PROBABILISTIC RANDOM FIELD BASED METHOD FOR ANNOTATED MACHINE PRINTED DOCUMENTS PREPROCESSING

By

Xujun Peng

29 September 2010

A DISSERTATION SUBMITTED TO THE
FACULTY OF THE GRADUATE SCHOOL OF STATE
UNIVERSITY OF NEW YORK AT BUFFALO
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER AND SCIENCE ENGINEERING

Acknowledgment

I would like to take this opportunity to express my sincere gratitude to my advisor, Dr. Venu Govindaraju, for his advice, guidance and encouragement. He opens the door of my research career. I would like to extend my appreciation to my committee member Dr Peter Scott and Dr Jason Corso for their valuable input throughout the course of my research. I also appreciate the collaboration from Dr Srirangaraj Setlur and Dr Sitaram Ramachandrula. Lastly but not least, I would like to thank my wife Linyan and my family for their always support.

To my be loved families.

Abstract

Today, the convenience of search, both on the personal computer hard disk and on the web, is essentially limited to machine-printed text documents and images because of the poor accuracy of handwriting recognizers. The proposed research will advance the state-of-the-art in realizing search of hand-annotated documents. We will primarily target machine-printed documents which have been annotated by hand by multiple writers in an office/collaborative environment. In applications where the annotations are action instructions (such as, “make 4 copies”, “remove Figure X” etc.) we can envision the proposed system serving as the front end of an OCR-based NLP module. We expect that the techniques developed in this dissertation will be also useful for retrieval of pages from material in languages for which accurate OCRs do not exist.

The main research task proposed is that of segmenting handwritten text, machine printed text, noise or overlapped text, sometimes referred to as the task of ”ink separation”. Prior techniques primarily use histogram thresholding and analysis of the connectivity of strokes. These algorithms, although effective, rely on heuristic rules of spatial constraints, and are not scalable across applications.

We have developed a system that is composed by three parts: the binariztion of document images (focus on hand-held devices captured documents), a boosted tree classifier to perform the initial classification which is followed by a Markov random field (MRF) based approach to re-label the initial segments based on their statistical dependencies within a neighborhood. The MRF based binarization will provide a reliable binarized document image for segmentation even with bad illumination. The boost tree will allow dividing the training data set into several small clusters and use a simple classifier to solve the initial labeling at the cluster (homogeneous) levels. The overlapped text will be further separated using a MRF based method.

The isolated handwritten textual blocks will be indexed (unsupervised) based on writing instrument, style, ink color, etc. as being possible indicators of different writers. We have shown the ability to selectively remove the annotations belonging to a particular writer and allow the end user of the system to view an unmarked document even though the original document image is marked up. This feature will be accomplished by intelligent document restoration whereby the removal of overlapping strokes does not damage the underlying machine-printed text.

We have performed experiments on a large document dataset and report results.

List of Figures

List of Tables

Contents

Chapter 1

Overview

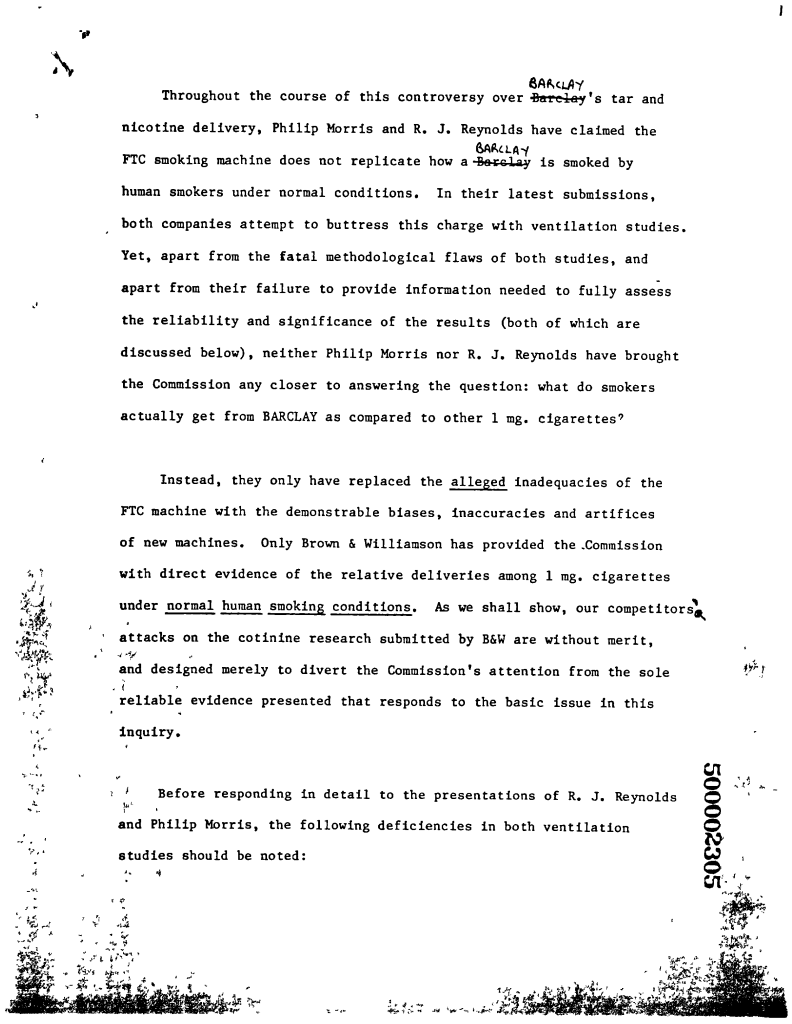
1.1 Text Separation

After decades of research and development, automatic document processing systems have attained considerable success in that large volumes of paper documents can be digitized and processed to recognize textual content and facilitate information retrieval. Unfortunately, the convenience of retrieval and search is limited to clean machine-printed text document images and recognition of free-form handwriting still remains a considerable challenge. In many scenarios, a mixture of machine printed text and handwriting occur within a single document, such as hand-filled medical forms, tax forms, annotated correspondence as shown in Fig. 1 and Fig. 2. Sometimes, it may be of interest to know who signed or edited a document or what was written on a document and retrieve memos or a specific keyword by a particular author. The preprocessing of mixed

documents to isolate handwritten text from machine printed text is a necessary step in the design of such recognition and retrieval systems.

Although research into the location of text blocks and discrimination of machine printed and handwritten text can be traced back to the early work on extraction and recognition of handwritten ZIP codes from mail pieces [2], much of it was focused on identification of clearly separated and clean text. A significant amount of prior research for handwriting identification is based on document layout analysis and zone classification wherein a document is segmented into words, lines, and zones in a bottom-up approach or in a top-down manner [3, 4] and contextual information based on the location of text zones points to the location of handwritten data.

