_3S_1S_4$ can be easily obtained. When there is noise interferring

with the signal and the obervation V_3 is replaced by noise V_3' as shown in figure 6.2, the calculated probabilitis of states in location $t = 3$ may be completely different from the correct one. Therefore, the following induction from $t+1$ to T will easily detour to a wrong path. For example, in this case the new optimal path will be $S_1S_2S_3S_1S_4S_2$ as illustrated in figure 6.2.

When HMM is utilized as a classifier, to simplify the calculation, in many cases the probability of $P(O|\lambda)$ is substituted by the $P(O, Q*|\lambda)$, where $Q*$ is the optimal path. The wrong estimation of optimal path will yield a wrong conclusion. In the normal evalution method as shown in equations 3.11 3.12 3.13, the memberships of the hidden states should be estimated from the former time slot's hidden states and present observation. With the presence of noise, the performance of the HMM based classifiers will be degraded.

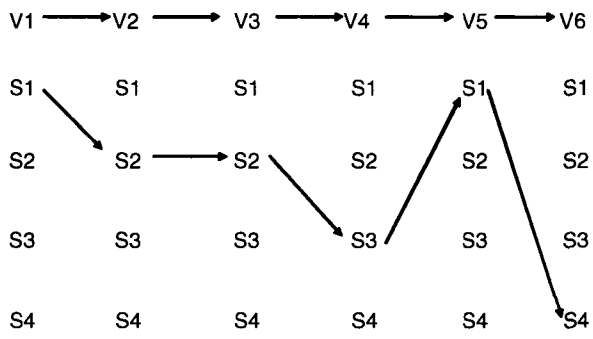

Figure 6.1: Optimal path search in noise free signal

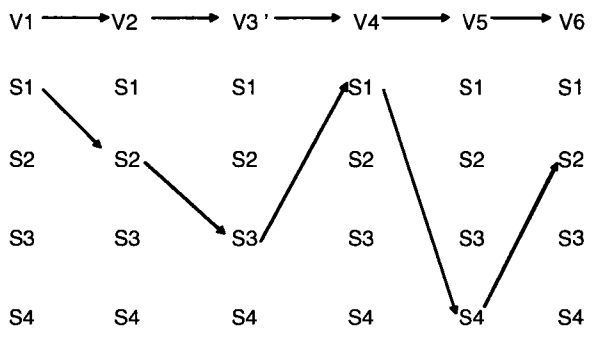

Figure 6.2: Optimal path search in noise free signal

6.2 The Proposed Self-Adaptive Hidden Markov Model

6.2.1 Elements of the proposed model

In this section we will propose a new Self-Adaptive Hidden Markov Model. Like the conventional HMM, in this model the following elements are introduced: N, number of hidden states and M, number of observation symbols in the alphabet.

Hypotheses about the relationships of the elements are proposed and the model is characterized by the following:

1) States combination probabilities L_{ij}.

2) Probability of distribution of observations to every state $B = b_j(k)$.

3) The distribution probability of states in every time slot of the observation sequence $D_t(j)$.

4) Probability of the state j occurring in the model C_j.

L_{ij} is the probability of the neighbor states combination. The major difference between the proposed method to the conventional HMM is that instead of considering the probability of the transition from one time slot to the next, a bi-directional combination of the neighbour states will be checked. In this way the proposed model is not a causal system anymore and the drawbacks mentioned in last section will be eliminated. L_{ij} can be expressed as below:

$$L_{ij} = P(q_t = i, q_{t+1} = j),\ 1 \leq i, j \leq N \tag{6.1}$$

It should be mentioned that we have assumed that this combination is a non-skip theme, hence:

$$L_{ij} = 0 \text{ when } j < i \text{ or } j > i + 1 \tag{6.2}$$

$B = b_j(k)$ is the same as conventional HMM:

$$b_j(k) = P(O_t = V_k | q_t = j),\ 1 \leq j \leq N,\ 1 \leq k \leq M \tag{6.3}$$

$D_t(j)$ is the distribution probability of states in every time slot of the observation sequence.

$$D_t(j) = P(q_t = j),\ 1 \leq t \leq T, 1 \leq j \leq N \tag{6.4}$$

In the conventional HMM there is a corresponding parameter π, which is only available to the hidden states in location 1, the memberships of the states at other locations can only be deduced by the equation 3.11 and 3.12, usually from left to right. In the new model every location has an inherent states distribution probability. Consequently, the memberships of the states in every location will

depend less on the neighbors. Therefore, noise will have less effect on the estimation of the states at other time slots.

The probability of the state, j, occurring in the model C_j is a newly introduced concept in this model, its usage will be demonstrated later.

Instead of beginning to estimate the hidden states from the two ends of the Markov chain, in this model the estimation of states will be started from every time slot in the sequence simultaneously. Hence the influence of noise in the sequence will be minimized. The detailed procedure of the model will be addressed in the following section.

6.2.2 Evaluation stage

The evaluation process of the proposed model consists of the following steps:

1) Single node states estimation.

The estimation of the states in every time slot will only be started from every single obervation. The memberships are decided by two issues: the location of the time slot and the observation at this time slot. The first one can be looked up from the $D_t(k)$, while the second one depends on observation symbol probability distribution $b_j(k)$. The general probability can be calculated as below:

$$P_t(O_t|s_i) = b_j(O_t)D_t(j),\ 1 \leq t \leq T,\ 1 \leq j \leq N,\ 1 \leq O_t \leq M \tag{6.5}$$

2) Coupled nodes states estimation.

After independantly obtaining the states' memberships of every slot in the sequence, the influence from neighbour slots are considered here to further estimate the memberships at every time slot. First of all, we have to prove the linearity of this model:

Assuming that we have a data set of states, after a statistic audit, besides probabilities of the neighbor states combination L_{ij}, we can obtain two extra conventional forward and backward transition parameters where:

$$a_{ij} = P(q_{t+1} = j|q_t = i),\ 1 \leq i,j \leq N \tag{6.6}$$

$$b_{ji} = P(q_t = i|q_{t+1} = j),\ 1 \leq i,j \leq N \tag{6.7}$$

The probability of every state i existing in the model c_i can also be obtained, where $1 \leq i \leq N$

Obviously one can get:

$$L_{ij} = c_i a_{ij} = c_j b_{ji} \tag{6.8}$$

Hence L_{ij} is conditionally independent. The probability of states i and j occuring in slot t and t+1 is L_{ij}; in general, it should be regarded that ${c_t^{(0)}}_i = c_i$ and ${c_{t+1}^{(0)}}_j = c_j$. The superscript 0 here is the iteration time and the 0 means it has not yet been processed. If the probability of state i occurring in location t is ${c_t^{(1)}}_i$, and ${c_{t+1}^{(0)}}_j = c_j$:

$$\widehat{{L_t^{(1)}}_{ij}} = {c_t^{(1)}}_i a_{ij} = \frac{{c_t^{(1)}}_i}{c_i} c_i a_{ij} = \frac{{c_t^{(1)}}_i}{c_i} L_{ij} \tag{6.9}$$

In a similar way, if the probability of state j in time slot $t+1$ is changed to ${c_{t+1}^{(1)}}_j$, and ${c_t^{(0)}}_i = c_i$, then:

$$\widetilde{{L_t^{(1)}}_{ij}} = {c_{t+1}^{(1)}}_j * b_{ji} = \frac{{c_{t+1}^{(1)}}_j}{c_j} c_j b_{ji} = \frac{{c_{t+1}^{(1)}}_j}{c_j} L_{ij} \tag{6.10}$$

If the ${c_t^{(0)}}_i$ is changed from c_i to ${c_t^{(1)}}_i$, and ${c_{t+1}^{(0)}}_j$ is changed from c_j to ${c_{t+1}^{(1)}}_j$ simultaneously, we can get:

$${L_t^{(1)}}_{ij} = \frac{{c_t^{(1)}}_i}{c_i} L_{ij} \frac{{c_{(t+1)}^{(1)}}_j}{c_j} \quad 1 \le t \le T-1 \tag{6.11}$$

Thereby, the linearity of the probability of combined states is proven. From equation 6.11 the probability of state i and j happening in slot t and $t+1$ will be:

$${L_t^{(1)}}_{ij} = \frac{P_t(O_t|s_i)}{c_i} L_{ij} \frac{P_{(t+1)}(O_{(t+1)}|s_j)}{c_j} \quad 1 \le t \le T-1 \tag{6.12}$$

The membership can be normalized as:

$${L_t^{(2)}}_{ij} = \frac{{L_t^{(1)}}_{ij}}{\sum_{i=1}^{N} \sum_{j=1}^{N} {L_t^{(1)}}_{ij}} \quad 1 \le t \le T-1,\ 1 \le i,j \le N \tag{6.13}$$

Based on the link information formulated as above, we can get the new membership of states in t and $t+1$ as:

$${P_t^{(2)}}_i = \sum_{j=1}^{N} {L_t^{(2)}}_{ij} \quad 1 \le t \le T-1,\ 1 \le i,j \le N \tag{6.14}$$

$${P_{(t+1)}^{(2)}}_j = \sum_{i=1}^{N} {L_t^{(2)}}_{ij} \quad 1 \le t \le T-1,\ 1 \le i,j \le N \tag{6.15}$$

The superscript (2) means the latest updated information.

3) Propagation of states information:

In a given chain, every node with the exception of the end nodes are connected to two neighbours. There are two sets of memberships of states derived from the two neighbours, while the two sets of memberships are not guaranteed to be identical in most of the cases. How to solve

this conflict is critical for the model. Several solutions were proposed to solve such noncausal problems, such as Recurrent Neural Network(RNN)[86] and Bidirectional Input Output Hidden Markov Model(BIOHMM)[87].

Here we propose a new asynchronous method to solve the noncausal problem. The process consists of following steps:

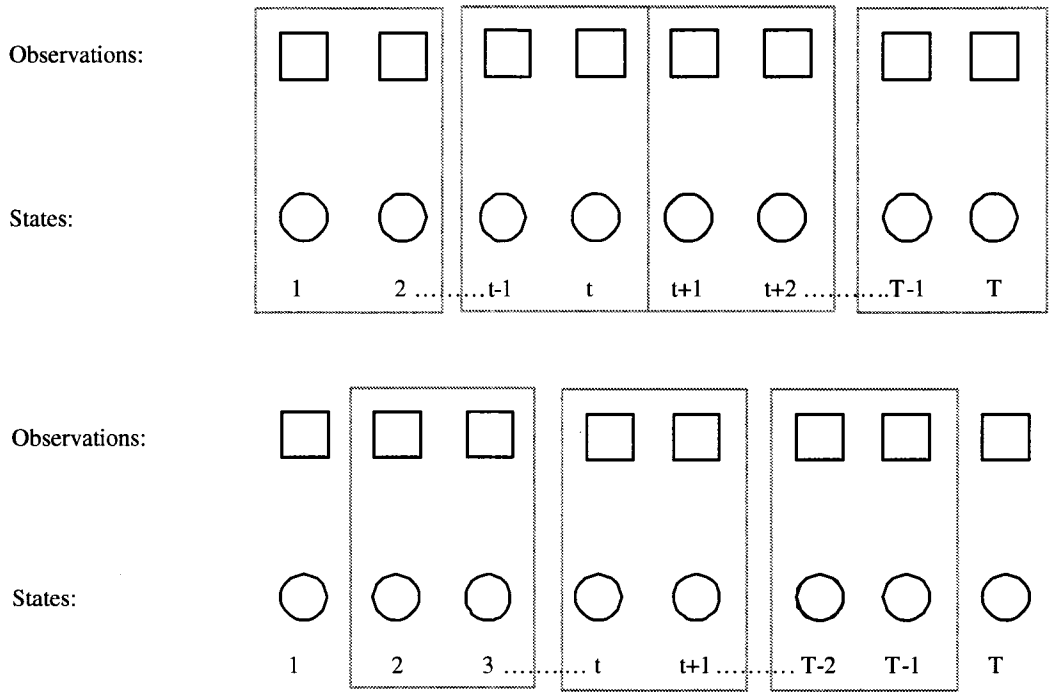

Figure 6.3: Asynchronous states estimation method

Step 1: As shown at the top of figure 6.3, the combined probabilities of states in location $\{1,2\},\{3,4\},\{T-1,T\}$ (Without loss of generality, T is assumed to be even) are calculated with the method shown in the last section. Then, we can get the combined probability: $L_t(ij)$, where t is odd. After normalization, we can get the memberships of states in every node in the sequence.

Step 2: As shown at the bottom of figure 6.3, the combined probabilities of states in location $\{2,3\},\{4,5\},\{T-2,T-1\}$ with updated memberships are calculated in the same way as the last step. Then we can get the combined probabilies:$L_t(ij)$, where t is even, and the latest memberships in every node except the two ends.

Step 3: Repeat above process. After each iteration, the information about states contained in a time slot can be transmitted into neighbour slots in two directions. The optimum number of iteration

will vary according to different applications. In the case of handwritten character recognition our simulation results show that we can get the best result when the process is repeated at least one fifth of the length of the sequence. This will be demonstrated in section 6.3.

4) Special cases:

For ideal clean signal sequence, it is easy for the membership information to be transmitted along the path. However, in the presence of noise one gets:

$$\sum_{i=1}^{N}\sum_{j=1}^{N} {L_t^{(k)}}_{ij} = 0,\ 1 \le i,j \le N \tag{6.16}$$

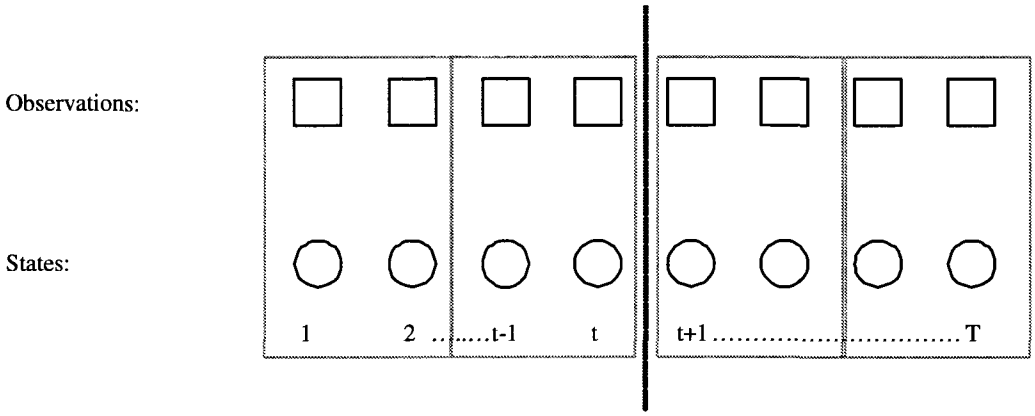

Figure 6.4: Asynchronous states estimation method

This indicates that the hidden states at the two ends of this link are not continuous. Then, according to equation 6.14 and 6.15, all of the memberships of hidden states at node t and $t+1$ will be 0, which is obviously wrong. To solve this problem, we ignore the new memberships which are 0 and will not update them. In the meantime, the original sequence is split into two sequences as shown in figure 6.4. Therefore, the influence of degradation will be removed; this was made possible because state information is not transmitted further. When there are more than one severe degradations in the sequence, the sequence will be divided into more than two segments while every segment is considered independently.

5)Objective functions:

After the above iterations, we can obtain the estimate of the memberships of states in every node. The probability of every node in the sequence generating the corresponding observation can

be calculated as below:

$$P_t^{(n)} = \sum_{i=1}^{N} P_t^{(n)}(i) b_i(O_t) \ 1 \leq t \leq T \tag{6.17}$$

n is the iteration time, t is the time slot, $P_t^{(n)}(i)$ is the probability of state i occurring in time slot t after n interation and $b_i(O_t)$ is the probability of observation O_t occurring in state i.

Then, the general probability of every node in each mode in the given sequence is:

$$P_{node} = \prod_{t=1}^{T} P_t^{(n)} \tag{6.18}$$

Besides the probability observed in every node, the chance of the combination of every two neighbour hidden states can be calculated between node t and $t+1$ as below:

$$L_t^{(n)}{}_{ij} = \frac{P_t^{(n)}{}_i}{c_i} L_{ij} \frac{P_{(t+1)}^{(n)}{}_j}{c_j} \ 1 \leq t \leq T-1 \tag{6.19}$$

We can therefore get the general probability of link between node t and $t+1$ as:

$$L_t = \sum_{i=1}^{N} \sum_{j=1}^{N} L_t^{(n)}{}_{ij} \ 1 \leq t \leq T-1, \ 1 \leq i,j \leq N \tag{6.20}$$

Then, all of the links(Combination of neighbour states) existing in the sequence will be:

$$P_{link} = \prod_{t=1}^{T-1} L_t \tag{6.21}$$

The combination of above two equations 6.18 and 6.21 will yield the final probability as:

$$P_{final} = P_{link} P_{node} \tag{6.22}$$

It is possible that probabilities of some nodes or links become 0. Therefore, after multiplication the general result will be 0. In such a case, we will substitue the 0 with a small positive value to avoid a rigid result.

When the model is implemented in the classification phase, the self-adaptive model based recognizer aims to determine the model that is most probably the one that produced the provided test signal. i.e.

$$\lambda = \arg \max_{All\ of\ the\ \lambda} P(\lambda|O) \tag{6.23}$$

This is Maximum A Postriori Probability(MAP) problem, where Bayes rule is utilized to reformulate the recognition problem as:

$$\lambda = \arg \max_{All\ of\ the\ \lambda} \frac{P(O|\lambda)P(\lambda)}{P(O)} \tag{6.24}$$

$P(\lambda)$ is usually considered equal for all models, i.e. all models are equally probable. Thus Equation 6.24 can be simplified as:

$$\lambda = \arg \max_{All\ of\ the\ \lambda} P(O|\lambda) \tag{6.25}$$

In the evaluation stage, signal sequence is sent to every model in the system to calculate the probabilily as shown above. The MAP is the final result of the classification. The next important problem is to estimate the parameters in the model during the training stage.

6.2.3 Training stage

Conventional HMM training methods

The training stage is the most critical step for the performance of a classifier. In this process the statistical parameters of the model will be optimized to fit a set of observed training data. For conventional HMM, many successful heuristic algorithms such as the Baum-Welch algorithms [94] and the gradient methods[88] are developed for the optimization of the model parameters. Baum-Welch algorithm iteratively provides parameter estimates that maximize the actual (Maximum Likelihood Estimation)MLE criterion, i.e. $P(O|\lambda)$; those estimates are based on partial path probabilities computed by forward and backward recurrences and converge at least to a local maximum. Even the maximum likelihood can not yield analytically optimal results. Another straight-forward method is the decision-directed estimation algorithm[100], where suboptimal results are obtained with less computational cost. Besides the above methods the Genetic Algorithm [89] is an alternative method to further improve the training process. Until now the Baum-Welch algorithm is still the dominant method because of its guanranteed convergence. In this method, an auxilary parameter ξ is introduced, where

$$\xi_t(i,j) = \frac{\lambda_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{P(O|\lambda)} = \frac{\lambda_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)} \tag{6.26}$$

Another parameter $\gamma_t(i)$ is added here to describe the probability of being in state S_i at time t, given the observation sequence and the model; hence, we can relate $\gamma_t(i)$ to $\xi_t(i,j)$ by summing over j, giving

$$\gamma_t(i) = \sum_{j=1}^{N}\xi_t(i,j) \tag{6.27}$$

When we sum $\gamma_t(i)$ over the time index t, we get a quantity which can be interpreted as the expected number of transitions made from state S_i. At the same time, summation of $\xi_t(i,j)$ over t can be

interpreted as the expected number of transitions from state S_i to state S_j. That is

$$\sum_{t=1}^{T-1} \gamma_t(i) = expected\ number\ of\ transitions\ from\ S_i \tag{6.28}$$

$$\sum_{t=1}^{T-1} \xi_t(i,j) = expected\ number\ of\ transitions\ from\ S_i\ to\ S_j \tag{6.29}$$

With the above formulas we can give a method for re-estimation of the parameters of an HMM. Hence the estimated π, A and B are

$$\overline{\pi} = expected\ frequency\ in\ state\ S_i\ at\ time\ (t=1) = \gamma_1(i) \tag{6.30}$$

$$\overline{a_{ij}} = \frac{expected\ number\ of\ transitions\ from\ S_i\ to\ S_j}{expected\ number\ of\ transitions\ from\ S_i} = \frac{\sum_{t=1}^{T-1} \lambda_t(i,j)}{\sum_{t=1}^{T-1} \xi_t(i)} \tag{6.31}$$

$$\overline{b_j(k)} = \frac{expected\ number\ of\ times\ in\ state\ j\ with\ symbol\ v_k}{expected\ number\ of\ times\ in\ state\ j} \tag{6.32}$$

Based on the above procedure, if we iteratively use $\overline{\lambda}$ as estimated parameters and repeat the reestimation calculation, we can then improve the probability of O being observed from the model until the time of iteration is over a specific number or the likelihood calculated in equation 6.25 is over a threshold.

Since the proposed method is non-linear and non-causal, it is difficult to train the model with conventional methods. Here, we will introduce two feasible methods. Simulation results show the sufficiency of the proposed training methods.

Estimation of proposed method parameter using conventional HMM

The proposed model has a similar architecture to the conventional HMM. The common cores of the two models are the structure of two layers: observations and hidden states. All of the parameters in both models indicate the relationship of the two layers. The only difference is that the relationship of the hidden states in the conventional HMM is causal, which means the states are derived from upstream or downstream states, while in the proposed method both of them will be considered. Here we will introduce a feasible training method. To train a proposed SAHMM, we establish a HMM with the same number of states and observations. After the training data are inputted into the HMM, it's easy to obtain the optimal hidden states in the database.

Assuming we have a self-adaptive model with N states and M observations, we can set up a HMM with the same number of states and observations. Optimal paths searching is a decoding problem that is well studied for the HMM. In this step the Viterbi algorithm[79] is the most used method.

In this method, to find the single best state sequence, $Q = q_1, q_2, ...q_T$ for the given observation sequence $O = O_1, O_2....O_T$, we need to define the quantity $\delta_t(i)$ which is the best score at t along a single path, considering the observations from the first element to the element t. We can also find that

$$\delta_{t+1}(j) = max[\delta_t(i)a_{ij}]b_j(O_{t+1}) \ 1 \leq i \leq N \tag{6.33}$$

For a given database, we can find the unique optimal path for every sequence under supervision. For example, if we have K data in the database and the length of the sequence is T with M observations. In this case, the sequences have constant length. After the decoding, we can find the K optimal paths for all of the sequences. After enumerating the K observation sequences and K hidden states sequences we can obtain the C_j, L_{ij}, $B = b_j(k)$ and $D_t(j)$.

The total number of state i occuring in the data base should be the $NumberStates_i$, in this case

$$\sum_{i=1}^{N} NumberStates_i = K \times T \tag{6.34}$$

then C_j can be obtained

$$C_j = \frac{NumberStates_i}{\sum_{i=1}^{N} NumberStates_i} = \frac{NumberStates_i}{K \times T} \tag{6.35}$$

The total number of combinations of neighbour state i and j occuring in the database should be $NumberCombination_{ij}$ and can be obtained

$$L_{ij} = \frac{NumberCombination_{ij}}{\sum_{i=1}^{N}\sum_{j=1}^{N} NumberCombination_{ij}} \tag{6.36}$$

where $L_{ij} = 0$ when $j < i$ or $j > i + 1$

With the same method we can obtain the parameter $D_t(j)$:

For every time slot in the K sequences, one more variable $Num_t(i)$ is defined, which is the number of times the state i occurring in the time slot t;

$$D_t(j) = \frac{Num_t(i)}{\sum_{i=1}^{N} Num_t(i)} \tag{6.37}$$

In the same way we can find the $B = b_j(k)$:

$$b_j(k) = \frac{Number\ of\ observation\ k\ occurring\ when\ the\ state\ is\ j}{number\ of\ states\ j} \tag{6.38}$$

Through this method, we can estimate all of the parameters in the Self-Adaptive HMM with the aid of the conventional HMM.

Iterative training method

Another available training method utilizes the iterative strategy. First, we assign initial values to the parameters in the model. Assuming we have the training set with K sequences and the length of each sequence is T, according to the method mentioned in the last section we can obtain the probability of the neighbor combination of the hidden states in every time slot as $L_{kt}^n(ij)$, where k is the index of the sequence in the database and t is the time slot, n is the iteration time, while i and j are the states. We can easily get the probability of the hidden states at time t $P_{kt}^n(i)$:

$$P_{kt}^n(i) = \sum_{j=1}^{N} L_{kt}^n(ij) \; 1 \leq i \leq N, \; 1 \leq t \leq T-1 \tag{6.39}$$

when t=T, the $P_{kT}^n(j)$ is:

$$P_{kT}^n(j) = \sum_{i=1}^{N} L_{k(t-1)}^n(ij) \; 1 \leq i \leq N \tag{6.40}$$

After estimating the fuzzy membership of states in every time slot in the database, we can obtain the new parameters in the model, like the method used in the last section. After several iteration, a new set of parameters will be available. Our simulation shows that this method can converge into a stable local maximum. A drawback of this method is that the performance of the model highly depends on the intial values because only a local maximum is reached at the end. Here the decision-directed estimation algorithm is proposed to estimate the initial parameters in the model to improve the performance of the classifier.

Decision Directed Estimation(DDE) method

The decision directed estimation method is a straight-forward way to find the inherent relationship of the states and observations. The details are explained as below:

1) Distribution probability

Since the states distribution $P(t|i)$ (t is the time slot, i is the state) would be exponential $a_i e^{-(\frac{(t-b_i)^2}{c_i^2})}$, where $1 \leq t \leq T$ and the a,b,c is the coefficient of the Gaussian function, b refers to the center of the Gaussian distribution, and we assume the center b is equally distributed in the sequence. Then

$$b_i = \frac{T}{(N-1)}(i-1) \; 1 \leq i \leq N \tag{6.41}$$

After normalizing the states distribution for every time slot, the probability of state j occurring at time slot t is $D_t(j)$:

$$D_t(j) = \frac{P(t|i)}{\sum_{i=1}^{N} P(t|i)} \tag{6.42}$$

Since this is just initial estimation, the distribution parameter of each model for every character in this case is the same.

2) The probability of every state i existing in the model is C_i that can be derived from $D_t(i)$:

$$C_i = \sum_{t=1}^{T} D_t(i) \tag{6.43}$$

3) The initial estimation of the probability of combination of state L_{ij} in a model can be obtained:

Here we assume there are equal chances for the L_{ij} occurring in a model. Since $L_{ij} = 0$ when $j < i$ or $j \geq i+1$, there are 2 possible combinations for every state except when $i = N$, which means there are total 2N-1 different combinations. Thereby initially we can assume the $L_{ij} = \frac{1}{2N-1}$.

4) The observation probability distribution in each of the states, $B = b_j(k)$ can be calculated as below:

From step 1, we can get the membership of the states in every location in every sequence. For example for the sequence w, in time slot t the observation is k, then the chance of this time slot belongs to state j is $P_{wt}(j) = D_t(j)$. Consequently the observation k belonging to state j here is $P_{wt}(k|j) = D_t(j)$. It's easy to accumulate the expectation value E(j) of state j shown in the database as:

$$E(j) = \sum_{w=1}^{K} \sum_{t=1}^{T} P_{wt}(j) \; 1 \leq j \leq N \tag{6.44}$$

And the expectation value of state j when observation is k:

$$E(k|j) = \sum_{w=1}^{K} \sum_{t=1}^{T} P_{wt}(k|j) \; 1 \leq j \leq N \tag{6.45}$$

Then we can get the observation distribution in state j as:

$$O(k|j) = \frac{E(k|j)}{E(j)} \; 1 \leq j \leq N, \; 1 \leq k \leq K \tag{6.46}$$

Until now we have obtained the estimate of the initial values of the parameter in the model. Inputting the initial values into the model to continue the training, we will find that better results can be obtained compared to random initial values.

6.3 Implementation of 1-D Self-Adaptive Model in handwritten character recognition

6.3.1 Dataset

This section presents the simulation results obtained by the proposed methods on the MNIST(Modified NIST) database, which is the widely used benchmark of handwritten digits that contains a training set of 60000 images and a test set of 10000 images. These data are considered by majority of researchers in the field. The original black and white (bilevel) images from NIST were fit in a 20x20 pixel box. After conversion in the MNIST the resulting images contain grey levels as a result of the anti-aliasing technique used by the normalization algorithm. Eventually the images were centered in a 28x28 pixel box with gray pixel values.

6.3.2 Feature extraction

The investigation of feature extraction methods has gained considerable attention because a discriminative feature set is considered the most important factor in achieving high recognition performance. The feature used here is just the pixel values in each column of gray images. The Self Organizing Map (SOM)[92] method is utilized here to quantize the feature vectors. First, we have to reduce the size of the image from 28 by 28 to 20 by 20. The 20 column pixel values from the 60000 characters in the MNIST will generate a code book. With SOM we can divide the feature space into a number of subspaces, where the average value of the vectors in the subspace can be regarded as the center of the subspace. Every column in an image can be represented by the discrete symbol of the subspace. Then every character image will be converted into a sequence of 20 elements.

6.3.3 Simulation results

To carry out a comparative study, we here set the same number of observations and states to HMM as to the proposed model. It is easy to train the HMM to obtain its parameters. It should be mentioned that this HMM is left to right single path, and no state skip scheme is allowed, which means $a_{ij} = 0$ *when* $j \leq \ i$ *and* $i \ \geq \ j + 1$. With the Baum-Welch algorithm, we get the optimal hidden state path in the database. With the method mentioned in the last section, we can obtain the parameters of the proposed model.

Effect of the iteration times in the evaluation phase to the performances of proposed model

In last section we mentioned that the iteration is critical for the estimation of the states' memberships. Our experiment shows too many iterations yield suboptimal results. The empericial results can be shown in figure 6.5. In the tested model the number of states is 15, the number of observation is 400 and this test is performed on the original MNIST dataset without degradation. It was discovered when the iteration time is 4, the optimal result is obtained. Additional iteration may not yield better performance. A large number of tests based on different models with different number of states and observations were carried out and similar curves were obtained. Such iterative method is widely utilized and proven in training of conventional HMM[97]. How to obtain the optimal iteration time in different applications and how to mathematically prove its validity at the SAHMM is yet to be studied.

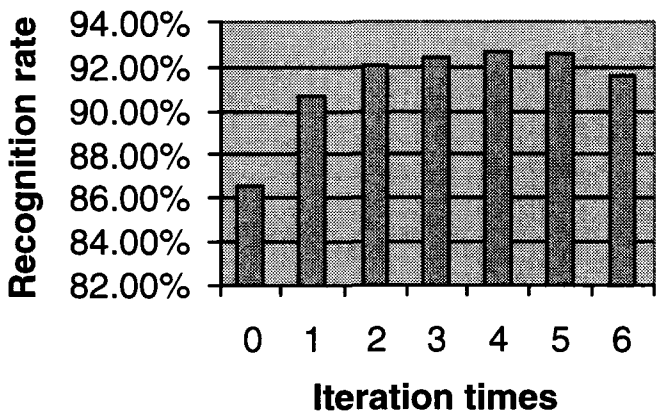

Figure 6.5: Recognition rate with different times of iteration in evaluation stage of the proposed model

Test results with the iterative training

In order to examine the validity of the proposed iterative training algorithm, we first use the direct decision method to obtain the initial values and iteratively train the model until the recognition rate stopped increasing. The number of states is 15 and the number of observation is 625. The learning curve can be checked in figure 6.6. One can observe the performance of Self-Adaptive HMM will be stable after some iterations.

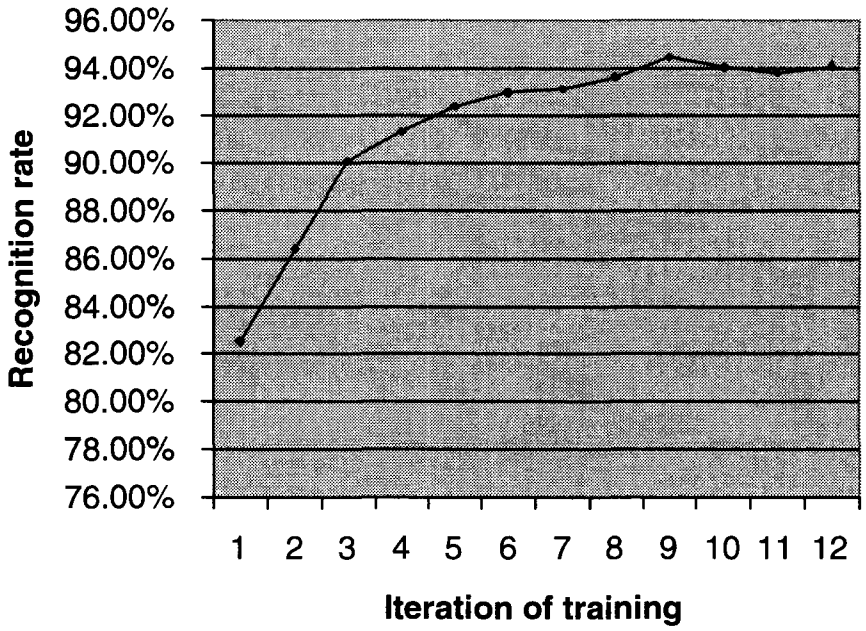

Figure 6.6: Learning curves for iterative training algorithm

Comparison of the performances of proposed model and conventional HMM under degraded environment

In the preceding section we have mentioned that the major goal of the designed model is to try to minimize its vulnerability to noise and other degradations as opposed to conventional HMM. To test the performance of the model in the degraded environment, we arbitrarily replace one or several elements in every sequence with random valid number(s) to imitate the presence of noise. The length of the sequence is 20 in this case. In this experiment we replace up to 3 elements in the feature sequences of the characters to test the performance of the proposed model and reference models.

Table 6.1: Performance of the proposed model when the number of observations is 400

Number of noise	Self-Adaptive HMM	HMM	Difference
0	92.7%	91.8%	0.9%
1	68.8%	62.6%	6.2%
2	54.1%	44.75%	9.35%
3	44.4%	33.67%	10.73%

Table 6.2: Performance of the proposed model when the number of observations is 625

Number of noise	Self-Adaptive HMM	HMM	Difference
0	93.82%	93.5%	0.32%
1	71.6%	67.00%	4.6%
2	55.33%	50.60%	4.73%
3	46.77%	38.80%	7.97%

The proposed model is examined for different structural parameters, namely, the number of states per model and number of clusters for quantization. The simulation results on the MNIST with 15 states and a different number of observations are shown in figures 6.7 and 6.8, which are tabulated in tables 6.1 and 6.2:

From simulation results one can notice that for the signals without degradation, the proposed method has similar performance as the conventional HMM. When the number of states is 15 and the number of observations are 400, the recognition rates of the proposed model and conventional HMM are 92.7% and 91.8% respectively; When the number of states is 15 and the number of observations are 625, the recognition rates of the proposed model and conventional HMM are 93.82% and 93.5% respectively.