Another thrust has been on using textual elements alone without considering context or layout. In [5], Eduardo et al. suggested two types of features (content-related features and shape-related features) to characterize handwritten text on bank check images. This approach uses a fixed-size frame to extract features, but locating and labeling overlapped text is not part of the model in their work. Jang et al. [6] proposed an approach using geometric features to classify machine printed and handwritten addresses on mail-pieces. Guo and Ma [7] separated handwritten material from documents using a Hidden Markov Model (HMM) by projecting each word horizontally and considering a projected word as a sequential signal. This model is reliable for recognizing machine printed isolated characters but it is difficult to extend the work to recognize annotations and handwritten text because



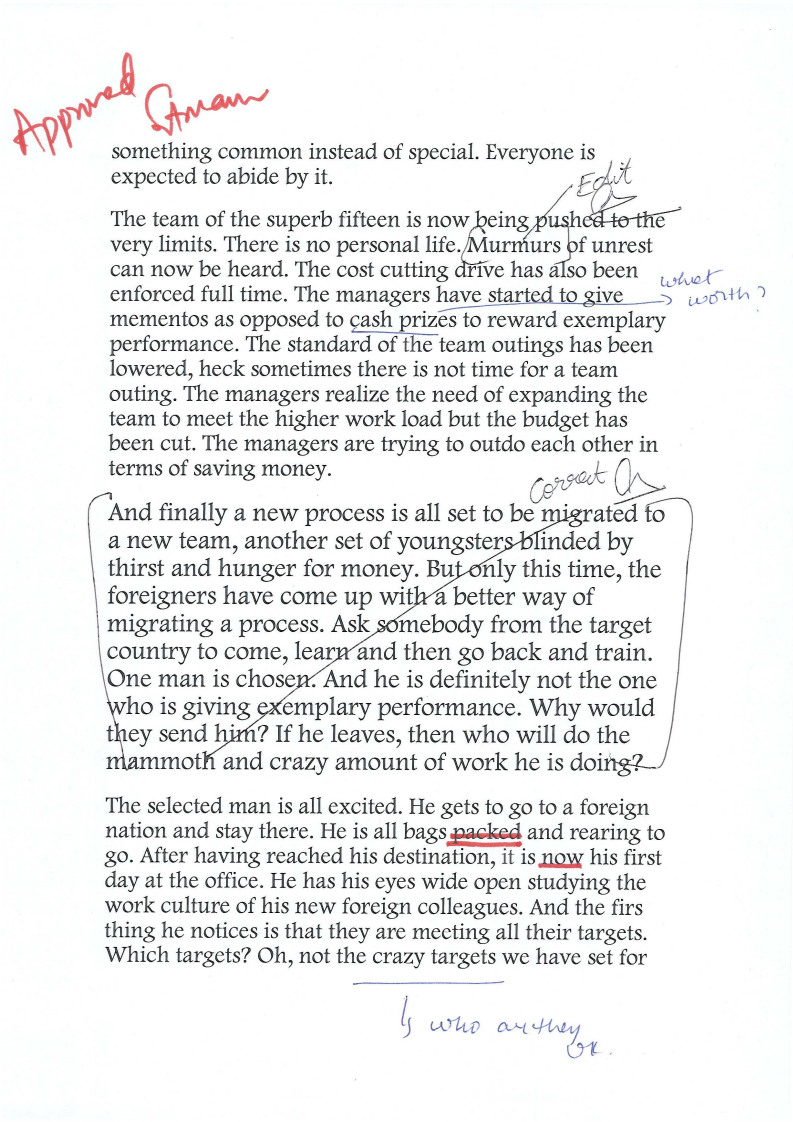
Throughout the course of this controversy over ~~Barclay~~ BARCLAY's tar and nicotine delivery, Philip Morris and R. J. Reynolds have claimed the FTC smoking machine does not replicate how a ~~Barclay~~ BARCLAY is smoked by human smokers under normal conditions. In their latest submissions, both companies attempt to buttress this charge with ventilation studies. Yet, apart from the fatal methodological flaws of both studies, and apart from their failure to provide information needed to fully assess the reliability and significance of the results (both of which are discussed below), neither Philip Morris nor R. J. Reynolds have brought the Commission any closer to answering the question: what do smokers actually get from BARCLAY as compared to other 1 mg. cigarettes?

Instead, they only have replaced the alleged inadequacies of the FTC machine with the demonstrable biases, inaccuracies and artifices of new machines. Only Brown & Williamson has provided the Commission with direct evidence of the relative deliveries among 1 mg. cigarettes under normal human smoking conditions. As we shall show, our competitors' attacks on the cotinine research submitted by B&W are without merit, and designed merely to divert the Commission's attention from the sole reliable evidence presented that responds to the basic issue in this inquiry.

Before responding in detail to the presentations of R. J. Reynolds and Philip Morris, the following deficiencies in both ventilation studies should be noted:

2500002305

Figure 1: Example of an annotated machine printed document with noises from the Tobacco data set [1].



something common instead of special. Everyone is expected to abide by it.

The team of the superb fifteen is now being pushed to the very limits. There is no personal life. Murmurs of unrest can now be heard. The cost cutting drive has also been enforced full time. The managers have started to give mementos as opposed to cash prizes to reward exemplary performance. The standard of the team outings has been lowered, heck sometimes there is not time for a team outing. The managers realize the need of expanding the team to meet the higher work load but the budget has been cut. The managers are trying to outdo each other in terms of saving money.

And finally a new process is all set to be migrated to a new team, another set of youngsters blinded by thirst and hunger for money. But only this time, the foreigners have come up with a better way of migrating a process. Ask somebody from the target country to come, learn and then go back and train. One man is chosen. And he is definitely not the one who is giving exemplary performance. Why would they send him? If he leaves, then who will do the mammoth and crazy amount of work he is doing?

The selected man is all excited. He gets to go to a foreign nation and stay there. He is all bags packed and rearing to go. After having reached his destination, it is now his first day at the office. He has his eyes wide open studying the work culture of his new foreign colleagues. And the firs thing he notices is that they are meeting all their targets. Which targets? Oh, not the crazy targets we have set for

Figure 2: Example of an annotated machine printed document with overlapped text from the HP data set.

of the large variations of personal fonts and styles. Farooq et al. [8] used an EM based Bayesian neural network model in order to identify Arabic handwritten text in mixed documents which contain machine printed text, handwritten text, overlapped text and/or noises. They used multiple classifiers which were weighted by their posterior probabilities and introduced a new neurons layer which used an error function to penalize solutions that led to misclassification.