However, it can be noticed that when the signals are corrupted by noise, the performance of the proposed model drops slower than conventional models when the noise is increased graduately. When the number of changes in the sequence increases to 3, in the case of 400 observations, the recognition rate of the proposed method is 10.73% higher than the conventional one; in the case of 625 observations, the recognition rate of the proposed method is 7.97% higher than the conventional one. It can be considered that the proposed model is more tolerant to the degradations in the signals.

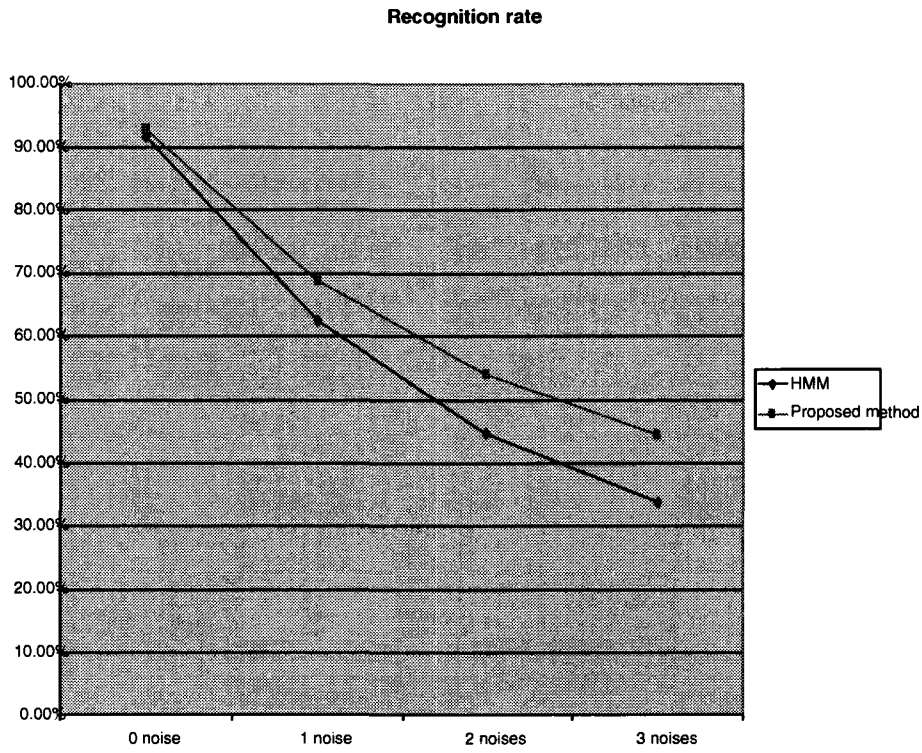

Figure 6.7: Simulation result with the number of state is 15, number of observation is 400

6.3.4 Computational complexity

Classification speed is also of prime importance. For the conventional HMM with T time slot and N hidden state, in the recognition stage, the general computational burden is approximately N^2T. For the proposed method, at the stage of single node states estimation, $N * T$ times multiplications are required. At the stage of coupled nodes states estimation, from equation 6.11 one can notice $2 * N * (T-1)$ times multiplications, $2 * N * (T-1)$ divisions are needed; in equations 6.14 6.15, $N^2 * T$ times additions are required. However, since $L_{ij} = 0$ when $j > i+1$ or $j < i$, actually $N * 2 * T$ times additions are needed. The total computation requirement at the coupled node states are $2 * N * (T-1)$ times multiplications, $2 * N * (T-1)$ divisions and $N * 2 * T$ times additions. And the iteration stage is to repeat this process, assuming r times iterations are required, then the total computational cost in this iteration will r times of above computational complexity. The calculation of the probability of the sequence belonging to model λ $P(O|\lambda)$ is based on the objective functions

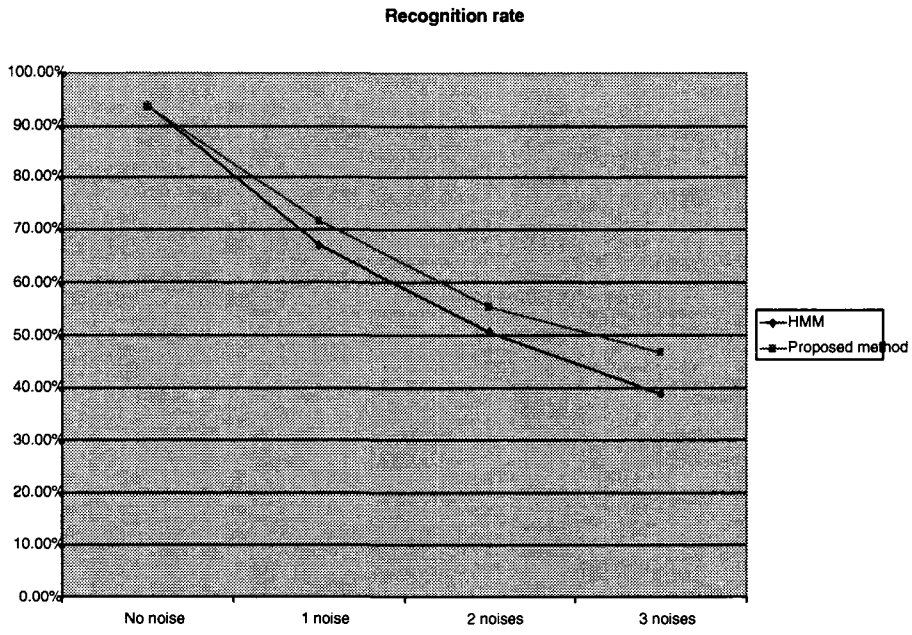

Figure 6.8: Simulation result with the number of state is 15, number of observation is 625

6.18, 6.21. Here approximately $3NT$ times multiplications and $2NT$ times divisions are needed. In summary, the total required computation cost will be approximately: $4NT + 2NTr$ multiplication and $2NT + 2NTr$ divisions.

Some variable issues affect the comparative study of the proposed method with conventional HMM in the respect of the speed issue. The iteration time varies in differenct cases, which affects the computation cost of the proposed method. Thereby for a simple system with small N, the conventional method has less computation cost, and when N is larger than some threshold, the proposed method is more efficient. Assuming the multiplication and division require the same computing time and far more than the addition. The total conmutation for HMM is $2 * N^2 * T$ virsus $6NT + 4NTr$ for our proposed SAHMM method.

6.4 Conclusion

In this chapter we have presented a new non-causal Self-Adaptive Hidden Markov Model, with two training strategies presented for the proposed model. In the first training method, a conventional HMM with the same architecture as the proposed model is established to estimate the optimal

states for every sequence in the dataset. The statistics of the hidden states and the observations can be calculated to obtain the parameters of the proposed model. In the other training method, a local minimum is obtained through iteration of the estimation of the memberships of the hidden states in every time slot in the dataset. Our simulation results show that with the presence of the 3 degradations in every sequence, as in the example of 400 and 625 observations, the recognition rates of the proposed method are 10.73% and 7.97% higher than the conventional one respectively.

For the conventional HMM, the deduction of the hidden states in a sequence is unidirectional and causal, therefore, any noise existing in the sequence may lead the state estimation in a wrong direction, which may yield wrong classification. The proposed model estimates the initial state memberships in every time slot simultaneously, and optimizes the memberships of states in every time slot with neighbors mutually. Consequently, the whole system is more tolerant to the noise in signals. Since the proposed model utilizes the iteration strategy in the evaluation stage, the computation cost is higher than the conventional method. Our simulation results also show that the performance of the proposed model is slightly higher than the conventional one when handling the noise free signals. Such problems should be solved with the further study of such noncausal systems.

Chapter 7

2-D Self-Adaptive HMM

7.1 Two dimentional Hidden markov Models

Images are two dimensional signals, while the model proposed in the last several chapters are one dimensional, which are suitable to 1-D signal processing. Theoretically, 2-D HMMs should yield better results in 2-D signal processing. There are two causal 2-D Markov chains available: the Markov mesh random field(MMRF)[208] and the Nonsymmetric Half-Plane(NSHP)[209] Markov chain. 2-D HMM usually means the MMRF because of its popularity in this field. The major bottleneck of expanding the 1-D HMM into the two dimensional model is its prohibitative computation cost in the training phase. There have been several attempts to extend the 1-D HMM to 2-D HMM[100][61]. The first breakthough was reported in 1998 by Hee-Seon Park[100], where the look-ahead technique was proposed. It is worth noting that the conventional Baum Welch method estimates the hidden states based on all of the elements in a sequence as shown in equation 3.30. The estimation of hidden state $s_{[m,n]}$ proposed by Hee-Seon Park is based on the signal from location $[1,1]$ to $[m+1,n+1]$ as illustrated in figure 7.1. Since the information beyond the $[m+1,n+1]$ is not considered in the training stage, the solution is obviously suboptimal, though the computation cost is reduced to a reasonable level. The model is the third-order Markov Mesh Random Field, which means:

$$P(q_{k,l}|q_{k-1,l}, q_{k,l-1}, q_{k-1,l-1}, ..., q_{1,1}) = P(q_{k,l}|q_{k-1,l}, q_{k,l-1}, q_{k-1,l-1}) \tag{7.1}$$

where $q_{k,l}$ is the current active state at location $[k,l]$.

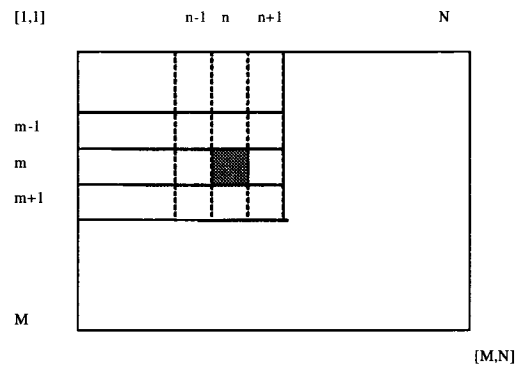

Figure 7.1: Demonstration of the look ahead training method

In the recognition phase, an observation O is recognized as model λ according to:

$$\lambda = \underset{all\lambda}{\operatorname{argmax}} P(O|\lambda) \tag{7.2}$$

The observation likelihood, $P(O|\lambda)$, is approximated by the joint probability of the observation O and the optimal state sequence, $\overline{Q}$, which is estimated by a look-ahead technique given the model λ, i.e.;

$$P(O|\lambda) = P(O, \overline{Q}|\lambda) \tag{7.3}$$

The joint probability of the observation O and a state sequence Q given by the class model λ, $P(O, Q|\lambda)$, is computed by:

$$P(O,Q|\lambda) = P(O|Q,\lambda)P(Q|\lambda) = \prod_{k=1}^{K}\prod_{l=1}^{L} P(q_{k,l}|q_{k-1,l}, q_{k,l-1}, q_{k-1,l-1})P(O_{k,l}|q_{k,l})$$

Obvisouly, the third order models require high computational cost in the evaluation stage. A simplified version of the above model was proposed in [189]. In this model the state in location $[i, j]$ can only be effected by the states in $[i-1, j]$ and $[i, j-1]$, therefore the active states at diagonal neighborhood $[i-1, j-1]$ will be ignored. Another assumption in this method is that the active states of the two observation blocks in anti-diagonal neighborhood locations, e.g. $[i, j-1]$, $[i-1, j]$ are statistically independent. Therefore:

$$P(q_{k,l}|q_{k-1,l}, q_{k,l-1}) = P(q_{k,l}|q_{k-1,l})P(q_{k,l}|q_{k,l-1}) \tag{7.4}$$

Hence, the computation burden will be reduced, while 100% recognition rate in face recognition was reported in [189]. In this case, the probability can be calculated as below:

$$P(O,Q|\lambda) = P(O|Q,\lambda)P(Q,\lambda) = \prod_{k=1}^{K}\prod_{l=1}^{L} P(q_{k,l}|q_{k-1,l})P(q_{k,l}|q_{k,l-1})P(O_{k,l}|q_{k,l}) \qquad (7.5)$$

Some compromising methods are also addressed by scholars, such as Two-Dimensional Psuedo Hidden Markov Model (2-D PHMM)[190][191] and Embedded HMM[192]. A two-dimensional Psuedo Hidden Markov Model consists of a number of superstates, each of them containing a Markov chain. 2-D PHMM is equivalent to a 1-D HMM whose states are lined in one chain according to their internal order within their superstates and the order of the superstates as well. The embedded HMM has a structure that is similar to the 2-D PHMM but the image is scanned in a two-dimensional manner and the 1-D Viterbi algorithm is implemented in two layers. In the first layer the 1-D Viterbi algorithm is used to estimate the probability of each sub-chain of states to generate each row individually. Then, the second layer estimates the overall probability of the main-chain of superstates based on the estimates of the preceding layer.

Similar to the 1-D model, such 2-D models are vulnerable to the influence of the noise mixed with signals. Any wrong estimation of state at present location will heavily effect the following deductions. Here we try to expand the proposed 1-D Self-Adaptive HMM method into a 2-D model and implement it for character recognition applications. Our test is based on the MNIST handwritten numeral dataset. The test results appear to be very promising.

7.2 Outline of the proposed 2-D method

For the conventional Markov Mesh Random Field(MMRF)[100] based 2-D classifiers, the images will be divided into different grids and the relationships among the grids will be analyzed to calculate the probability of the signal belonging to different classes(models). The calculations are performed on all of the grids from left up to right down. From figure 7.2, one can find the observation of every zone in a font is completely different from ones extracted from other fonts. The relationships among the neighbor zones also vary with the variations of the font, written style and distortions of the images. Here we propose a new classification strategy, where the relationship between the strokes can be studied and variations of the written style are ignored. In this way the computation cost is reduced greatly in comparison with other 2-D models. Here we propose a new 2-D Self-Adaptive HMM which is implemented for recognition of the database MNIST. The procedure of the proposed 2-D model consists of several steps:

1. Skeleton points extraction.

2. Feature extraction.

3. 2-D Self-Adaptive HMM based classification.

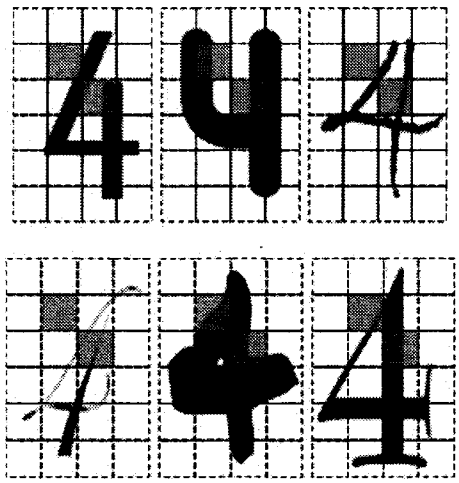

Figure 7.2: Images of character 4 with different fonts

7.3 Skeleton points extraction

As we mentioned in chapter 2, the procedure of OCR can be divided into 2 steps: feature extraction and classification. Most of the feature extractions and classifications for 2-D signal processing are grid-based[193], which means the whole image will be divided into overlapped or non-overlapped grids. Some features[193], such as gradient feature[43], DCT[189] and Gabor features[91], are extracted in every grid to represent the characteristics of the grid. The total information from all of the grids in the images will be inputted to the classifiers such as NN[212], HMM[100]. The major advantage of such methods is the simplicity of feature extraction. However, the variations of the features in every grid for differnt fonts as depicted in figure 7.2 make it difficult for the conventional HMM methods to track and determine the hidden states in the 2-D signals. Training classifiers with larger scale numbers of training data is a straight forward solution, however, over training always means the generation of fuzzier system with degraded performance. On the other hand it is usually difficult to collect enough training data, especially for real world applications.

For character recognition, the structure of the skeletons of characters contain the invariant information of the images of the characters. Skeleton based character recognition is also one of the

mainstream methods[75]. This method can usually be divided into two steps: Skeletonization and skeleton based classification.

Skeletonization plays an important role in digital image processing and pattern recognition, especially for the analysis and recognition of binary images. It has been widely used in OCR and fingerprint recognition[76]. The process can be viewed as a transformation to transform the width of a binary pattern into just one single pixel. Essentially, such a transformation can be achieved by successively removing points or layers of the outline from a binary pattern until all the lines or curves are of unit width, which is called thinning. The resulting set of lines or curves is called the skeleton of the pattern. As we know, the purpose of skeletonization is to reduce the amount of redundant data embedded in a binary image and to facilitate the extraction of distinctive features from the binary image thereafter. A good skeletonization algorithm should possess the following properties: (1) preserving connectivity of the skeleton, (2) converging to skeleton of unit width, (3) preventing excessive erosion, and (4) possessing insensitivity to boundary noise. Until now, thinning has been the most frequently used method to achieve skeletonization goal. Such an iterative process always requires high computation cost.

In the second stage, the structural methods are usually based on the skeleton of a character. The skeleton is first decomposed into a set of primitives, and then features are extracted for each primitive. The topological relations among the primitives are also useful information. There are, however, no general methods for the decomposition and feature extraction. On the one hand, inadequate primitives and features for a pattern result in a high rejection rate and high substitution rate. On the other hand, redundant primitives and features increase computational burden and make recognition more difficult.

The skeleton based method suffers from low speed, noise sensitivity, loss of continuity, and distortion. Another drawback of the skeletonization method is that some critical information will be easily lost after skeletonization, especially if there are self-touching strokes in the characters shown in figure 7.3. From the figure 7.3, one can find that it is even difficult for a human being to distinguish the 3 and 5 from the skeleton of the self-touching characters. This method is also sensitive to blurs and other noises in the character images shown in figure 7.4. One can notice that the distortion and noise in an image will be exaggerated after skeletonization.

Here we try to combine the conventional feature extraction with a skeleton based method. In this way we will extract neighborhood information around skeleton points as features, instead of the neighborhood information in some fixed grids. Here we propose a novel skeleton extraction method with high speed and comparably low accuracy which is acceptable to this application. The procedure

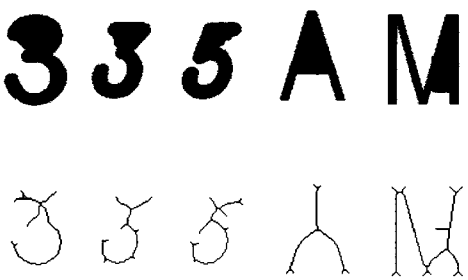

Figure 7.3: Skeletons of self-touching characters

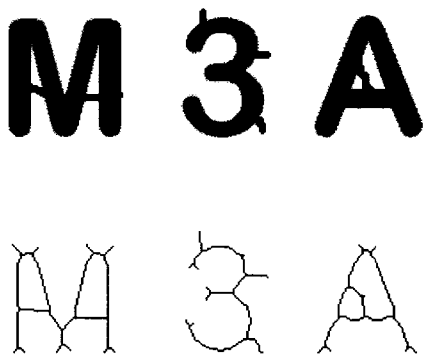

Figure 7.4: Skeletons of characters with blurs

can be depicted as below:

1. Binarization.
2. Skeleton extraction.
3. Skeleton points determination.

Binarization

Most of the skeletonization methods are based on the binary images, therefore it is essential to binarize the image before the skeletonization. Many binarization methods are discussed in chapter

4. The databse used in this experiment is MNIST. Our experiment shows that a simple thresholing method is efficient enough in this application. In this method the total number of non-zero pixels in an image is counted as *NumberofTotalPixels*. We regard a certain number of pixels in an image as foreground, which is $\frac{NumberofTotalPixels}{Coefficiency}$, where the variable *Coefficiency* is constant. It should be mentioned that the accuracy of skeletonization is not critical, because the skeletion points in this application only determine the locations of feature extraction instead of being regarded as feature. In this case we set the *Coefficient* as 7 empirically. From the histogram of the image, we can select the darkest $\frac{NumberofTotalPixels}{Coefficiency}$ pixels as foreground. Our simulation shows that this method is good enough for the purpose, in spite of its simplicity.

Skeleton extraction

Since, in this proposed method, the skeletons do not work independantly, speed instead of accuracy is the major concern in this process. We proposed a new method to skeletonize the images as shown below:

Two two-dimensional arrays, P and Q, are used to store the original binary image and the output (skeleton) respectively. First, a run-length measurement is taken in four directions: left to right, up to down, leftup to righdown, leftdown to rightup. Here the run-length is the number of continuous black points in the run direction. Skeletons are assumed to be the center of the strokes in a character, therefore the skeleton must be in the center of the run-length segment. Thereafter, only the centers of the run-length segment are kept and saved in array Q. When the run-length is even, two center pixels are remained. In this way we can find the skeletons of the characters, although they are coarse and some redundant pixels are left at this stage. The major advantage of this method is its low computational burden. It should be mentioned that any other finer skeletonization method is applicable here as well.

Skeleton points determination

In this method we will not take features from the skeleton pixels themselves, because there is a risk of loss of information in the stage of skeletonization. The skeleton points will be regarded as the critical points and features will be extracted around the critical points based on the grey image pixels. Thereby, it is unnecessary to take features around every skeleton pixels. To reduce the computation cost, we subsample the skeleton pixels to make sure no two skeleton points are immediate neighbors to each other as shown in figure 7.5. The details of the process are explained as below: Select one pixel from the skeleton pixel and eliminate the neighbor skeleton pixels around

it within the window size of 3, and move to the next skeleton pixel that has not been eliminated; repeat the above process until the skeleton pixels are equally distributed in the image. Through the above process, the connectivity of the neighbor skeletons can be found and saved for next step processing as shown in figure 7.5. The connectivity here is based on the corresponding binary image of the character and defined as "If some continuous black pixels can compose a line to connect two neighbor critical points, the two critical points are regarded as connected, otherwise they are regarded as non-connected."

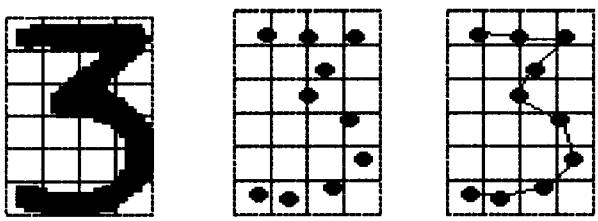

Figure 7.5: Illustration of skeleton points determination

7.3.1 Feature extraction

In this step we will extract the features in the neighborhood around every critical point. The features are extracted from pixels in a rectangle window around every criticl point and the features will be quantized for the next step process. To simplify the simulation, only raw data of the pixels are regarded as a feature in this case. The size of the window is a critical issue to the performance of the whole process. Bigger window sizes means bigger coverage and usually yield better results, however, a big sized window also means larger computational burden. For example, if the size of window increase from a to $a * c$, where c is a real number bigger than 1, the size of the feature space will expand from 256^{a^2} to $256^{(a*c)^2}$, which means it will be $256^{(a*c)^2-a^2}$ times bigger than the original one. The bigger the feature space means bigger the number of clusters and much more computation cost.

To reduce the computation cost we propose a new feature extraction method. Instead of taking the 2-D features around a skeleton point in a window, we take pixel values in the gray images across the skeleton point in 4 directions. For instance, in a 7×7 window around pixel $P(x, y)$ as shown in figure 7.1, for the conventional method, all of the pixels in this window will be regarded as a feature vector. In the proposed method, there are only 4 1-D vectors:

Table 7.1: Demonstration of feature extraction

P(x-3,y-3)	P(x-3,y-2)	P(x-3,y-1)	P(x-3,y)	P(x-3,y+1)	P(x-3,y+2)	P(x-3,y+3)
P(x-2,y-3)	P(x-2,y-2)	P(x-2,y-1)	P(x-2,y)	P(x-2,y+1)	P(x-2,y+2)	P(x-2,y+3)
P(x-1,y-3)	P(x-1,y-2)	P(x-1,y-1)	P(x-1,y)	P(x-1,y+1)	P(x-1,y+2)	P(x-1,y+3)
P(x,y-3)	P(x,y-2)	P(x,y-1)	P(x,y)	P(x,y+1)	P(x,y+2)	P(x,y+3)
P(x+1,y-3)	P(1,y-2)	P(x+1,y-1)	P(x+1,y)	P(x+1,y+1)	P(x+1,y+2)	P(x+1,y+3)
P(x+2,y-3)	P(x+2,y-2)	P(x+2,y-1)	P(x+2,y)	P(x+2,y+1)	P(x+2,y+2)	P(x+2,y+3)
P(x+3,y-3)	P(x+3,y-2)	P(x+3,y-1)	P(x+3,y)	P(x+3,y+1)	P(x+3,y+2)	P(x+3,y+3)

1. $[P(x-3,y), P(x-2,y), P(x-1,y), P(x,y), P(x+1,y), P(x+2,y), P(x+3,y)]$
2. $[P(x-3,y-3), P(x-2,y-2), P(x-1,y-1), P(x,y), P(x+1,y+1), P(x+2,y+2), P(x+3,y+3)]$
3. $[P(x,y-3), P(x,y-2), P(x,y-1), P(x,y), P(x,y+1), P(x,y+2), P(x,y+3)]$
4. $[P(x-3,y+3), P(x-2,y+2), P(x-1,y+1), P(x,y), P(x+1,y-1), P(x+2,y-2), P(x+3,y-3)]$

After the extraction of the critical points in the database, the 4 1-D feature vectors around every critical point will be collected to form a codebook. The Kohonen SOM[195] is used to construct the code book vectors, thereby the feature space can be divided into different clusters. According to the Euclidean distance to the centers of clusters we can quantize the 4 1-D feature vectors. It should be noted that the SOM is utilized here instead of the K-Mean. The continuity of neighbor center vectors from SOM makes it possible to implement some searching strategy to speed up the quantization stage in the future if necessary, while for the K-Mean, the neighbor center vectors have no such feature.

After obtaining the center vectors from the SOM, we can quantitize the four feature vectors in every location into 1 vector with 4 elements $[V_1V_2V_3V_4]$. The advantage of such a method is the realistic computational burden with robust performance. Assuming that the size of window is N by N and the number of clusters is M, for conventional 2-D window based quantization method $M \times N \times N$ times multipications and $M \times N \times N$ times additions are needed. For the proposed feature exttraction, only $4 \times M \times N$ times multipications and $4 \times M \times (N-1)$ times additions are needed. One can tell, when N is bigger than 4, the proposed feature extraction will be much faster than the conventional 2-D method. Here we consider the length of the sliding window around every skeleton point to be $5 * 2 + 1 = 11$. Thereby for every selected skeleton point, four observations(quantized feature vectors) are generated for next step classification.

7.3.2 Parameters in 2-D Self-Adaptive Model

Here a novel 2-D Self-Adaptive HMM is introduced. The same as the 1-D Self-Adaptive Model, the 2-D model is composed of following elements: $I \times J$ number of hidden states and M number of observation symbols in the alphabet. Differentiating from the hidden states in 1-D SAHMM, the states in a 2-D system will be indexed in 2 directions i, j. Some hypotheses about the relationships of the elements are proposed and the model is characterized by the following progess, which are slightly different from the 1-D model:

1. States combination probability $L^{(d)}_{ij-\tilde{i}\tilde{j}}$, where d is the direction of the link and $1 \leq d \leq 8$; ij and $\tilde{i}\tilde{j}$ are the coordinates of the hidden states at the two neighbor locations.

2. Probability of distribution of observations to every state $B = b^d_{ij}(k)$, $[i, j]$ is the coordinate of the states, k is the symbol of the obervation, the superscript d is the direction of the four 1-D vector around the point.

3. The distribution probability of states in every location of the 2-D observation sequence $D_{xy}(ij)$, x,y is the coordinate of the critical point, while i,j are the coordinate of the hidden state.

4. Probability of the state $[i, j]$ occurring in the model is $C_{i,j}$.

Then the compact notation λ for the proposed model is usually used to combine all of the parameters,

$$\lambda = (L, B, D, C) \tag{7.6}$$

The same as in the 1-D system, the 2-D model consists of number of states and number of observation symbols in the alphabet. However, in the 2-D model, the states are not indexed by one number any more, because every node has more than left or right neighbor nodes as in the 1-D system. Therefore the transition from one state to another is more than two direction. Hence, we define the $L^{(d)}_{ij-\tilde{i}\tilde{j}}$, where d is the direction of the link. Here we define the direction as a mutual link in 8 directions as shown in figure 7.6.

Assuming the differential vector of the locations of two ends E_1, E_2 of a link is $[D_i, D_j]$, and the states in the two ends are $S_{i_1 j_1}$, $S_{i_2 j_2}$, some constraints are defined as below:

$$L^{(d)}_{ij-\tilde{i}\tilde{j}} = 0 \text{ When } D_i > 0 \text{ and } i_2 - i_1 < 0 \tag{7.7}$$

$$L^{(d)}_{ij-\tilde{i}\tilde{j}} = 0 \text{ When } D_i < 0 \text{ and } i_2 - i_1 > 0 \tag{7.8}$$

$$L^{(d)}_{ij-\tilde{i}\tilde{j}} = 0 \text{ When } D_j > 0 \text{ and } j_2 - j_1 < 0 \tag{7.9}$$

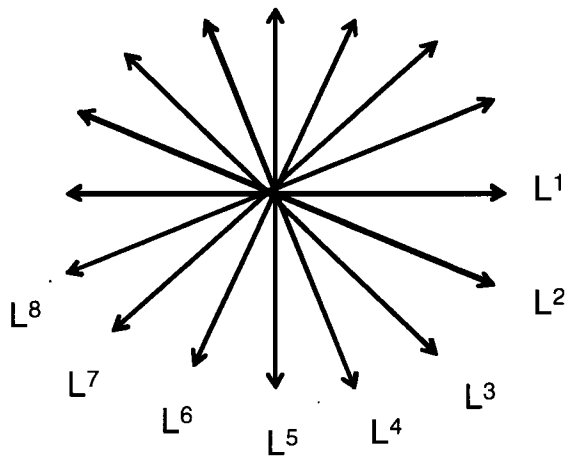

Figure 7.6: 8 directional mutual links

$$L^{(d)}_{ij-\bar{i}\bar{j}} = 0 \text{ When } D_j < 0 \text{ and } j_2 - j_1 > 0 \tag{7.10}$$

And a non-skip scheme is utilized here, which means $-1 \leq i_2 - i_1 \leq 1$ and $-1 \leq j_2 - j_1 \leq 1$. So:

$$L^{(d)}_{ij-\bar{i}\bar{j}} = 0 \text{ When } i_2 - i_1 > 1 \text{ or } i_2 - i_1 < -1 \text{ or } j_2 - j_1 > 1 \text{ or } j_2 - j_1 < -1 \tag{7.11}$$

The distribution probability in the 2-D model is expanded to two dimensional, which is $D_{xy}(S_{ij})$. The distribution of the hidden states are supposed to be a Gaussian distribution and the distribution can be illustrated as in figure 7.7. Therefore, we quantize the distribution probability in this way that the image of a character is divided into 5×5 zones. All of the pixels in one zone have identical distribution probability.

The definition of observation distribution probability $B = b^d_{ij}(k)$ is very similar the one defined in the 1-D model. The major difference is that in this case there are four observations for every hidden state. To simplify the calculation, we assume the four observations are independant to each other. After multiplying the probability of the four observations together, we can easily obtain the general probability of the four observations belonging to the location.

$$P_t(O_t|s_{ij}) = \prod_{d=1}^{d=4} b^d_{ij}(o_t(d)),\ 1 \leq i \leq I,\ 1 \leq j \leq J,\ 1 \leq O_t \leq M \tag{7.12}$$

The probability of state $[i, j]$ occurring in model C_{ij} is the same as the parameter C_t in 1-D model, except that the indexs are expanded into two dimensional.

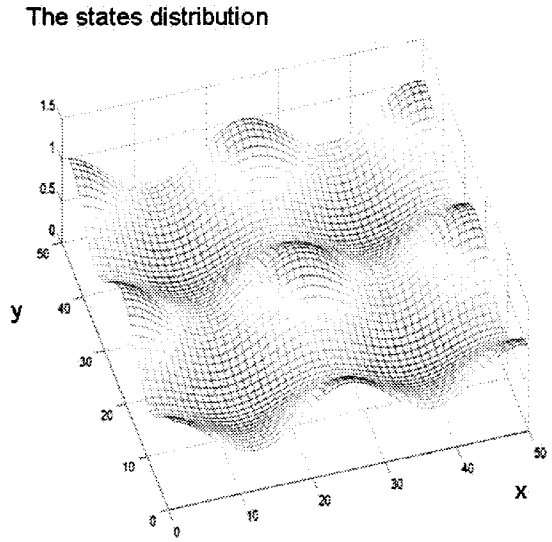

Figure 7.7: Illustration of the 3 by 3 states distributing on a planar

7.3.3 Evaluation stage

The evaluation stage of the 2-D SAHMM is similar to the 1-D SAHMM. There are two major differences between the two systems: To simplify the computation cost, the iteration in 1-D system is cancelled in the proposed 2-D system; and disconnected neighbor critical points are independent to each other. The detail of the process is described as below:

Single node states estimation

The estimation of the hidden states at critical points in an image will start from every single obervation. The memberships are decided by two issues: the location of the critical point and the observations at this critical point. The first one can be looked up from the $D_{xy}(S_{ij})$, while the second one depends on observations symbol probabilities distribution $b_{ij}^{d}(k)$. The general probability can be calculated as below:

$$P_t(O_t, S_{ij}|xy) = P(S_{ij}|xy)P_t(O_t|S_{ij}, xy) = D_{xy}(S_{ij}) \prod_{d=1}^{d=4} b_{ij}^{d}(o_d) \tag{7.13}$$

Here i,j are the coordinate of the hidden state; x,y are the coordinate of the critical point; o_d is the observation in the d direction.

Neighbor nodes states estimation

In this stage we start to consider the influence of neighbor nodes on each other. In the conventional method, the memberships of hidden states in $S_{i,j}$ will be determined by the past neighbor nodes $S_{i-1,j}$, $S_{i,j-1}$, $S_{i-1,j-1}$. There are two drawbacks in such methods. First, the same as in 1-D HMMs, the hidden states are only decided by past states, while the information from 'future' nodes, (in the case of image, they are the nodes at the right or down side of the present node), are ignored. Actually the 'future' information is available as well as the 'past' information, when HMM is utilized as a classifier. Secondly, any past neighbor nodes have to be considered, as shown in figure 7.2, even though there is no inherent relationship between some of the points.