One potential problem of the previous research on text separation is that although the basic unit (zone, line or word) of the system works well to classify those patches whose sizes are typically larger than words, it cannot separate overlapped patches which contain both handwritten and machine printed text because they are considered as a single unit in the system. In [9], Zhao and Davis used color information of each pixel to segment picture foreground from background which can also be used for document segmentation. However, color information is rarely available in the document analysis field where the input is usually a scanned binary document image. Some work in computer vision also provides us with interesting clues to address the problem of overlapped text patches. Eran and Shimon presented a top-down/bottom up segmentation method using small fragments that contain common object parts [10]. In [11], Corso suggested a multilevel aggregation procedure which groups each pixel to a set of aggregate regions in a multilevel coarsening of the image. Alpert et al. [12] used a Bayes rule to determine whether or not to merge two neighbor regions and segment images using these regions. All these approaches show that we can use a unit which is smaller

than words as the basic element to separate overlapped text.

In recent years, inspired by the success of Markov Random Field (MRF) in the area of image processing and restoration [13, 14], considerable research has been made to extend MRF to document restoration and preprocessing. In [15, 16], Cao and Govindaraju proposed a method using small fixed size patches to represent handwriting and restore broken handwritten text based on a MRF frame work. Similarly, Banerjee[17] used a flexible MRF-based optimization framework to remove noise and restore machine-printed text. These methods can be looked at as an extension of Freeman's algorithm [13]. In [1], Zheng et al. proposed a two step approach to identify three different types of text in mixed documents. They initially separated text using a Fisher classifier followed by a Gibbs network based relabeling which was further optimized using a Highest Confidence First algorithm. A similar approach but using Conditional Random Field has been proposed by Shetty et al. [18].

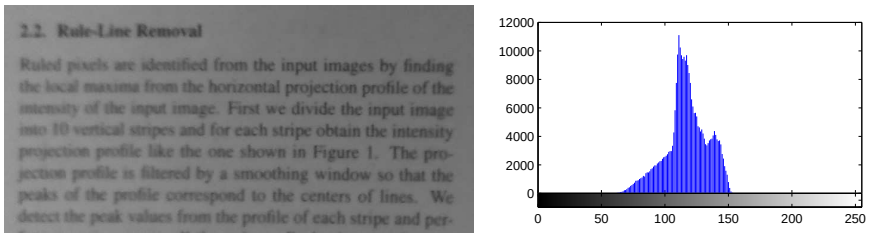
1.2 Binarization

Normally, the underlying components of the document processing systems, such as optical character recognition (OCR) and the proposed text separation system, rely on high quality binarized document images. Hence, pre-processing steps such as binarization play a critical role in the success of these systems. The traditional and most popular device for document image acquisition has been the

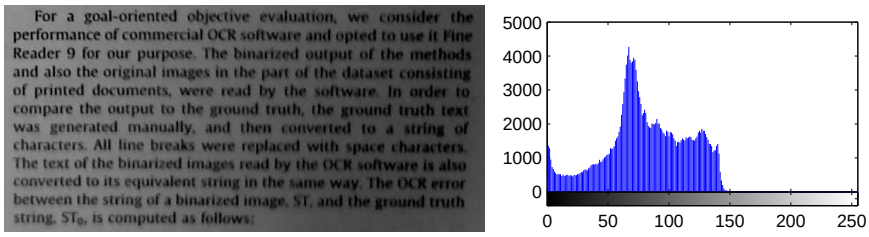
scanner which provides the most reliable digitizing result as users can control the capture environment. However, the advent of small, high resolution digital cameras and hand-held devices such as mobile phones with imaging capabilities have led to hand-held devices being widely used by users to capture document images of interest for future use.

Despite the convenience provided by hand-held devices, there are a number of factors that hinder downstream document processing tasks on images captured by these devices.

The first obstacle is a lack of control of lighting conditions which causes insufficient, non-uniform or over illumination of document images. This causes most global thresholding based binarization algorithms, such as Otsu's [19] method, to be ineffective. The idea of Otsu's method is to calculate a single threshold value that maximizes the inter-class (foreground and background) variance or minimizes the intra-class variance to segment the entire image. The drawback of this method is that it assumes the histogram of images has two distinct peaks for different classes respectively and can be separated. But this assumption is hardly satisfied in most real applications, especially in camera-captured document images. Fig.3 shows examples of degraded camera-captured document images, along with their corresponding histograms. It is observed that the histograms of these two camera-captured document images only have a single peak globally and any single threshold-based method will fail to separate foreground text from background.



(a) A document with insufficient illumination



(b) A document with uneven illumination where right part is darker than left part

Figure 3: Examples of hand-held devices captured document images with bad or uneven illumination.

To overcome the limitations of single threshold-based methods, researchers have tried mapping the original document image to a new feature domain prior to binarization. Valizadeh et al. [20] used a sigmoid function to enhance the document image before Otsu thresholding. Lu and Tan [21] employed two rounds of polynomial smoothing procedure to normalize gray scale images and binarized them using a single threshold. Similarly, a linear plane estimation method was proposed by Shi and Govindaraju [22] to segment text from degraded historical documents. Pilu and Pollard [23] used a retinex algorithm to estimate the reflectance surface on which they applied their binarization.

For images with a wide variation in intensities across the document image, researchers have used many local or adaptive thresholding methods for binarization. The earliest attempt at using local thresholds can be traced back to Niblack's method [24] which utilizes mean value and variance within a small window to determine the property of centered pixels and is formulated as:

$$T(i) = \mu(i) + k \times \sigma(i) \tag{1.1}$$

where $\mu(i)$ and $\sigma(i)$ are mean value and standard variance in a small window centered on pixel i, k is a parameter less than 0. The potential problem of this method is that large amount of noise is introduced in pure blank background areas and it is sensitive to the window size. An improvement of Niblack's method was described by Sauvola et al [25] using the formula:

$$T(i) = \mu(i) + \left[1 + k\left(\frac{\sigma(i)}{R} - 1\right)\right] \tag{1.2}$$

where $\mu(i)$ and $\sigma(i)$ have the same meaning as Equation 1.1, k takes positive values and R is 128 for a 8-bits grayscale document image.

Both Niblack's algorithm and Sauvola's algorithm are sensitive to the parameter k and window size s which limit their performances, so Bukhari et al. [26] suggested an adaptive method to estimate free parameters according to local ridge properties. By using Otsu's method locally, Moghaddam and Cheriet [27] calculated the optimal window size based on stroke width and text line height within the document image.

Another drawback of a document image captured using a hand-held devices is that the low cost lenses that are typically used can cause out of focus blur and down sampling blur and produce lots of noise [28, 29, 30, 31].

1.3 Proposed System

In this dissertation, we focus our researches on two main parts: binarization of hand-held devices captured document images and text separation for binarized document images.

As described in previous sections, the most popular document image acquisition techniques are scanners and cameras. Most scanners provide environment control techniques and intergrade the binarization algorithm into the hardware system, so in this dissertation, we only explore a novel binarization algorithm for hand-held camera captured document images.