Here we propose a new method, where only the connected neighbor critical points will be considered for the estimation of hidden states. As depicted in figure 7.8, there are 4 direct neighbors *a b d e* skeleton points around the point *c*. Only points *b d* are connected to the point *c*, therefore, points *a e* will be ignored when calculating the combined link possibility of point *c*, since they are not directly connected to *c*.

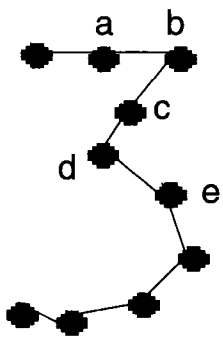

Figure 7.8: Illustration of connected critical points in an image

For a skeleton point $[x, y]$ in a character image we can find U pieces of directly connected points around it, therefore U number of links around it. For a link u connected to point $[x, y]$, the coordinate of another end of the node is $[x'y']$. From the last step, we have calculated the membership of hidden state as $P_t(O_t, S_{ij}|xy)$ and $P_t(O'_t, S_{i'j'}|x'y')$: similar to equation 6.11 in 1-D model, a link u can be calculated as:

$$L^u_{ij,i'j'} = \frac{P_t(O_t, S_{ij}|xy)}{c_{ij}} L_{ij,i'j'} \frac{P_t(O'_t, S_{i'j'}|x'y')}{c_{i'j'}} \tag{7.14}$$

Thereby the membership of the hidden states in the two nodes will be:

$$P_{xy}^{u} S_{ij} = \sum_{i'=1}^{I} \sum_{j'=1}^{J} L_{ij,i'j'}^{u} \tag{7.15}$$

$$P_{x'y'}^{u} S_{i'j'} = \sum_{i=1}^{I} \sum_{j=1}^{J} L_{ij,i'j'}^{u} \tag{7.16}$$

Then considering all of the influences from the connected neighbors around a node, one can get:

$$P_{xy}(S_{ij}) = \prod_{u=1}^{U} P_{xy}^{u}(S_{ij}), \tag{7.17}$$

With the same method, we can easily find the membership of the hidden states in every critical point in an image.

Objective functions:

After the above calculation, we can obtain the estimate of the memberships of states in every nodes. A normalization method is used in every critical point:

$$P'_{xy}(S_{ij}) = \frac{P_{xy}(S_{ij})}{\sum_{i=1}^{I} \sum_{j=1}^{J} P_{xy}(S_{ij})} \tag{7.18}$$

The probability of every node in the sequence generating the corresponding observation can be calculated as below:

$$P(O) = \sum_{i=1}^{I} \sum_{j=1}^{J} P_{xy}(ij) b_{ij}(O_{xy}) \; 1 \leq t \leq T \tag{7.19}$$

where n is the iteration time.

Assuming there are totally K critical points and for every point, the probability is P_{ij}^{k}, then the general probability of every point occurring in the 2-D model will be:

$$P(O|\lambda) = \prod_{t=1}^{K} P^{k}(O) \tag{7.20}$$

It may happen that some probabilities of nodes are 0, then after multiplication the general result will be 0. In such a case we will generally substitue the 0 with small positive value to avoid a rigid result.

When the model is implemented for the classification application, the self-adaptive model based recognizer aims to determine the model that it is most probable the one that produces the provided test signal. i.e.

$$\lambda = argmax P(\lambda|O) \tag{7.21}$$

This is a problem of Maximum A Postriori Probability(MAP), where Bayes rule is utilized to reformulate the recognition problem as:

$$\lambda = argmax \frac{P(O|\lambda)P(\lambda)}{P(O)} \tag{7.22}$$

Since all models are considered equally probable, equation7.21 can be simplified as:

$$\lambda = argmax P(O|\lambda) \tag{7.23}$$

In the evaluation stage, extracted features for every image are sent to every model in the system to find the λ with the maximum $P(O|\lambda)$.

7.3.4 Training stage

Here the iterative strategy is utilized to train the model. First, we assign initial values to the parameters λ in the model. In this simulation we divided the character image whose size is 20 by 20 into $X = 5$ by $Y = 5$ zones to quantize the states distribution probability $D_{xy}(S_{ij})$; the size of the hidden states is $I = 3$ by $J = 3$; the number of observations is defined as 64.

After estimating the fuzzy membership of states in every critical point in every image of the database, we can obtain the new parameters in the model just as in the method used in last chapter. After several iterations, a new set of parameters will be available. Our simulation shows that this method can converge into stable local maximum. However the performance of the model will depend highly on the intial values because only a local maximum is reached at the end. Here the decision-directed estimation algorithm is proposed to estimate the initial parameters in the model to improve the performance of classifier.

Decision Directed Estimation(DDE) method

A decision directed estimation method is a straight-forward way to find the inherent relationship of the states and observations. The details can be depicted as below:

1) Distribution probability

Assuming that the states distribution $P(S_{ij}|xy)$ (x,y is the quantitized coordinate of the critical points, i,j is the state) is an exponential $a_{ij}e^{-(\frac{(x-b1_i)^2}{c1_i^2}+\frac{(y-b2_i)^2}{c2_i^2})}$, where $1 \leq x \leq X$, $1 \leq y \leq Y$ and the a,b1,b2,c1,c2 is the coefficient of the two dimensional Gaussian function, and [b1, b2] refers to the center of the Gaussian distribution. Initially we suppose the center $[b1, b2]$ is equally distributed in the sequence. Then:

$$b1_i = \frac{X}{(I-1)} * (i-1) \; 1 \leq i \leq I \tag{7.24}$$

$$b2_i = \frac{Y}{(J-1)} * (j-1)\ 1 \leq i \leq J \tag{7.25}$$

$c1_i$ $c2_i$ are experimentally obtained values and considered to be 2 here. After normalizing the states distribution the probability of state $[i,j]$ occurring at the location $[x,y]$ would be $D_{xy}(S_{ij})$:

$$D_{xy}(S_{ij}) = \frac{P(S_{ij}|xy)}{\sum_{i=1}^{I}\sum_{j=1}^{J} P(S_{ij}|x,y)} \tag{7.26}$$

Since this is just an initial estimate, the distribution parameters of each model for every character in this case is assumed to be the same.

2) The initial estimate of the probability of every state $[i,j]$ existing in the model is $C_{i,j}$ that can be derived from $D_{xy}(S_{i,j})$.

$$C_{i,j} = \sum_{x=1}^{X}\sum_{y=1}^{Y} D_{xy}(i,j) \tag{7.27}$$

3) The initial estimation of the probability of the combination of state ij and $i'j'$ in a model can be $L^{(d)}_{ij,i'j'}$:

Here we assume there is an equal chance for every $L^{(d)}_{ij,i'j'}$ occurring in a model, except in some cases indicated in equations 7.11, where $L^{(d)}_{ij,i'j'}$ would be be 0.

4) The observation probability distribution in each of the states, $B = b_{ij}(k)$ can be calculated as below:

From step 1, we can get the membership of states in every cirical point in every character image. For example there are total W character images in the database, in the image wth, at location $[x,y]$ the observation is k, then the chance of it is generated by state S_{ij} is $P_{w[x,y]}(S_{ij}) = D_{[x,y]}(S_{ij})$. Consequently, the observation k belonging to state S_{ij} here is $P_{w[i,j]}(k|S_{ij}) = D_{x,y}(S_{ij})$. It's easy for us to accumulate the expectation value $E(S_{ij})$ of state S_{ij} from the W images in the database:

$$E(S_{ij}) = \sum_{w=1}^{W}\sum_{x=1}^{X}\sum_{y=1}^{Y} P_{w[x,y]}(S_{ij}) \tag{7.28}$$

And the expectation value of state $[i,j]$ when the observation is k is:

$$E(k|S_{ij}) = \sum_{w=1}^{W}\sum_{x=1}^{X}\sum_{y=1}^{Y} P_{w[x,y]}(k|S_{ij}) \tag{7.29}$$

Then we can get the observation distribution in state [i,j] as:

$$O(k|S_{ij}) = \frac{E(k|S_{ij})}{E(S_{ij})}\ 1 \leq j \leq N,\ 1 \leq k \leq K \tag{7.30}$$

We can easily know that

$$\sum_{k=1}^{K} O(k|S_{ij}) = 1 \; 1 \leq j \leq N \; 1 \leq k \leq K \tag{7.31}$$

Up to now we have finished the estimation of the initial values of the parameter λ in the model. Inputting the initial values into the model to continue the training, we will find the better results can be obtained than from random initial values.

7.3.5 Simulation results

We carry out preliminary experiments to estimate the performance of our classifiers. The database used here is the handwritten digits in the MNIST database. The size of the image will be downscaled from 28 by 28 to 20 by 20.

HMM has been implemented for character recognition and the test results in different dataset are shown in table 7.2. From the table, one can notice, because of the variation of database, various performances are obtained. The recognition rates not only depend on the sufficiency of the methods, but are also affected by the database itself, such as the lexicon size, degradation of the signals, size of the database,e.t.c. Therefore it is difficult to compare the performances of different models only based on the recognition rate. Our proposed method's recognition rate is 5.6% higher than the conventional 2-D HMM[100], whose experiment is based on the database of CENPARMI(Center for Pattern Recognition and Machine Intelligence). The MNIST database that we used contains 60,000 handwritten digits in the training set and 10,000 handwritten digits in the test set. CENPARMI is generated by Concordia University of Canada. It consistes of 6000 unconstrained numberals, while 4000 of them are used for training and 2000 of them are used for testing. In this way the dataset of the MNIST is almost 10 times larger than CENPARMI. A dataset with larger size usually contains more variation of the degradation, and it is more challenging. One can say that the performance of the proposed method is better than the conventional 2-D HMM[100].

Various pattern recognition technologies were implemented in the character recognition problem[43]. Different classifiers tested on this database MNIST had shown very high recognition. The comparative results of our proposed method and other methods are tabulated in table 7.3. In the table the RR stands for Recognition Rate; CC is Computation Cost and it is number of 1000 Multiply-Accumulate Operations(MAO) for the recognition of a single character starting with a size-normalized image; Memory in the table is measured in 1000 variables for each of the methods.

The computation cost of proposed 2-D SAHMM varies according to the number of the critical points and number of connected neighbor critical points. Our simulation results show there are

Table 7.2: Comparison results of HMM based OCR engine

Method	Classifier	Lexicon Size	RR(%)	Test Set	Database
(Bunke95)[196]	HMM	150	98.4	3,000	WORDS(eng)
(Mohamed96)[197]	HMM-DP	100	89.3	317	City names
(Knerr98)[198]	HMM-NN	30	92.9	40,000	LA words
(Guillevic98)[199]	HMM-k-NN	30	86.7	4,500	LA words
(Yacoulbi99)[200]	HMM	100	96.3	4,313	City names
(Yacoulbi99)[200]	HMM	1,000	88.9	4,313	City names
(Kim00)[201]	HMM-MLP	32	92.2	2,482	LA words
(Freitas01)[202]	HMM	39	77	2,387	LA words
(Oliveira02)[203]	MLP	12	87.2	1,200	Month words
(Xu02)[204]	HMM-MLP	29	85.3	2,063	Month words
(Kundu02)[205]	HMM	100	88.2	3,000	Postal words
(Arika02)[206]	HMM	1,000	90.8	2,000	Words
(Kapp04)[207]	HMM-MLP	39	81.7	2,387	LA words
(Koerich04)[210]	HMM	1,000	91	4,674	City names
(Parker98)[100]	2d HMM	10	90.80	10,000	CENPARMI
Proposed	2D SAHMM	10	96.4	10000	MNIST

Table 7.3: Comparison results from MNIST

Classifier	RR(%)	CC(k)	Memory(k)
Linear classifier[43]	88	4	4
nearest neighbor-NN[43]	91.60	24000	24000
Pairwise linear classifier[43]	92.40	36	35
K-NN Euclidean[43]	95.00	24000	24000
2-layer NN,300 hidden units(20*20*300*10)[43]	95.3	123	123
2-layer NN,1000 hidden units(28*28*1000*10)[43]	95.50	795	795
1000RBF + linear classifier[43]	96.40	794	794
40PCA + quadratic classifier[43]	96.70	39	40
3-layer NN, 300+100HU[43]	96.95	267	267
3-layer NN, 500+150HU[43]	97.05	469	469
K-NN Euclidean, deslant[43]	97.60	24000	24000
LeNet-1[16*16][43]	98.30	100	3
Boosted LeNet-4[distortions][43]	99.30	460	24000
Virtual SVM poly 5 [distortions][43]	99.20	28000	28000
HMM[212]	94.19	N/A	N/A
Combination of HMM and SVM[212]	98.02	N/A	N/A
Markov Random Field-based[213]	94.60	N/A	N/A
Proposed Method	96.4	45	26

average 16 critical points in a character image and 2.3 connected neighbors to every nodes. At the stage of single node states estimation, from equation 7.13 one can notice for every critical point $9\times4 = 36$ multiplications are required. At the stage of neighbor nodes states estimation, $2.2\times2 = 4.4$ multiplications are required in equation 7.14, and 9 addations are required for equation 7.15. When calculating the objective functions, totally $9 + 9 = 18$ additions and 9 multiplications are needed. Considering there are average 16 critical points in every image, the total required additions and multiplications needed are $16 \times (9+9+18) = 576$ and $16 \times (36+9) = 720$. Generally no more than 720 times of multiply-accumulate operations for the recognition of a single character are required, which are much smaller than any classifiers mentioned in the table7.3. Quantizations are required in the discrete model. There are 4 observations for every location, 64 clusters, average 16 critical points in every image and the legnth of the vector is 11 elements, therefore $4 \times 64 \times 16 \times 11 = 45056$ multiply-accumulate operations(MAO) are needed for quantization. For the discrate model, the total calculational complexity are $45056+720 = 45776$ multiply-accumulate operations, which is still smaller than most of other methods with similar performance. The major machine time consumed here is the quantization. Since the Self Organizing Map (SOM)[92] method is utilized here to quantize the feature vectors, the central vectors of the clusters from SOM are continuous. It is possible to speed up the quantization several times faster with some hierarchial strategies, which is out of range of this thesis.

A summary of the performance of our proposed classifier and other mainstream classifiers is shown in table 7.3. The SVM[211] and LeNet-4[43] are reported to yield the highest performances. Support vector machines (SVMs) were introduced in [35] as learning machines with capacity control for regression and binary classfication problems. In the case of classification, a SVM constructs an optimal separating hyperplane in a high-dimensional feature space. The computation of this hyperplane relies on the maximization of the margin. Because SVM is a binary classifier, to implement it in multi-class classification problems, the most common solution is that a SVM based classifier is built for every pair of classes to separate the classes two by two. LeNet5 takes a raw image of 28 by 28 pixels as input. It is composed of 7 layers: three convolutional layers (C1, C3 and C5), two subsampling layers (S2 and S4), one fully connected layer (F6) and the output layer. From table7.3, the computation cost and memory requirement of the SVM are 622 and 1077 times higher than the proposed method; and the computation cost and memory requirement of the LeNet-4 are 10 and 923 times higher than the proposed method. Therefore the proposed method is more suitable for real world applications.

For the proposed 2-D SAHMM, there are four sets of parameters $\lambda = (L, B, D, C)$ in the model.

Table 7.4: Learning curves of proposed 2D SAHMM

Iterative time	1	2	3	4	5	6	7	8
Initial values from DDE	76.00%	86.00%	90.00%	94.70%	95.20%	96.00%	95.60%	95.80%
Random (case 1)	57.10%	59.00%	68.00%	71.20%	77.90%	75.00%	76.40%	73.00%
Random (case 2)	51.00%	55.30%	59.10%	67.20%	71.20%	74.70%	74.80%	71.20%

In this experiment, we divide the image into 5 by 5; the number of hidden states is 3×3; the number of obervations is 64. Then there are $5 \times 5 \times 3 \times 3$ parameters in $D_{xy}(S_{ij})$; there are $8 \times 3 \times 3 \times 2$ parameters in $L^{(d)}_{ij,i'j'}$; C_{ij} contains 3×3 data; $O(k|S_{ij})$ has $3 \times 3 \times 4 \times 64$ variables. Since there are a total of 10 classes in the system, the memory requirement of the system are 26820 float numbers. Therefore, the memory requirement of the NN is 8.8 times higher than the proposed method with similar performance. Other classifiers, such as,k-NN, SVM, LeNet-4 are reported to require even more memory than the NN mentioned here[43].

Test results with the iterative training

In order to exmine the validity of the proposed iterative training algorithm, we firstly use the direct decision method to obtain the initial values and then iteratively train the model until the recognition rate stopped increasing. In this simulation, we divided an image into 5×5 zones to quantize the states distribution probability $D_{xy}(S_{ij})$. The size of the hidden states is 3×3 and the number of observations is defined as 64. The learning curve can be checked in figure 7.9, where three sets of learning curves are presented in the diagram. One set of initial values in the parameters are derived from the decision directed estimation, and the other two sets of initial values are just random valid values. Though all of the iterative training can converge to a local minimmum, the deliberatly selected values lead to optimal result. The details of the recognition rates can be found in table 7.4.

7.4 Conclusion

In this chapter we present a new non-causal 2-D Self-Adaptive Hidden Markov Model(2-D SAHMM), which require lower memory and computational burden in comparison with other classifiers with the same recognition rate. Like conventional HMM and proposed 1-D SAHMM, this proposed model is composed of two layers: the hidden states and observations. First, the skeleton of a character

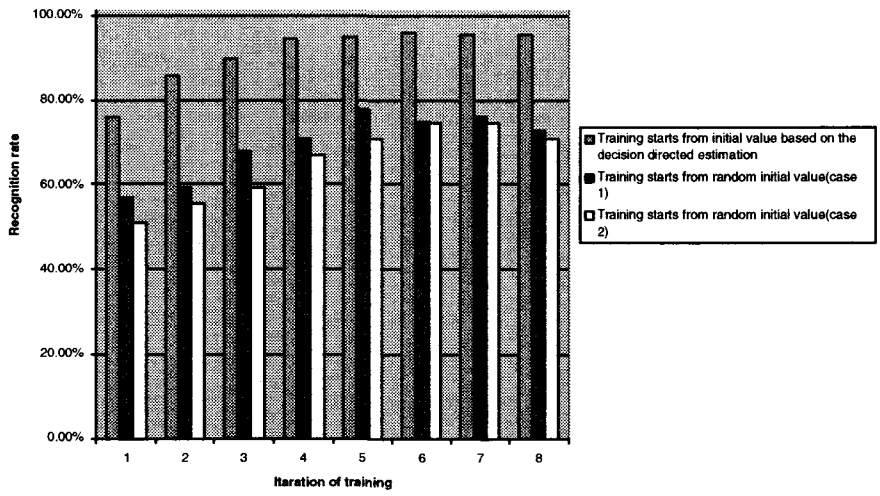

Figure 7.9: Learning curves for iterative training algorithm

will be extracted from the binarized image. After subsampling, some of the skeleton points will be selected as critical points for next step feature extraction. After feature extraction, the hidden states will be estimated based on the observation around every critical points and their connected neighbor critical points. The final probability of the image belongs to model can be calculated from the proposed objective functions.

Some new methods and concepts are introduced in this model. Most of the other gird based methods extract features from all of the grids in a plane and all of the features will be inputted into the classifers. In this proposed method, features are only extracted from the neighborhood of the critical points. In this method the influence of the variation of fonts, written styles and degradations from the environment noise can be minimized. At the same time, the computation cost will be reduced. In the conventional 2-D HMMs, the deduction of hidden states are based on the states in the left, up, or left-up immediate neighbors grids and the information from other neighbors are ignored. Obviously the imperfect deduction will yield a degraded performance. As a noncausal system, within the proposed model the connected neighbor points will mutually affect each other. Only the influence from connected neighbor points will be considered, which is a suitable strategy for binary image processing. A new feature extraction method is utilized in the method. In this way,

one 2-D feature matrix will be substitued by 4 1-D feature vectors. Therefore the computational complexity will be dramatically reduced with acceptable performance.

There are several prominent differrences between the 2-D system and the 1-D system introduced in the the last chapter. Besides the fact that the index of the states are expanded to 2-D, some strategies in the two systems are different. In the 1-D SAHMM system the evaluation will be iterated to make the state propagate along the 1-D path. To reduce the computation cost, iteration is not utilized in the 2-D system anymore and the recognition rate is 96.4%, which is still higher than the 1-D system. In the objective functions, the link probability shown in equation 6.21 is also not considered in the 2-D system to further increase the speed.

Chapter 8

Conclusion and future research

8.1 Conclusions

In this thesis two novel binarization methods are introduced here. A HMM based binarization method is presented in chapter 4. Our test results show the binarized images from the proposed method is much more robust than the ones from other references[107][108][110]. A commercial OCR engine is tested on 42 binarized images from proposed method and other reference binarization strategies. A recognition rate of 77% is obtained using the proposed technique while the closest performance was that of Otsu's which yielded 53% correct rate. It should be mentioned that this binarization method is a local pixel's characteristics based binarization method. HMM functions as a classifier to identify the attribute of every pixel (foreground or background) according to the features of neighborhood around every pixel. For the local pixel's characteristics based binarization method, the major drawback is that the computation cost is extremely high, which hampers its implementation in real world applications. With the aid of look up table and other strategies, this proposed method is fast enough for most of the real world applications without degradation of its performance. The size of the reference set is only 10k bits, which is acceptable to real world applications as well.

Due to its high efficiency, histogram based binarizations are the most prefered methods. However such kind of methods can only work well when the histogram of an image is bimodel, which is untrue for most of the real world images. An edge based binarization method is also proposed in the thesis.

In this method after the edges are detected in the process, some pixels around edges are selected from an image to represent the foreground and the background. A recognition rate of 67.3% is obtained using the proposed technique while the closest performance was that of Otsu's which yielded 53% correct rate. The binarization is based on the histogram of selected pixels. One can notice the histogram is usually bimodel, since most of the disturbing background information is ignored after the edge detection. The promising experiment results of the proposed method are demonstrated in chapter 5.

The proposed binarization methods are effective for different types of applications. The HMM based binarization is based on the feature of the neighborhood around every pixel and the training set is selected from the strokes of the characters in different images. Thereby at the recogizing stage, pixels in strokes and stroke-like objects are easily discriminated, even though it may be difficult for other conventional methods. For the edge based method, robust edges and their neighbors are selected to decide the suitable threshold for the whole image or different regions in an image. Less computation complexity is expected in comparison with HMM based method. This method is more suitble for images with simple background, otherwise the edges generated from background will degrade the whole performance. The edge based binarization can be implemented for various applications; while the HMM based method is most suitable for character extraction applications.

For the conventional HMM the deduction of the hidden states in a sequence is unidirectional and causal, therebfore, any noise existing in the sequence may lead the state estimation to wrong direction, which may yields wrong classification. Here we propose a new Self-Adaptive HMM. The same as the conventional HMM, every model has two layers: observation and hidden state. The proposed model estimates the initial state membership at every time slot simultaneously, and optimizes the memberships of states in every time slot with neighbors mutually.A iterative asynchronous method is utilized in the evaluation stage and solves the noncausal problem successfully. The application of this new proposed model for off-line optical character recognition is tested in this thesis. Test results indicates a 93.82% recognition rate for a noise free dataset which is close to a coventional HMM performance. However with the presence of severe noise a 9% improvement in recognition performance over conventional HMM is obtained.

To further enhance the performance of the proposed SAHMM in optical pattern recognition, we expand the proposed SAHMM system into two dimensional. Non-causal strategy is still implemented in this method with some new concepts. After skeletonization and subsampling, the sparse skeleton points will be chosen as critical points, around which some features will be extracted. In the new feature extraction method the 2-D feature matrix around every critical point will be substitued by

4 simpler 1-D features to obtain the optimal tradeoff of coverage of feature extraction window and computation cost. The hidden states at every critical points will be estimated from the features there and the location of the critical point, while the connected neighbor hidden states will affect each other mutually. This method is more tolerant to the variation of the images and promising result is reported when it is implemented in MNIST. Our experiment shows 96.4% recognition rate is obtained with smaller memory requirement and lower computational complexity in comparison with other classifiers with similar performance.

8.2 Future work

All of the HMMs mentioned in this thesis are discrete HMM. The continuous HMM is the more general HMM type. The only difference of the two types is that the $b_j(O_t)$ are expressed in different ways, as shown in equation 3.44 and 3.45. The memory requirement of continuous HMM will be slightly higher than the discrete HMM. Since the quantization stage in a discrete HMM will be substituted by the sum of finite number of Gaussian distributions in continuous HMM, the computational burden will be released slightly, if the continuous HMM is implemented here instead of the discrete HMM. Corresponding continuous HMMs and SAHMMs should be set up and tested in identical appplications to compare the difference of the two types in term of recognition rate, memory requirement and speed.

According to our experiment, noncausal systems yield more satisfactory results in the degraded environment, which means they are more suitable for real world application. The major problem for noncausal system is that every node has more than one neighbors and there must be some conflicts among the information derived from different neighbors. Here we tried several methods, such as asynchronous method or multiplication of the membership of every hidden states, which have not been mathematically proven yet. Iteration is utilized in the asychronous method. How to obtain the optimal iteration time at the stage of the evalution of SAHMM is another challenging topic. The 'optimal' number of hidden states M and number of observation N are unsloved problems for the conventional HMM and the proposed SAHMM.

In this thesis we address iterative training method for the proposed SAHMM. Further study will be carried out to prove the rationality of such local miximum search strategies. Genetic Algorithm[89] and other optimization methods will be tried at the training stage of SAHMM. The proposed 1-D and 2-D SAHMMs have obtained promising results in the application of OCR, expanding its application to bioinformatics, machine vision is also the major highlights of future study.

References

[1] J. Serra, "Image analysis and mathematical morphology" London, Academic Press, 1982.

[2] R. M. Haralick, S. R. Sternberg, X. Zhuang, "Image analysis using mathematical morphology," IEEE Trans. Pattern Anal. Machine Intell. PAMI-9, pp. 532-550, 1987.

[3] J. M. S. Prewitt and M. L. Mendelsohn, "The analysis of cell images," in Ann. New York Acad. Sci., Vol. 128, pp. 1035-1053, 1966.

[4] W. Doyle, "Operation useful for similarity-invariant pattern recognition," J. Assoc. Comput. Mach, Vol. 9, pp. 259-267, 1962.

[5] N. Otsu, "A threshold selection method from gray-level histogram," IEEE Trans. Systems Man Cybernet., Vol.9, pp. 62-66, 1979.

[6] Andreas E. Savakis, "Adaptive document image thresholding using foreground and background cluster," Proceedings of International Conference on Image Processing ICIP98, pp. 438-441, 1998.

[7] J. Kittler. And J. Illingworth, "On Threshold Selection Using Illustering Criteria," IEEE SMC, Vol.15, pp. 652-655, 1985.

[8] J. Kittler, J. Illingworth, "Minimum Error Thresholding," Pattern Recognition, Vol. 19, pp. 41-47, 1986.

[9] L. Hertz, R. W. Schafer, "Multilevel Thresholding Using Edge Matching," CVGIP, Vol. 44, pp. 279-295, 1988.

[10] A. Pikaz, and A. Averbuch, "Digital Image Thresholding Based on Topological Stable State," Pattern Recognition, Vol. 29, pp. 829-843, 1996.

[11] L. O'Gorman, "Binarization and Multithresholding of Document Images Using Connectivity," CVGIP, Vol. 56, pp. 496-506, 1994.

[12] A.D. Brink, "Grey-Level Thresholding of Images Using a Correlation Criterion," Pattern Recognition Letters, Vol. 9, pp.335-341, 1989.

[13] F. Morii, "An Image Thresholding Method Using a Minimum Weighted Squared-Distortion Criterion," Pattern Recognition, Vol. 28, pp. 1063-1071, 1995.

[14] L.K. Huang, M.J.J. Wang, "Image Thresholding by Minimizing the Measures of Fuzziness," Pattern Recognition, Vol. 28, pp. 41-51, 1995.

[15] H. Yan and J. Wu, "Character and line extraction from color map images using a multi-layer neural network," Pattern Recognition Letters, Vol. 15 pp. 97-103, 1994.

[16] A Fletcher, K. Kasturi, "A robust algorithm for text string separation from mixed text /graphics images," IEEE Trans, Pattern Anal, Machine Intell, PAMI-10, pp. 910918, 1988.

[17] A. K. Jain, S. K. Bhattacharjee, "Text segmentation using Gabor filters for automatic document processing," Machine Vision Appl. Vol. 5, pp. 169184, 1992.

[18] A.K. Jain and B. Yu, "Document Representation and Its Application to Page Decomposition," IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 20, no. 3, pp. 294-308, Mar. 1998.

[19] L. O'Gorman, "The Document Spectrum for Page Layout Analysis," IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 15, no. 11, pp. 1162-1173, Nov. 1993.

[20] G. Nagy, S. Seth, and S. Stoddard, "Document Analysis with an Expert System," Pattern Recognition in Practice II, pp. 149-155, Elsevier Science, 1984

[21] D. Sylwester and S. Seth, "Adaptive Segmentation of Document Images," Proc. Intl Conf. Document Analysis and Recognition, pp. 827-831, 2001.

[22] H.S. Baird, S.E. Jones, and S.J. Fortune, "Image Segmentation by Shape-Directed Covers," Proc. Intl Conf. Pattern Recognition, pp. 820-825, 1990.

[23] T. Pavlidis and J. Zhou, "Page Segmentation and Classification," CVGIP, Vol. 54, no. 6, pp. 484-496, 1992.

[24] R.M. Haralick, "Document Image Understanding: Geometric and Logical Layout," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 385-390, 1994.

[25] R.G. Casey, E. Lecolinet, "A survey of methods and strategies in character segmentation," Vol. 18, Issue 7, pp. 690 - 706, 1996.

[26] A.K. Jain, R.P.W. Duin and J. Mao, "Statistical pattern recognition: a review," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, no. 1, pp. 4-37, 2000.

[27] P. D. Gader, B. Forester, M. Ganzberger, A. Billies, B. Mitchell, M. Whalen, T. Youcum, "Recognition of handwritten digits using template and model matching," Pattern Recognition, Vol. 5, no. 24, pp. 421-431, 1991.

[28] G. Dimauro, S. Impedovo, G. Pirlo, A. Salzo, "Automatic bankcheck processing: A new engineered system." In S.Impedovo et al, editor, International Journal of Pattern Recognition and Artificial Intelligence, World Scientific, pp. 467-503, 1997.

[29] S. L. Xie, M. Suk, "On machine recognition of hand-printed chinese character by feature relaxation," Pattern Recognition, Vol. 21, no. 1, pp. 1-7, 1988.

[30] D. Guillevic, C. Y. Suen, "Cursive script recognition applied to the processing of bank cheques," In Proc. of 3th International Conference on Document Analysis and Recognition, Montreal-Canada, August, pp. 11-14, 1995.

[31] L. Mico, J. Oncina, "Comparison of fast nearest neighbour classifier for handwritten character recogniton," Pattern Recognition Letters, Vol. 19, pp. 351-356, 1999.

[32] R. O. Duda, P. E. Hart, D. G. Stork, "Pattern Classification," John Wiley and Sons, second edition edition, 2001.

[33] J. Schurmann, "Pattern Classification - A unified view of statistical and neural approaches," Wiley interscience, 1996.

[34] V. Vapnik, "The Nature of Statistical Learning Theory," Springer-Verlag, New York- USA, 1995.

[35] N. E. Ayat, M. Cheriet, C. Y. Suen, "Optimization of the SVM kernels using an empirical error minimization scheme," In Proc. of the International Workshop on Pattern Recognition with Support Vector Machine, Niagara Falls-Canada, August, pp. 354-369, 2002.

[36] H. Byun, S. W. Lee, "Applications of support vector machines for pattern recognition," In Proc. of the International Workshop on Pattern Recognition with Support Vector Machine, Niagara Falls-Canada, August, pp. 213-236, 2002.

[37] L. S. Oliveira, R. Sabourin, "Support Vector Machines for Handwritten Numerical String Recognition," 9th International Workshop on Frontiers in Handwriting Recognition, October 26-29, Kokubunji, Tokyo, Japan, pp 39-44, 2004.

[38] M. Shridhar and A. Badreldin, "Recognition of isolated and simply connected handwritten numerals," Pattern Recognition, Vol. 19, no. 1, pp: 1-12, 1986.

[39] H. Y. Kim and J. h. Kim, "Handwritten korean character recognition based on hierarchical randon graph modeling," In Proc. 6th International Workshop on Frontiers of Handwriting Recognition, Taegon-Korea, August, pp. 557-586, 1998.

[40] M. Bishop, "Neural Networks for Pattern Recognition," Oxford Univ. Press, Oxford- U.K. 1995.

[41] Y. LeCun, L. Bottou, G. B. Orr, K. R. Muller, Eficient backprop. In G. Orr and K. Miller, editors, "Neural Networks: Tricks of the Trade," Springer, 1998.

[42] G. P. Zhang, "Neural networks for classification: a survey," IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews, Vol. 30, no. 4, pp. 451-462, 2000.

[43] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner. "Gradient-based learning applied to document recognition," Proc. of IEEE, Vol. 86, no. 11, pp. 2278-2324, 1998.

[44] M. Zhang. H. Fu. Yan, and M. A. Fabri. "Handwritten digit recognition by adaptative subspace self organizing map," IEEE Trans. on Neural Networks, Vol. 10, pp. 939-945, 1999.

[45] O. Matan, J. C. Burges, Y. LeCun, J. S. Denker, "Multi-digit recognition using a space displacement neural network," In J. E. Moody, S. J. Hanson, and 165 R. L. Lippmann, editors, Advances in Neural Information Processing Systems, Vol. 4, Morgan Kaufmann, pp. 488-495, 1992.

[46] Lethelier, M. Leroux, M. Gilloux, "An automatic reading system for handwritten numeral amounts on french checks," In Proc. 3th International Conference on Document Analysis and Recognition, Montreal-Canada, August, pp. 92-97, 1995.

[47] Ling, M. Lizaraga, N. Gomes, A. Koerich. "A prototype for brazilian bankcheck recognition," In S.Impedovo et al, editor, International Journal of Pattern Recognition and Artificial Intelligence,World Scientific, pp. 549-569. 1997.