The second part of our system which is text separation of document images contains two modules: word level text separation and overlapped text separation. The word level separation module considers each patch whose size is approximately compare to the size of words as the basic element and separates the entire document as machine printed text, handwritten text, overlapped text or noises. The overlapped text separation module which takes the overlapped text as its input uses smaller patches as the basic element and separate overlapped to machine printed text and handwritings.

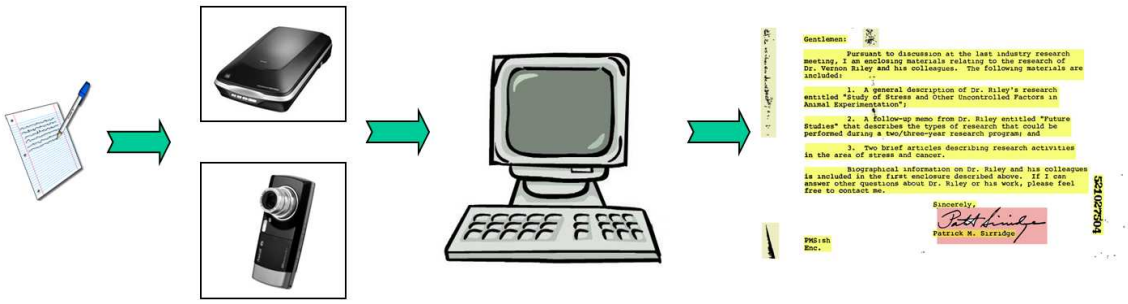


Figure 4: Overall system for document text acquisition and separation.

The overall structure of the proposed system is shown in Fig 4 where the data acquisition can be from scanner or hand-held camera. If the document image is captured from hand-held camera, the proposed MRF based binarization algorithm will be applied. The output of our system is the separated document where different regions are labeled as different colors or different documents.

The remainder of dissertation is organized as following. In chapter 2, the terminologies and basic concepts of Markov random fields are introduced. The Markov random field framework will be used repeatedly in our system for different purposes including document binarization, word level text separation and overlapped text separation. In chapter 3, a Markov random field based binarization algorithm for hand-held camera captured is described. In this chapter, a non-linear transformation function and edge potential based features are proposed. In chapter 4, we will study the system of separating annotated machine printed documents in word level. A Markov random field based relabeling method is described in this chapter. To overcome the imbalanced data set problem, a boosted tree structured

classifier which is inspired by the Adaboost classifier is proposed in chapter 5. In chapter 6, a MRF framework is proposed to further separate overlapped text to machine printed text and handwritings. Finally we conclude the dissertation in Chapter 7.

Chapter 2

Terminologies and Concepts of MRF

Markov random fields, MRF, which are used in both binarization algorithms for camera-captured document image and text separation in our dissertation, are graphical models in which a set of random variables have a Markov property described by an undirected graph and "*provides a convenient and consistent way of modeling context-dependent entities such as image pixels and correlated features*" [32].

The early use of MRF was the exact inference which is a sharp-P-complete problem. The proof of the equivalence of MRF and Gibbs field by Hammersley and Clifford [33] stimulates the widely use of MRF which calculates the optimal configuration of MRF using different range of energy functions and only by taking

consideration of a simple mathematic form. MRFs are widely used in image processing and document analysis field, including image binarization [34, 35, 36], document restoration and image super-resolution [14, 37, 38, 16, 39, 40], image and document segmentation [11, 41, 1], noise remove and filtering [17], etc. These applications can be categorized as the low-level processing. High-level processing in image processing and vision field includes object detection and recognition. Li proposed a general framework of using MRF for computer vision problem by converting the high-level vision problem to a low-level labeling problem [42] and he used this idea to match object in [43]. To detect objects in the image, Sheikh designed a two step algorithm which used the probability density to estimate the object and refined the final result by using MAP-MRF decision framework [44].

From the view of mathematic, the MRF can be considered as a labeling procedure which is a proper model for the purpose of the document image binarization and the text separation (labeling). In the following sections, we introduce the basic concepts and terminologies of MRF and inference of MRF.

2.1 Definition

We assume that the pixel/text patch/overlapped text/noise in document image I is represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{\mathcal{V}_1, \mathcal{V}_2, \cdots, \mathcal{V}_n\}$ is the set of vertices which correspond to pixels or patches or aggregations in the image and $\mathcal{E}$ is the set of edges which connect vertices based on a neighborhood system.

The neighborhood system can be based on 4-neighbors lattice connectivity or any other non-grid based metric defined by the user.

We denote the random field as $\mathcal{X}$, which is indexed by vertices of graph $\mathcal{G}$. Let Ω be the set of all possible configurations (labels) of $\mathcal{X}$, and $\mathcal{Y}$ be the observations which are the corresponding features for each pixel/patch/aggregation which are defined in the following chapters.

Given the graph $\mathcal{G}$, the Markov Random Field is used to compute an optimal configuration of $\mathcal{X}$ which maximizes the posterior:

$$\overline{\mathcal{X}} = \arg\max_{\mathcal{X}} P(\mathcal{X}|\mathcal{Y}) = \arg\max_{\mathcal{X}} \prod_{i=1}^{n} P(x_i|y_i, x_{\mathcal{V}-\{i\}}) \tag{2.3}$$

where y_i and x_i are the observed feature and hidden configuration respectively for a given vertex $\mathcal{V}_i$, and $\mathcal{V}-\{i\}$ is the set of all vertices in graph $\mathcal{G}$ except vertex $\mathcal{V}_i$. This equation shows that the configuration of site $\mathcal{V}_i$ is dependent on its observation and all other vertices in the graph.

By taking the Bayes rule and the Markov property, Equation 2.3 can be rewritten as:

$$\begin{aligned}
\overline{\mathcal{X}} &= \arg\max_{\mathcal{X}} \prod_{i=1}^{n} \frac{P(y_i|x_i, x_{\mathcal{V}-\{i\}})P(x_i|x_{\mathcal{V}-\{i\}})}{P(y_i|x_{\mathcal{V}-\{i\}})} \\
&= \arg\max_{\mathcal{X}} \prod_{i=1}^{n} P(y_i|x_i, x_{\mathcal{V}-\{i\}})P(x_i|x_{\mathcal{V}-\{i\}}) \\
&= \arg\max_{\mathcal{X}} \prod_{i=1}^{n} P(y_i|x_i)P(x_i|x_{N(i)})
\end{aligned} \tag{2.4}$$

where prior $P(x_i|x_{N(i)})$ means the configuration of site $\mathcal{V}_i$ is conditioned by its immediate neighbors $N(i)$ and likelihood $P(y_i|x_i)$ describes the relation of a configuration and its corresponding observation for a given vertex $\mathcal{V}_i$. Equation 2.4 shows that the configuration of vertex $\mathcal{V}_i$ is dependent on its observation and immediate neighbors only.