[48] Q. Xu. "Automatic Segmentation and Recognition System for Handwritten Dates on Cheques," PhD thesis, Concordia University, Montreal-Canada, December, 2002.

[49] Koch, T. Paquet, L. Heutte, "Combination of Contextual Information for Handwritten Word Recognition," 9th International Workshop on Frontiers in Handwriting Recognition, October 26-29, Kokubunji, Tokyo, Japan, pp 468-473, 2004.

[50] L. Koerich, R. Sabourin, C. Y. Suen, "Fast TwoLevel HMM Decoding Algorithm for Large Vocabulary Handwriting Recognition," 9th International Workshop on Frontiers in Handwriting Recognition, October 26-29, Kokubunji, Tokyo, Japan, pp 232-238, 2004.

[51] I. Yamato, I.Ohya, and K. Ishii, "Recognizing Human Action in Time-Sequential Images Using Hidden Markov Model," in CVPR92, pp. 379-385, 1992.

[52] Y. Deng and W. Byrne, "Hmm word and phrase alignment for statistical machine translation," in Proc. of HLT-EMNLP, 2005.

[53] C.H. Wu, S. Zhao, H.L. Chen, "A protein class database organized with ProSite, protein groups and PIR, superfamilies," J.Comput. Biol., Vol. 3, pp. 547-561, 1996.

[54] L. Bahl, P. Brown, P. de Souza, and R. Mercer, "Maximummutual information estimation of hiddenmarkov model parameters for speech recognition," in Proc. of of ICASSP, Tokyo, pp. 4952, 1986.

[55] J-L. Gauvain and C-H. Lee, "Map estimation of continuous density hmm: Theory and applications," in Proc. of DARPA Speech& Nat. Lang. Processing, Feb. 1992.

[56] L. Saul and M. Rahim, "Maximum likelihood and minimum classification error rate factor analysis for automatic speech recognition," IEEE Trans. on Speech and Audio Processing, Vol. 8, no. 2, pp. 115125, 2000.

[57] A. Stolcke and S. Omuhundro, "Hidden markov model induction by bayesian model merging," in Advances in Neural Information Processing 5, S. Hanson, J. Cowan, and C. Giles, Eds., pp. 1118. Morgan Kaufmann, 1992.

[58] T. Brants, "Estimating markov model structures," Proc. of ICLSP Philadelphia, PA, 1996.

[59] A. Biem, "A model selection criterion for classification: Application to hmm topology optimization," in Proc. 17th ICDAR, Edinburgh, U.K, pp. 104108, 2003.

[60] Hee-Seon Park and Seong-Whan Lee, "A truly 2-D Hidden Markov Model For Off-Line Handwritten Character Recognition," Pattern Recognition, Vol. 31, No. 12, pp. 1849-1864, Dec. 1998

[61] Oscar E. Agazzi Shyh-shiaw Kuo, Esther Levin, and Roperto Pieraccini, "Connected and Degraded Text Recognition Using Planar Hidden Markov Models," Proc. IEEE Intl Conf. Acoustics, Speech, and Signal Processing (ICASSP), Vol. V, pp. 113-116, 1993.

[62] Zoubin Ghahramani and Michael I. Jordan, "Factorial hidden Markov models," In David S. Touretzky, Michael C. Mozer, and Michael E. Hasselmo, editors, Proceedings of Advances in Neural Information Pro- cessing Systems, NIPS, Vol. 8, pages 472-478. MIT Press, 1995

[63] M. Sezgin and B. Sankur, "Survey over Image Thresholding Techniques and Quantitative Performance Evaluation," Journal of Electronic Imaging, Vol. 13, No. 1, pp. 146-165, 2004.

[64] S. U. Le, S. Y. Chung, and R. H. Park, "A comparative performance study of several global thresholding techniques for segmentation," Graph. Models Image Process. Vol. 52, pp. 171-190, 1990.

[65] J. S. Weszka and A. Rosenfeld, "Threshold evaluation techniques," IEEE Trans. Syst. Man Cybern. SMC-8, pp. 627629, 1978.

[66] P. W. Palumbo, P. Swaminathan, and S. N. Srihari, "Document image binarization: Evaluation of algorithms," Proc. SPIE 697, pp. 278286, 1986.

[67] P. K. Sahoo, S. Soltani, A. K. C. Wong, and Y. Chen, "A survey of thresholding techniques," Comput. Graph. Image Process.41, pp. 233 260, 1988.

[68] C. A. Glasbey, "An analysis of histogram-based thresholding algorithms," Graph. Models Image Process. Vol. 55, 532537, 1993.

[69] O. D. Trier and A. K. Jain, "Goal-directed evaluation of binarization methods," IEEE Trans. Pattern Anal. Mach. Intell. PAMI-17, 11911201, 1995.

[70] B. Sankur, A.T. Abak, U. Baris, "Assessment of thresholding algorithms for document processing," in: ICIP99, pp. 580-584, 1999.

[71] J.R. Parker, "Gray level thresholding in badly illuminated images," IEEE Trans. Pattern Anal. Mach. Intell. Vol. 13, no. 8, pp. 813-819, 1991.

[72] E. Giuliano, O. Paitra, L. Stringa, "Electronic character reading system," U. S. Patent 4,047,15,6 September, 1977.

[73] J.M. White, G.D. Rohrer, "Imager segmentation for optical character recognition and other applications requiring character image extraction," IBM J. Res. Dev. Vol. 27, No. 4, pp. 400-411, 1983.

[74] A. Antonacopoulos, "Page Segmentation Using the Description of the Background," Computer Vision and Image Understanding, Special Issue on Document Analysis and Retrieval, Vol. 70, No. 3, pp. 350-369, June 1998.

[75] Rejean Plamondon and Sargur N. Srihari, "On-line and off-line handwriting Recognition: A comprehensive survey," IEEE Transactions on pattern analysis and machine intelligence, Vol. 22, No. 1, pp. 63-84, Jan, 2000.

[76] Jianjiang Feng, Anni Cai, "Fingerprint Representation and Matching in Ridge Coordinate System," Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06), Vol. 04, no. 20-24, pp. 485-488, Aug, 2006.

[77] Cheng-Lin Liu, Kazuki Nakashima, Hiroshi Sako, Hiromichi Fujisawa, "Handwritten digit recognition: benchmarking of state-of-the-art techniques," Pattern Recognition, Vol. 36, pp. 22712285, 2003.

[78] A. L. Koerich, R. Sabourin, C. Y. Suen, "Large vocabulary off-line handwriting recognition: A survey," Pattern Analysis and Applications, Vol. 6, No. 2, pp. 97-121, 2003.

[79] L. R. Rabiner. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proceedings of the IEEE, Vol. 77, No. 2, pp. 257-286, 1989.

[80] El Yacoubi, M. Gilloux, R. Sabourin, C. Y. Suen, "An hmm-based approach for offline unconstrained handwritten word modeling and recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21, No. 8, pp. 752-760, 1999.

[81] Kundu, Y. He, M. Chen, "Alternatives to variable duration hmm in handwriting recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 11, pp.1275-1280, 2002.

[82] W. Senior, A. J. Robinson, "An off-line cursive handwriting recognition system," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 3, pp. 309-321, 2002.

[83] J. Cai, Z. Liu. "Integration of Structural and Statistical Information for Unconstrained Handwritten Numeral Recognition," Proceedings of the Fourteenth International Conference on Pattern Recognition, Vol. I, pp. 378-380, 1998.

[84] S. Britto Jr., R. Sabourin, F. Bortolozzi, C. Y. Suen, "Foreground and Background Information in an HMM-Based Method for Recognition of Isolated Characters and Numeral Strings," 9th International Workshop on Frontiers in Handwriting Recognition, October 26- 29, 2004, Kokubunji, Tokyo, Japan, pp. 371-376, 2004.

[85] Zoubin Ghahramani and Michael I. Jordan, "Factorial hidden Markov models," Proceedings of Advances in Neural Information Processing Systems, NIPS, Vol. 8, pp. 472, 1995.

[86] F. J. Pineda, "Generalization of back-propagation to recurrent neural networks," Physical Review Letters, Vol.59, pp. 2229-2232, 1987.

[87] Y. Bengio, P. Frasconi, "Input-output HMM's for sequence processing," IEEE Trans. on Neural Networks, Vol. 7, pp. 12311249, 1996.

[88] S. E. Levinson, L. R. Rabiner and M. M. Sondhi, "An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition," The Bell System Technical Journal, Vol. 62, No. 4, pp. 1035-1074, 1983.

[89] C.W. Chau, S. Kwong, C.K. Diu, W.R. Fahrner, Optimization of HMM by a genetic algorithm, Acoustics, Speech, and Signal Processing, ICASSP-97. Volume 3, pp. 1727-1730, April 1997.

[90] O. D. Trier, A. K. Jain, and T. Taxt, "Feature Extraction Methods for Character Recognition: A Survey," Pattern Recognition, Vol. 29, pp. 641-662, 1996.

[91] Cheng-Lin Liu, Masashi Koga, Hiromichi Fujisawa, "Gabor Feature Extraction for Character Recognition: Comparison with Gradient Feature," Proceedings of International Conference on Document Analysis and Recognition, pp. 121-125, 2005.

[92] T. Kohonen. "Self-Organization and Associative Memory, 3rd ed," Springer, Berlin, 1993.

[93] L.E. Baum, "An inequality and associated maximization technique in statistical estimation for probabilistic functions of finite state Markov chains," in Inequalities III. New York: Academic, pp. 18 1972.

[94] L. E. Baum and T. Petrie,"Statistical inference for probabilistic functions of finite state Markov chains," Ann. Math. Stat., Vol. 37, pp. 15541563, 1966.

[95] L. E. Baum and J. A. Eagon, "An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology," Bull. Amer. Math. Stat., Vol. 37, pp. 360363, 1967.

[96] L.E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," Ann. Math. Stat., Vol. 41, No. 1, pp. 164171, 1970.

[97] A.P. Dempster, N.M. Laird, and D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. Roy. Stt. Soc., Vol. 39, No. 1, pp. 1-36, 1977.

[98] R. Cole, L. Hirschman, L. Atlas, and M. Beckman et al., "The challenge of spoken language systems: Research directions for the nineties," IEEE Trans. Speech Audio Processing, Vol. 3, pp. 121, Jan. 1995.

[99] H.J. Kim, K.H. Kim, S.K. Kim and J.K. Lee, Online Recognition of Handwritten Chinese Characters based on Hidden Markov Models, Pattern Recognition, Vol. 30, No. 9, pp. 1489-1500, 1997.

[100] H.-S. Park and S.-W. Lee, "A truly 2-D hidden Markov model for offline handwritten character recognition," Patt. Recogn. Vol. 31, No. 12, pp. 1849-1864, 1998.

[101] Jia Li, Amir Najmi, and Robert M. Gray, "Image Classification by a Two Dimensional Hidden Markov Model," IEEE Transactions on Signal Processing," Vol. 48, No. 2, pp. 517-33, February 2000.

[102] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, 1. University of California Press, Berkeley, CA, pp. 281-297, 1967.

[103] M. Sezgin and B. Sankur, "Survey over Image Thresholding Techniques and Quantitative Performance Evaluation," journal of Electronic Imaging, Vol. 13, No. 1, pp. 146-165, 2004.

[104] S. U. Le, S. Y. Chung, and R. H. Park, "A comparative performance study of several global thresholding techniques for segmentation," Graph. Models Image Process. Nol. 52, pp. 171190, 1990.

[105] J. M. S. Prewitt and M. L. Mendelsohn, "The analysis of cell images," in Ann. New York Acad. Sci., Vol. 128, pp. 1035-1053, 1966.

[106] W. Doyle, "Operation useful for similarity-invariant pattern recognition," J. Assoc. Comput. Mach, Vol. 9, pp. 259-267, 1962.

[107] N. Otsu, "A threshold selection method from gray-level histogram," IEEE Trans. Systems Man Cybernet., Vol.9,pp. 62-66, 1979.

[108] Andreas E. Savakis, .Adaptive document image thresholding using foreground and background cluster The 7th International Conference on Electronics, Information, and Communications (ICEIC04), pp. 438-441, Proceedings of International Conference on Image Processing ICIP98, 1998

[109] J. Kittler And J. Illingworth, "On Threshold Selection Using Illustering Criteria," IEEE SMC, Vol.15, No. 29, pp. 652- 655, 1985.

[110] J. Kittler., J. Illingworth, "Minimum Error Thresholding," Pattern Recognition, Vol. 19, pp. 41-47, 1986

[111] L. Hertz, R. W. Schafer, "Multilevel Thresholding Using Edge Matching," CVGIP, Vol. 44, pp. 279-295, 1988.

[112] A. Pikaz, and A. Averbuch, "Digital Image Thresholding Based on Topological Stable State," Pattern Recognition, Vol. 29, pp. 829-843, 1996.

[113] , L. O'Gorman, "Binarization and Multithresholding of Document Images Using Connectivity," CVGIP, Vol. 56, pp. 496-506, 1994.

[114] A.D. Brink, "Grey-Level Thresholding of Images Using a Correlation Criterion," Pattern Recognition Letters, Vol. 9, pp. 335-341, 1989.

[115] F. Morii, "An Image Thresholding Method Using a Minimum Weighted Squared-Distortion Criterion," Pattern Recognition,Vol. 28, pp. 1063-1071, 1995.

[116] L.K. Huang, M.J.J. Wang, "Image Thresholding by Minimizing the Measures of Fuzziness," Pattern Recognition, Vol. 28, pp. 41-51, 1995.

[117] P. W. Palumbo, P. Swaminathan, and S. N. Srihari, "Document image binarization: Evaluation of algorithms," Proc. SPIE 697, pp. 278286, 1986.

[118] P. K. Sahoo, S. Soltani, A. K. C. Wong, and Y. Chen,"A survey of thresholding techniques," Comput. Graph. Image Process. Vol. 41, pp. 233 260, 1988.

[119] C. A. Glasbey, "An analysis of histogram-based thresholding algorithms," Graph. Models Image Process. 55, 532537, 1993.

[120] O. D. Trier and A. K. Jain, "Goal-directed evaluation of binarization methods," IEEE Trans. Pattern Anal. Mach. Intell. PAMI, Vol. 17, pp. 1191 1201, 1995.

[121] B. Sankur, A.T. Abak, U. Baris, "Assessment of thresholding algorithms for document processing," in: ICIP99, pp. 580-584, 1999.

[122] J.R. Parker, "Gray level thresholding in badly illuminated images," IEEE Trans. Pattern Anal. Mach. Intell. Vol. 13, No. 8, pp. 813-819 (1991).

[123] E. Giuliano, O. Paitra, L. Stringa, "Electronic character reading system," U. S. Patent 4, 047,15, 6 September, 1977.

[124] J.M. White, G.D. Rohrer, "Imager segmentation for optical character recognition and other applications requiring character image extraction," IBM J. Res. Dev. Vol. 27, No. 4, pp. 400-411, 1983.

[125] A. Antonacopoulos, "Page Segmentation Using the Description of the Background," Computer Vision and Image Understanding, Special Issue on Document Analysis and Retrieval, Vol.70, No.3, pp.350-369, June 1998.

[126] T. Pun, "A New Method for Gray-Level Picture Threshold Using the Entropy of the histogram," EUFL4SIP:Signal Processing, Vol. 2, No. 3, pp. 223-237, July 1980.

[127] T. Pun, "Entropic Thresholding, A New Approach," Computer Graphics and Image Processing, Vol. 16, pp. 210-239, 1981.

[128] P. Sahoo, C. Wilkins, and J. Yeager, "Threshold Selection Using Renyis Entropy," Pattern Reconition, Vol. 30, pp. 71-84, 1997.

[129] N.R. Pal, and S.K. Pal, "Entropic Thresholding," EURASIP: Signal Processing, Vol. 16, No. 2, pp. 97-108, 1989.

[130] A. D. Brink, and N.E. Pendock, "Minimum Cross Entropy Threshold Selection," Pattern Recognition, Vol. 29, pp. 179-188, 1996.

[131] J.N. Kapur, P.K. Sahoo, and A.K.C. Wong, "A New Method for Gray-Level Picture Thresholding Using the Entropy of the Histogram," Graphical Models and Image Processing, Vol. 29, pp. 273-285, 1985.

[132] A.G. Shanbag, "Utilization of Information Measure as a Means of Image Thresholding," CVGIP, Vol. 56, pp. 414-419, 1994.

[133] A. Beghdadi, A.L.P. Negrate, and V. DeLesegno, "Entropic Thresholding Using A Block Source Model," CVGIP CMIP. Vol. 57, pp. 197-205, 1995.

[134] C.H. Li, C.K. Lee, "Minimum Cross-Entropy Thresholding," Pattern Recognition, Vol. 26, pp. 617-625, 1993.

[135] M.I. Sezan, "A Peak Detection Algorithm and its Application toHistogram-Based Image Data Reduction," CVGIP. Vol. 29, pp. 47-59, 1985.

[136] S.C. Sahasrabudhe and K.S.D. Gupta, "A Valley-seeking Threshold Selection Technique," Computer Vision and Image Processing, (A. Rosenfeld, L.Shapiro eds.), pp. 55-65, Academic Press, 1992.

[137] A. Rosenfeld, P. De laTorre, "Histogram Concavity Analysis as an Aid in Threshold Selection," IEEE SMC, Vol. 13, pp. 231-235, 1983.