According to Hammersley's theory which shows the equivalence of Markov random field and Gibbs random field, the set of random variable x on graph $\mathcal{G}$ is said to be a Gibbs random field with respect to system of neighborhoods $\mathcal{N}$. A Gibbs distribution is defined as:

$$P(x) = \frac{1}{Z}\exp\{E(x)/T\} \tag{2.5}$$

where Z is a normalizing constant value, T is a constant called the temperature and E is the energy function. The energy function $E(x) = \sum_{c\in C} V_c(x)$ is a sum of clique potentials V_c over all possible cliques $c \in C$. The first order and second order of the neighborhood system are most frequently used in applications. Fig. 5 and Fig. 6 show the regular lattice 4-neighborhood system and 8-neighborhood system and their corresponding cliques. In most cases, the energy function of Gibbs random field is expressed as several terms where each term represents the energy of cliques of different orders:

$$E(x) = \sum_{c \in C} V_c(x)$$
$$= \sum_{i \in C_1} V_i(x_i) + \sum_{(i,j) \in C_2} V_{i,j}(x_{i,j}) + \cdots \quad (2.6)$$

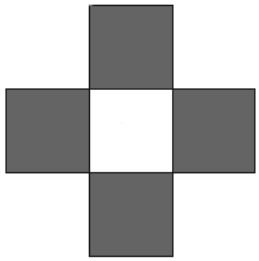
In this dissertation, we only uses the first order neighborhood system in the binarization algorithm, which is described in chapter 3 in detail. In text separation, because patches (words) are not located as the rigid grid, an irregular neighborhood and cliques system [32] are used for the word level text separation and the overlapped text separation.

2.2 Inference of MRF

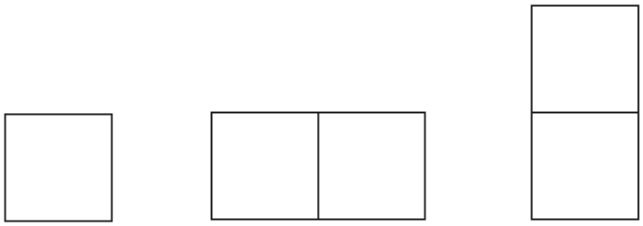
Inference of the MRF model can be achieved using either global or local optimization algorithm. The global optimization method calculates the conditional distribution of each node by summing over all possible configurations which leads to a minimal solution of energy function. The widely used global minimization methods include simulated annealing, graph cuts and genetic algorithms.

Simulated annealing (SA) which is inspired from annealing in metallurgy computes an approximation of the global optimum of a given energy function. More backgrounds of SA can be found in [45, 46] and the use of SA for MRF can be found in [47].

The graph cut was first used by Greig et al. [48] for MAP (maximum a posteriori) estimate of a binary image. In graph theory, given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ which



(a) Regular lattice of 4-neighborhood system



(b) Cliques for 4-neighborhood system

Figure 5: 4 neighborhood and its clique system

has a set of nodes $\mathcal{V}$ and weighted edges $\mathcal{E}$, two more special nodes (terminals or labels) which are called source s and sink t are added into the graph $\mathcal{G}$. To two-label MRF model, such as MRF for binarization, source s and sink t represent the two labels of the system. All edges which are not connected to terminals represent the neighborhood system of the graph and all edges connected to two terminals measures the penalty for assigning the corresponding terminal to the nodes. The cost associated with all edges corresponds to the energy of the function 2.6. According to the theory from [49], the maximum flow from s to t is corresponding

to a minimum cut in the graph $\mathcal{G}$ which divides the nodes in the graph to two separate parts. And each part is assigned one of the labels. Thus, the graph cuts algorithm calculates the minimum energy of function 2.6 of Gibbs random field using the max-flow/min-cut theorem [50]. In this dissertation, a modified graph cuts algorithm [51] is used to calculate the optimal binarized document image in chapter 3 by introducing a source to represent the foreground configuration and a sink to represent the background configuration.

A typical local optimization method for calculating the minimal energy of MRF is the belief propagation (BP) method. This method is suitable for unregular lattice system for MRF which is covered in [32]. We use the topology as shown in Fig. 7 to illustrate the belief propagation and message update procedure for our MRF model. The messages in a network propagate in two opposite directions [52, 13, 53] which lead to two update rules for vertex $\mathcal{V}_i$.

1) The procedure for calculating *maximum a posteriori probability* (MAP) of belief can be expressed as:

$$\hat{x_i} = \arg\max_{x_i} P(y_i|x_i) \prod_{j \in N(i)} m_{i,j}(x_i) \tag{2.7}$$

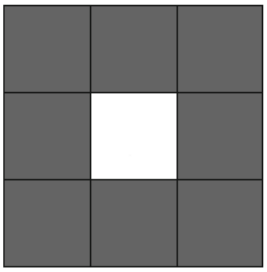
where $m_{i,j}(x_i)$ is the incoming message to vertex $\mathcal{V}_i$ from its neighbor $\mathcal{V}_j$ and j runs over all neighbors of vertex $\mathcal{V}_i$. Fig. 7(a) shows the process of computing the belief of vertex $\mathcal{V}_i$.

2) The method to update the message from vertex $\mathcal{V}_j$ to vertex $\mathcal{V}_i$ is given by:

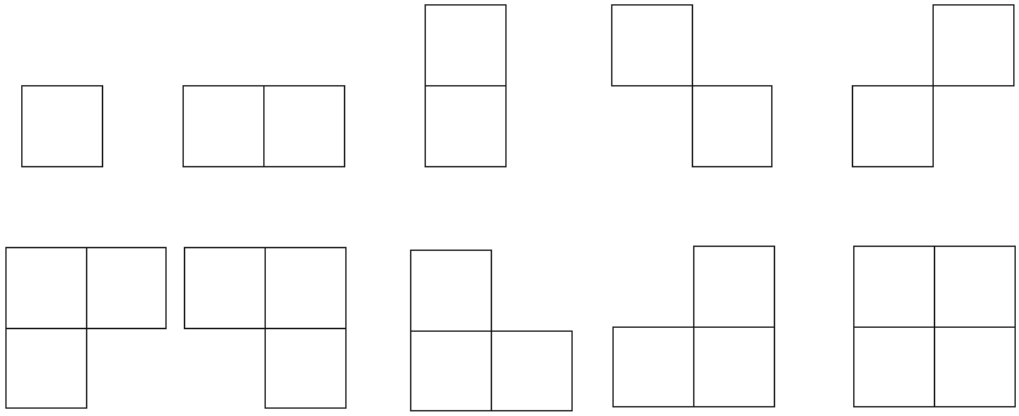
$$m_{i,j}(x_i) = \max_{x_j} P(x_i|x_j) P(y_j|x_j) \prod_{k \in N(j) \setminus i} m_{j,k}(x_j) \tag{2.8}$$

where k runs over all neighbors of vertex $\mathcal{V}_j$ except vertex $\mathcal{V}_i$. Fig. 7(b) shows the procedure for updating messages for $m_{i,j}$.