[138] M.J. Carlotto, "Histogram Analysis Using a Scale-Space Approach," IEEE PAMI, Vol. 9, pp. 121-129, 1987.

[139] J.C. Ohio, "Automatic Threshold Selection Using the Wavelet Transform",CVGIP, Vol. 56, pp. 205-218, 1994.

[140] J. Weszka, A. Rosenfeld,"Histogram Modification for Threshold Selection," IEEESMC, Vol. 9, pp. 38-52, 1979.

[141] G. Sapiro, V.Casselles, "Histogram Modification via Partial Differential Equations," Proc. of ICIP 95. pp. 632-635, 1995

[142] L.A. Fletcher and R. Kasturi, "A robust algorithm for text string separation from mixed textigraphics images", ZEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 10, No. 6, pp. 910-918, 1988.

[143] Y. Yang and H. Yan, "An adaptive logical method for binarization of degraded document images," Pattern Recogn. Vol. 33, pp. 787807, 2000.

[144] T. W. Ridler and S. Calvard, "Picture thresholding using an iterative selection method," IEEE Trans. Syst. Man Cybern.SMC, Vol. 8, pp. 630632, 1978.

[145] C. K. Leung and F. K. Lam, "Performance analysis of a class of iterative image thresholding algorithms," Pattern Recogn. Vol. 29, No. 9, pp. 15231530, 1996.

[146] H. J. Trussel,"Comments on picture thresholding using iterative selection method," IEEE Trans. Syst. Man Cybern. SMC-9, pp. 311, 1979.

[147] M. K. Yanni and E. Horne, "A new approach to dynamic thresholding," EUSIPCO94: 9th European Conf. Sig. Process. Vol. 1, pp. 3444, 1994.

[148] D. E. Lloyd, "Automatic target classification using moment invariant of image shapes," Technical Report, RAE IDN AW126,Farnborough, UK, Dec, 1985.

[149] S. Cho, R. Haralick, and S. Yi, "Improvement of Kittler and Illingworthss minimum error thresholding," Pattern Recogn. Vol. 22, pp. 609617, 1989.

[150] C.V. Jawahar, P.K. Biswas, and A.K. Ray, "Investigations on fuzzy thresholding based on fuzzy clustering," Pattern Recogn. Vol. 30, No. 10, pp. 16051613, 1997.

[151] K. Pal and A. Rosenfeld, "Image enhancement and thresholding by optimization of fuzzy compactness," Pattern Recogn. Lett. Vol. 7, pp. 7786, 1988.

[152] A. Rosenfeld, "The fuzzy geometry of image subsets," Pattern Recogn. Lett. 2, pp. 311317, 1984.

[153] W. H. Tsai, "Moment-preserving thresholding: A new approach," Graph. Models Image Process. Vol. 19, pp. 377393, 1985.

[154] S. C. Cheng and W. H. Tsai, "A neural network approach of the moment-preserving technique and its application to thresholding," IEEE Trans. Comput. Vol. 42, pp. 501507, 1993.

[155] E. J. Delp and O. R. Mitchell, "Moment-preserving quantization," IEEE Trans. Commun. Vol. 39, pp. 15491558, 1991.

[156] Rishi R. Rakesh, Probal Chaudhuri, and C. A. Murthy, "Thresholding in Edge Detection:A Statistical Approach," IEEE Transaction on Image processing, Vol. 13, No. 7, JULY, 2004.

[157] C. A. Murthy and S. K. Pal, "Fuzzy thresholding: A mathematical framework, bound functions and weighted moving average technique," Pattern Recogn. Vol. 11, pp. 197206, 1990.

[158] R. Yager, "On the measure of fuzziness and negation. Part I: Membership in the unit interval," Int. J. Gen. System. Vol. 5, pp. 221229, 1979.

[159] K. Ramar, S. Arunigam, S. N. Sivanandam, L. Ganesan, and D. Manimegalai, "Quantitative fuzzy measures for threshold selection," Pattern Recogn Letter, Vol. 21, pp. 17, 2000.

[160] X. Fernandez, "Implicit model oriented optimal thresholding using Kolmogorov-Smirnov similarity measure," ICPR 2000: Intl. Conf. Patt. Recog., pp. 466469, Barcelona, 2000.

[161] C. K. Leung and F. K. Lam, "Maximum segmented image information thresholding," Graph. Models Image Process. Vol. 60, pp. 5776, 1998.

[162] S. D. Yanowitz and A. M. Bruckstein, "A new method for image segmentation," Comput. Graph. Image Process. Vol. 46, pp. 8295, 1989.

[163] D. Shen and H. H. S. Ip, "A Hopfield neural network for adaptive image segmentation: An active surface paradigm," Pattern Recogn. Lett. Vol. 18, pp. 3748, 1997.

[164] H. Yan and J. Wu, "Character and line extraction from color map images using a multi-layer neural network," Pattern Recognition Letters, Vol. 15, pp. 97-103, 1994.

[165] MacQueen, J.B. Some methods for classification and analysis of multivariate observations. In: Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, 1. University of California Press, Berkeley, CA, pp. 281-297, 1967.

[166] W. Niblack, An Introduction to Image Processing, pp. 115-116, Prentice-Hall, Englewood Cliffs, NJ 1986.

[167] J. Sauvola and M. Pietaksinen, Adaptive document image binarization, Pattern Recogn. Vol. 33, pp. 225-236, 2000.

[168] W. A. Yasnoff, J. K. Mui, and J. W. Bacus, "Error measures for scene segmentation," Pattern Recogn. Vol. 9, pp. 217231, 1977.

[169] M. D. Levine and A. M. Nazif, "Dynamic measurement of computer generated image segmentations," IEEE Trans. Pattern Anal. Mach. Intell. PAMI Vol. 7, pp. 155164, 1985.

[170] Y. J. Zhang, "A survey on evaluation methods for image segmentation," Pattern Recogn. Vol. 29, pp. 13351346, 1996.

[171] S.M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "ICDAR 2003 Robust Reading Competitions," Proc.of the ICDAR, pp. 682-687, 2003.

[172] R.E. Crochiere and L. R. Rabiner. "Multirate Digital Signal Process," inf. Prentie-Hall, 1983.

[173] J. T. Favata, G. Srikantan, and S. N. Srihari, "Handprinted character/digit recognition using a multiple feature/resolution philosophy," In International Workshop on the Frontiers of Handwriting Recognition, IWFHR-4, 1994.

[174] Nafiz Arica, Student Member, IEEE, and Fatos T. Yarman-Vural, Senior Member, IEEE, "Optical Character Recognition for Cursive Handwriting," IEEE Transaction on Pattern analysis and machine intelligence, Vol. 24, No. 6, JUNE, 2002.

[175] O. Trier, A. K. Jain, and T. Taxt. "Feature extraction methods for character recognition - A Survey," Pattern Recognition, Vol. 29, No. 4, pp. 641662, 1996.

[176] X. Wang, X. Ding, C. Liu, "Optimized Gabor filter based feature extraction for character recognition," Proc. 16th ICPR, Quebec, Canada, Vol.4, pp. 223-226, 2002.

[177] D.H.Ballard and C.M. Brown, "Computer Vision," pp.65-70 Englewood Cliffs, New Jersey: Prentice Hall, 1982.

[178] W.K.Pratt, "Digital Image Processing," New York:John Wiley & Sons,second ed., 1991.

[179] A.D. Bimbo, S. Santini, and J. Sanz, "OCR from poor quality images by deformation of elastic template," in Proceedings of 12th IAPR Int. Conf. Pattern Recognition, Vol. 2, pp. 433-435, 1994.

[180] T. H. Reiss, "The revised fundamental theorem of moment invariants," IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 13, pp. 830-834, Aug. 1991.

[181] M.K. Hu, "Visual pattern recognition by moment invariant," IRE Trans. Information Theory, Vol. 8, pp. 179-187, Feb. 1962.

[182] A. Khotanzad and Y.H. Hong, "Rotation invariant image recognition using feature selected via a systematic method," Pattern recognition, Vol. 23, No. 10, pp. 1089-1101,1990.

[183] H. Feng and M. Effros, "Separable KarhunenLoeve transforms for the weighted universal transform coding algorithm," in Proc. IEEE Int. Conf, Acoustics, Speech, and Signal Processing, Phoenix, AZ, Mar. 1999.

[184] M. H. Glauberman, "Character recognition for business machines," Electronics, Vol. 29, pp. 132-136. Feb. 1956.

[185] F. Kimura and M. Shridhar, "Handwritten numerical recognition based on multiple algorithms," Pattern Recognition, Vol. 24, no. 10, pp. 969-983, 1991.

[186] I. Sekita, K. Toraichi, R. Mori, K. Yamamoto, and H. Yamada, " Feature extraction of handwritten Japanese character by spine functions for relaxation matching," Pattern Recognition, no. 1, pp. 9-17, 1988.

[187] F.P. Kuhl and C.R. Giardina, " Elliptic Fourier features of a closed contour," Computer Vision, Graphics and Image Processing, Vol. 18, pp. 236-258, 1982.

[188] C. S. Lin and C. L. Hwang, "New forms of shape invariants from elliptic Fourier descriptors," Pattern Recognition, Vol. 20, no. 5, pp. 535-545, 1987.

[189] Hisham H. A. Othman, "A Novel Reduced-Complexity Approach to Hidden Markov Modeling of 2-D Processes with Application to Face Recognition," thesis, 2002.

[190] Ferdinando Silvestro Samaria, "Face Recognition Using Hidden Markov Model," A dissertation, Ph. D., University of Cambridge, 1994.

[191] Oscar E. Agazzi Shyh-shiaw Kuo, Esther Levin, and Roperto Pieraccini, "Connected and Degraded Text Recognition Using Planar Hidden Markov Models," Proc. IEEE Intl Conf. Acoustics, Speech, and Signal Processing (ICASSP), Vol. V, pp. 113-116, 1993.

[192] Ara Nefian and Monson H. Hayes III, "An Embedded HMM for Face Detection and Recognition," Proc. IEEE Intl Conf. Acoustics, Speech, and Signal Processing (ICASSP), Vol. VI, pp. 3553-3556, 1999.

[193] Patrice Y. Simard, Dave Steinkraus, John Platt, "Best Practice for Convolutional Neural Networks Applied to Visual Document Analysis," International Conference on Document Analysis and Recogntion (ICDAR), IEEE Computer Society, Los Alamitos, pp. 958-962, 2003.

[194] Karim Abou-Moustafa, Ching Suen, Mohamed Cheriet, "A GENERATIVE-DISCRIMINATIVE HYBRID FOR SEQUENTIAL DATA CLASSIFICATION," The IEEE Intl. Conf. On Acoustics and Signal Processing (ICASSP'04), Montreal, pp. 805-808, May, 2004.

[195] G. A. Carpenter and S. Grossberg, "The ART of Adaptive Pattern Recognition by a Self-Organizing Neural Network," IEEE Computer, Vol. 21, No. 3, pp. 77-88, 1988.

[196] Bunke, M. Roth, and E. G. Schukat-Talamazzini, "Off-line cursive handwriting recognition using hidden markov models," Pattern Recognition, Vol. 28, No. 9, pp. 1399-1413, 1995.

[197] D. Gader, M. A. Mohamed, J. M. Keller, "Fusion of handwritten word classifiers," Pattern Recognition Letters, Vol. 17, No. 6, pp. 577-584, 1996.

[198] S. Knerr and E. Augustin, "A neural network-hidden markov model hybrid for cursive word recognition," In Proc. of 14th International Conference on Pattern Recognition, Vol. 2, Brisbane-Australia, August, pp. 1518-1520, 1998.

[199] D. Guillevic, C. Y. Suen. "Cursive script recognition applied to the processing of bank cheques," In Proc. of 3th International Conference on Document Analysis and Recognition, Montreal-Canada, August, pp. 11-14, 1995.

[200] El Yacoubi, M. Gilloux, R. Sabourin, C. Y. Suen, "An hmm-based approach for offline unconstrained handwritten word modeling and recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21, No. 8, pp. 752-760, 1999.

[201] J. H. Kim, K. K. Kim, C. Y. Suen, "An HMM-MLP hybrid model for cursive script recognition," Pattern Analysis and Applications, Vol. 3, pp. 314-324, 2000.

[202] Freitas, F. Bortolozzi, and R. Sabourin, "Handwritten isolated word recognition: An approach based on mutual information for feature set validation," In Proc. 6th International Conference on Document Analysis and Recognition, Seattle-USA, September, pp. 665-669, 2001.

[203] S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen. "Automatic recognition of handwritten numerical strings: A recognition and verification strategy," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 1, pp. 1438-1454, 2002.

[204] Q. Xu, J. H. Kim, L. Lam, and C. Y. Suen, "Recognition of handwritten month words on bank cheques," In Proc. 8th International Workshop on Frontiers of Handwriting Recognition, Niagara on the Lake-CA, Vol. 169, pp. 111-116, August,2002.

[205] Kundu, Y. He, M. Chen, "Alternatives to variable duration hmm in handwriting recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 11, pp. 1275-1280, 2002.

[206] Arika, F. T. Yarman-Vural, "Optical character recognition for cursive handwriting," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 6, pp. 801-813, 2002.

[207] M. N. Kapp, C. Freitas, R. Sabourin, "Handwritten Brazilian Month Recognition: An Analysis of Two NN Architectures and a Rejection Mechanism," 9th International Workshop on Frontiers in Handwriting Recognition, October 26-29, Kokubunji, Tokyo, Japan, pp. 209-214, 2004.

[208] K. Abend, T. J. Harley and L. N. Kanal, "Classification of binary random patterns," IEEE Trans, Inform, Theory, Vol. 11, pp. 538-544, 1965.

[209] M. P. Ekstrom and J. W. Woods, "Two-dimensional spectral factorization with applications in recuesive digital filtering," IEEE Trans, Acoust. Speech Signal Process, Vol. 24, pp. 115-128, 1976.

[210] L. Koerich, R. Sabourin, C. Y. Suen, "Fast TwoLevel HMM Decoding Algorithm for Large Vocabulary Handwriting Recognition," 9th International Workshop on Frontiers in Handwriting Recognition, October 26-29, Kokubunji, Tokyo, Japan, pp 232-238, 2004.

[211] D. Decoste, B. Scholkopf, "Training invariant support vector machines," Machine Learning 46, pp. 161-190, 2002.

[212] Karim Abou-Moustafa , Ching Suen , Mohamed Cheriet, "A GENERATIVE-DISCRIMINATIVE HYBRID FOR SEQUENTIAL DATA CLASSIFICATION," The IEEE Intl. Conf. On Acoustics and Signal Processing (ICASSP'04), Montreal, pp. 805-808, May, 2004.

[213] Sylvain Chevalier,Edouard Geoffrois, and Francoise Preteux, "A 2D Dynamic Programming Approach for Markov Random Field-based Handwritten Character Recognition," Pattern Recognition, 1996., Proceedings of the 13th International Conference, Vol 2, pp. 320 - 324, Aug. 1996.

VITA AUCTORIS

Songtao Huang was born in Qinglong, Hebei, China, in February, 1973. He received his B.A.Sc. degree in electrical engineering in 1993 from the University of Wuhan in China and his Master of Science degree in electrical engineering in 2003 from the University of Windsor in Canada. He is currently a candidate in the electrical and computer engineering Ph.D. program at the University of Windsor. His research interests include digital signal processing, image processing, pattern recognition.

Accepted papers:

Songtao Huang, Chunhong Chen, Majid Ahmadi, Pedram Mokrian, "An optimal variable voltage scheduling", the 2002 IEEE International Midwest Symposium on Circuits and Systems, Vol. 1, pp. 4-7, 2002.

Songtao Huang, Majid Ahmadi ,William.C.Miller, "An Alternative Search Motion Estimation Algorithm", the 3rd International Symposium on Image and Signal Processing and Analysis, Vol. 2, pp. 844 - 848, 2003.

Songtao Huang, Majid Ahmadi ,William.C.Miller (2003) "A novel hierarchical search Motion Estimation Algorithm", the 46th Midwest symposium on circuit and system, Vol. 2, pp. 564 - 567, 2003.

I. El-Feghi, S. Huang, M.A. Sid-Ahmed, M. Ahmadi, "X-ray Image Segmentation using Auto Adaptive Fuzzy Index Measure", the 47th Midwest Symposium on Circuits and Systems, Vol. 3, pp. 471-474, 2004.

I. El-Feghi, S. Huang, M.A. Sid-Ahmed, M. Ahmadi, Image Processing, "Contrast enhancement of radiograph images based on local heterogeneity measures", International Conference on Image Processing, Vol. 2, pp. 989 - 992, 2004.

Songtao Huang, Majid Ahmadi, M.A. Sid-Ahmed, "A binarization method for scanned documents based on hidden Markov model", International Symposium on Circuits and Systems, May 21-24, 2006.

Songtao Huang, Majid Ahmadi ,M.A. Sid-Ahmed, "An edge based thresholding method", IEEE international conference on systems, Man and Cybernetics in Taiwan, 2006.

Revised journal papers:

Songtao Huang, Majid Ahmadi ,M.A. Sid-Ahmed (2006) "A HMM based character extraction method" to Pattern Recognition.

Sent journal papers

Songtao Huang, Majid Ahmadi ,M.A. Sid-Ahmed (2006) "A Self-Adaptive HMM and its application in Optical Character Recognition" to Pattern Recognition.