The prior $P(x_i|x_j)$ and likelihood $P(y_i|x_i)$ in Equation 2.7 and 2.8 are defined as a similarity function $\Psi_h(x_i,x_j)$ and dependency function $\Psi_o(x_i,y_i)$ respectively in our MRF model and are described in detail in section 4.5.2 for module of word level text separation and section 6.3.2 for module of overlapped text separation.

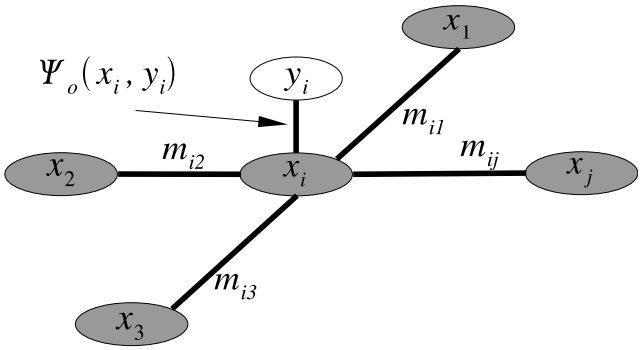


(a) Regular lattice of 8-neighborhood system

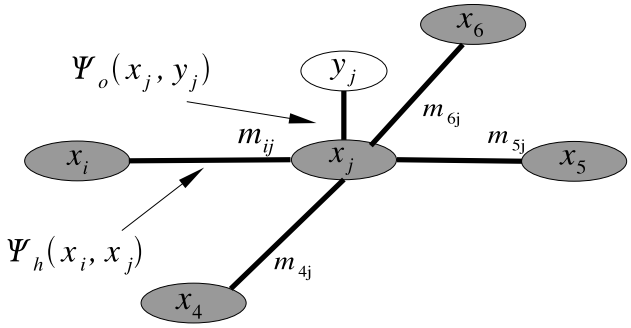


(b) Cliques of 8-neighborhood system

Figure 6: 8 neighborhood and its clique system



(a)



(b)

Figure 7: Message update and belief propagation in MRF. Each grey node x_i which is a configuration for a text patch/aggregation connects to its four nearest neighbors. Each observation node y_i which is a feature point in the feature space connects to its hidden label x_i. Edges between any pair of nodes indicate their similarity.

Chapter 3

Binarization of Document Images

3.1 Introduction

As discussed in previous chapters, we consider two types of data acquisition methods in this dissertation: scanner and hand-held camera. As most scanners have an environment control method which can reduce the noises and provide efficient illumination, with hardware integrated binarization technique or simple adaptive binarization software, it is convenient to obtain high quality binarized document image. On the other hand, the document images captured by using hand-held suffer from noise, uneven, bad or over illumination, and out of focus distortion. In this chapter, we focus our research on binarization of hand-held camera captured document images and propose a MAP-MRF based framework to restrain noises as well as keep the continuity of the stroke.

3.2 Initial Binarization

Due to uneven or bad illumination, the intensity of background may not be consistent within a camera captured document image. Fig.8(a) shows an example document image captured under uneven lighting condition whose background is darker on the left part of image than on the right side. Fig.8(b) shows the vertical profile intensity of this document image and it is apparent that any single threshold based binarization method will not work since the intensity varies gradually from left to right. Ideally, the thresholding should be an adaptive curve that tracks the intensity. The mean value of intensity in a window around a given pixel is a likely threshold which is the basis of many adaptive thresholding binarization techniques [24, 25]. As shown in Fig. 8(b), the average of profile intensity which is calculated using a 10×10 sized window crosses the intensity curve and can be used as a starting point to estimate an optimal local threshold.

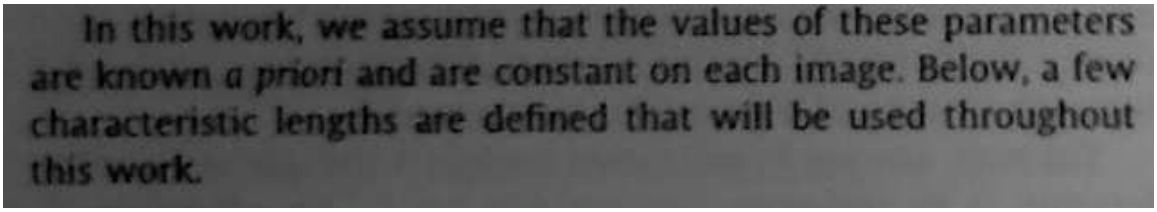
In our work, prior to estimating the adaptive threshold surface, we compute the mean value:

$$\mu_i = \sum_{x,y} I(x,y)/(m \times n) \tag{3.9}$$

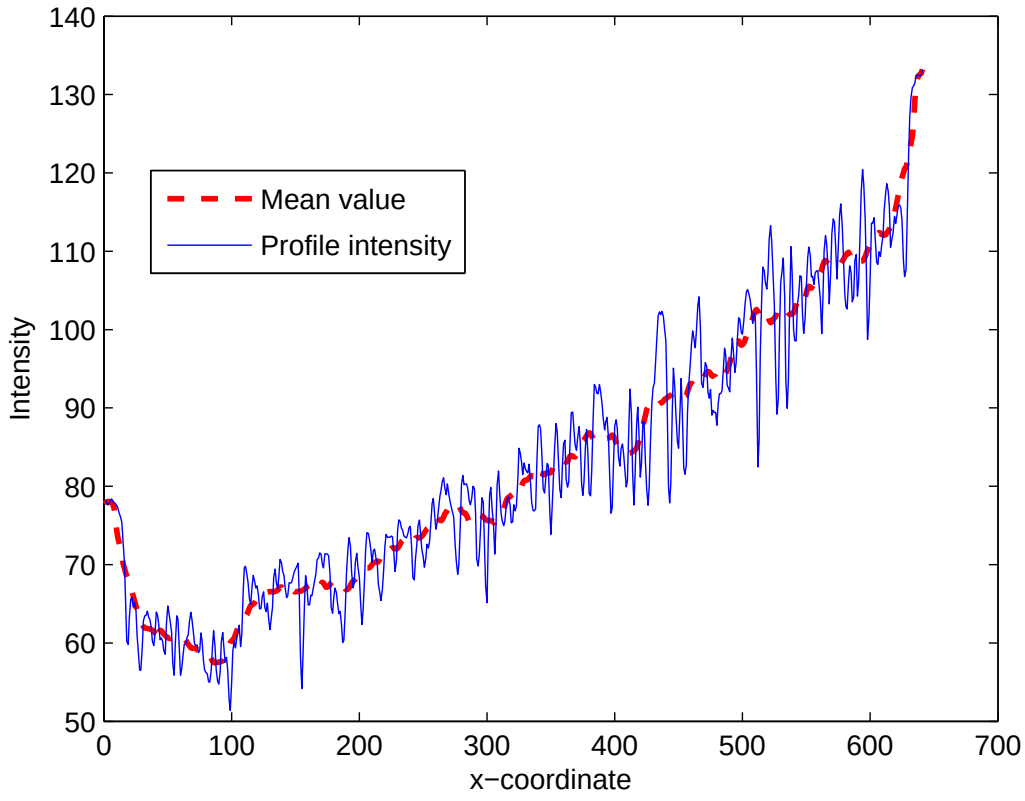
and standard deviation:

$$\delta_i = \sqrt{\sum_{x,y} (I(x,y) - \mu_i)^2/(m \times n)} \tag{3.10}$$

for a given pixel i using a $m \times n$ sized window centered on this pixel. The maximum variance δ_{max} and minimum variance δ_{min} are also obtained over the entire



(a)



(b)

Figure 8: A sample image and its vertical profile

document image. The adaptive threshold for each pixel is calculated according to a logistic function as described in Equation 3.11:

$$O(i) = \mu_i \left\{ \frac{1-k}{\left(1+e^{-B\left(\frac{\delta_i-\delta_{min}}{\delta_{max}-\delta_{min}}-M\right)}\right)^{1/\nu}} + k \right\} \tag{3.11}$$

where B controls the growth rate of the logistic curve, M and ν affect the time for which maximum growth occurs and k is the minimum value the curve can achieve.

The idea behind Equation 3.11 is that text areas have larger values of variance and mean value of given pixel i can be a candidate threshold. On the other hand, the variance of background pixels is small and the threshold should be smaller than the mean value to avoid noise. Then the free parameters of Equation 3.11 are chosen such that the threshold is close to mean value μ_i if variance δ_i is big and the threshold is not close to the mean value if variance δ_i is small.

To enlarge the difference between foreground and background, we map the original document image into a new domain using Eq. 3.12:

$$\mathcal{F}(i) = \begin{cases} 128 + 128 \times \left(\frac{I(i)-O(i)}{db_{max}}\right)^{\alpha} & if \quad I(i) \geq O(i) \\ 128 - 128 \times \left(\frac{O(i)-I(i)}{df_{max}}\right)^{\alpha} & if \quad I(i) < O(i) \end{cases} \tag{3.12}$$

where $O(i)$ is the adaptive threshold obtained from Eq. 3.11 for pixel i, $I(i)$ is the original gray intensity of the pixel, df_{max} is the maximum difference between foreground and segmentation surface and db_{max} is the maximum difference between

background and segmentation surface respectively, α is a parameter that controls the growth rate of the mapping curve and is set to be 0.1 in our experiment.

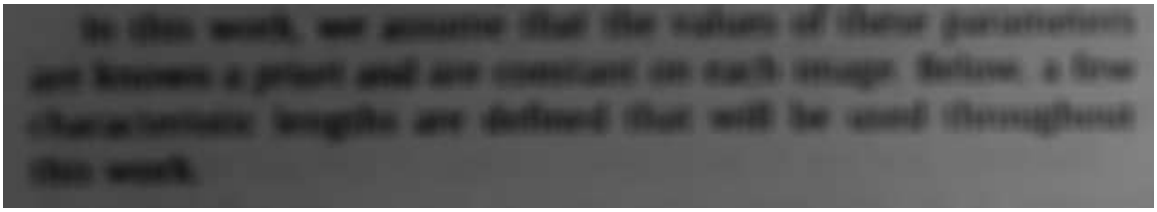
Fig. 9(c) shows the adaptive threshold surface calculated using Equation 3.11, the normalized image using Equation 3.12 and the initial binarization result of Fig. 8(a) using the normalized image. From Fig. 9(a) we can see that the intensity of the threshold surface changes gradually along with the intensity of the document image and the result of binarized image has consistent text stroke width which is not affected by the variation in intensities caused by uneven lighting.

3.3 MRF based Relabeling

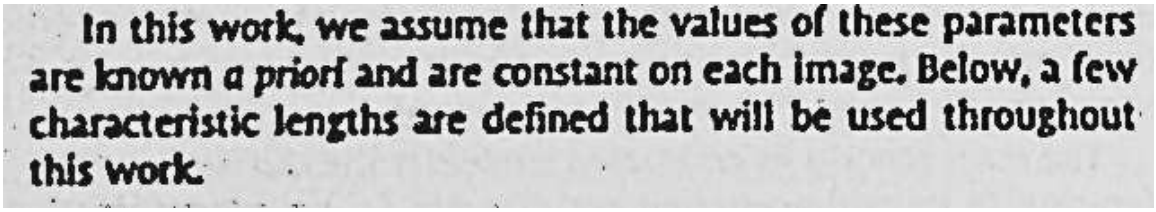
Normally, noise in the background and holes within text cannot be avoided by using adaptive threshold based binarization only as shown in Fig. 9(c). In this chapter, we propose a Markov random fields based relabeling procedure to remove noise and holes from the initial binarized document image.

3.3.1 MRF and Gibbs Model

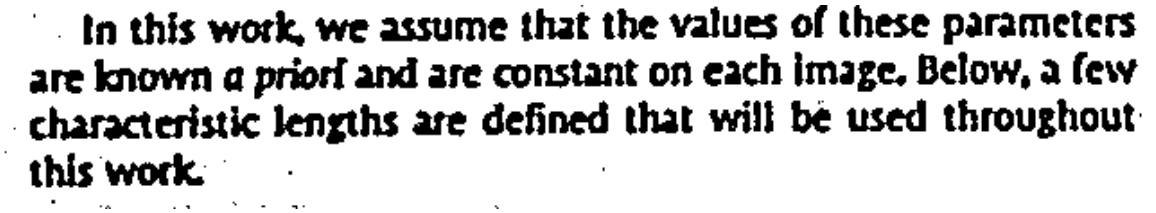
In recent years, Markov Random Fields (MRF) based image restoration algorithms have attracted interest from researchers in document processing. Considering the ideal machine printed black-white document image to be a binary image which is down-sampled and blurred to a gray-scale image by adding Gaussian noise, document binarization can be looked at as a special restoration problem. Lelore



(a)



(b)



(c)

Figure 9: Adaptive threshold surface, normalized image and initial binarization result

and Bouchara [34] proposed a MRF model for binarization which can remove noise and improve the character connectivity. Lettner et al. [35] used a similar framework but defined the Gibbs distribution in a different form to binarize the degraded documents. To binarize unevenly illuminated documents, Kuk and Cho [36] initially segmented the entire document image using mean filter response as features and relabeled the pixels using a minimization technique which can be

looked at as a variant of Gibbs model.

The relabeling of the initial binarized document image which assigns one of two labels (black or white) to each pixel of the document can be modeled as a maximum a *posteriori* Markov random field (MAP-MRF) estimation of ideal binarized document image x given only the degraded camera captured image y. In our MRF framework, the observed degraded image y is the original gray-scale document image along with its initial binarized image and our task is to calculate an optimal configuration of x which maximizes the posteriori:

$$\begin{aligned} \overline{x} &= \arg\max_{x} P(x|y) \\ &= \arg\max_{x} \prod_{i=1}^{n} P(x_i|y_i, x_{N-\{i\}}) \end{aligned} \tag{3.13}$$

where y_i is the observed feature and x_i is hidden configuration of pixel i respectively, and $N-\{i\}$ denotes all pixels in document image except pixel i.

By taking Bayesian rule and considering Markov property, the MAP-MRF estimation can be re-written as:

$$\begin{aligned} \overline{x} &= \arg\max_{x} \prod_{i=1}^{n} \frac{P(y_i|x_i, x_{N-\{i\}})P(x_i|x_{N-\{i\}})}{P(y_i|x_{N-\{i\}})} \\ &= \arg\max_{x} \prod_{i=1}^{n} P(y_i|x_i, x_{N-\{i\}})P(x_i|x_{N-\{i\}}) \\ &= \arg\max_{x} \prod_{i=1}^{n} P(y_i|x_i)P(x_i|x_{N(i)}) \\ &= \arg\min_{x} \left[-\sum_{i=1}^{n} logP(y_i|x_i) - \sum_{i=1}^{n} logP(x_i|x_{N(i)}) \right] \end{aligned} \tag{3.14}$$

where likelihood $P(y_i|x_i)$ represents the dependency of observations on hidden configuration and prior $P(x_i|x_{N(i)})$ shows the influence from immediate neighbors $N(i)$

to centered pixel i. In our work, we use a 4-connected lattice system where each pixel has four neighbors.

Generally, the optimal configuration X of Equation 3.14 can be achieved by minimizing energy function [51, 54]:

$$E(X) = \sum_{i \in \mathcal{V}} U_i(x_i) + \sum_{(i,j) \in \mathcal{E}} V_{i,j}(x_i, x_j) \tag{3.15}$$

where $\mathcal{V}$ is the vertex corresponding to pixels in the image and $\mathcal{E}$ is the edge connection between pixels, $U_i(x_i)$ denotes the unary energy which is derived from $-logP(y_i|x_i)$ and pairwise energy $V_{i,j}(x_i, x_j)$ is derived from $-logP(x_i|x_j)$ in Equation 3.14 respectively. The unary energy $U_i(x_i)$ tends to force hidden configuration x_i to have a value which is compatible with its observation y_i and pairwise energy $V_{i,j}(x_i, x_j)$ forces x_i to be smoothly connected with its neighbors.

3.3.2 Edge potentials

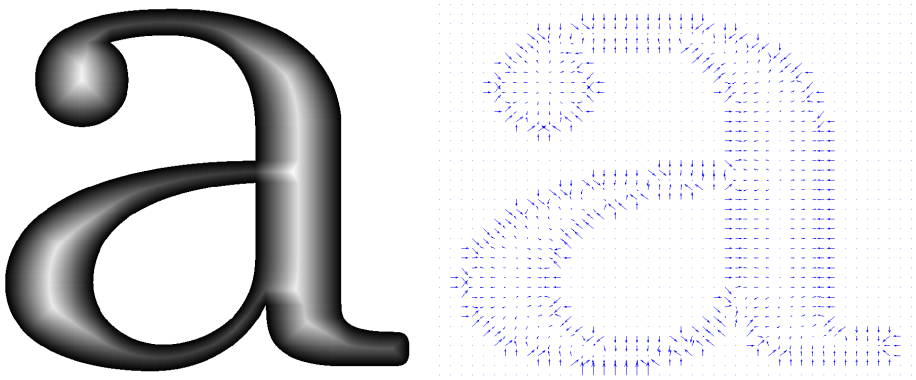
Unlike other MRF based relabeling algorithms which only use the intensity difference between neighboring pixels and smooth the entire document image [36, 35], we explore a stroke width-related feature which preserves the edge of strokes and removes noise from the document.

For each connected text component of the binarized document image obtained from section 3.2, we compute the shortest distance from foreground pixels to the background which is denoted as $e_n(i)$ for pixel i in connected component n. The maximum distance from the foreground pixel within a connected component n to

the background is represented as $\hat{e}_n$. To measure the potential of a foreground pixel to be on the edge, the distance from the foreground pixels to the innermost pixel within a connected component is calculated as:

$$s_n(i) = \hat{e}_n - e_n(i) \tag{3.16}$$

Fig. 10(a) shows the edge potential of an example character **a** where the brighter area in the character corresponds to a low edge potential and the darker area corresponds to higher edge potential. The velocity vectors which indicate the direction and strength of edge potential for each foreground pixel are shown in Fig. 10(b), from which we can see that the edge potential decreases gradually from edge pixels to inner pixels within the character.



(a) (b)

Figure 10: Edge Potential

3.3.3 Unary $U(x)$ and Pairwise $V_{i,j}(x_i, x_j)$ Energy Function

The goal of the MRF-based relabeling procedure is to remove noise and smooth the entire document image. As described in section 3.3.1, $-logP(y_i|x_i)$ can be represented by an unary energy function $U_i(x_i)$ which forces the label x_i of pixel i to be close to its observation y_i. We define $U_i(x_i)$ as:

$$U_i(x_i) = \lambda\sqrt{(y_i - x_i)^2} \tag{3.17}$$

where y_i is the gray-scale value for pixel i and x_i takes value of 255 for background and 0 for foreground.

Pairwise energy function $V_{i,j}(x_i, x_j) = -logP(x_i, x_j)$ is defined in Equation 3.18 using edge potential features along with the gray-scale value for each pair of pixels,

$$V_{i,j}(x_i, x_j) = \begin{cases} \alpha \exp\left(\frac{1}{|[s(i)-s(j)]^2-(sw/2)^2|+1}\right) + \beta \exp\left(\frac{|y_i-y_j|}{256}\right) & if \quad x_i = x_j \\ \alpha \exp\left(\frac{-1}{|[s(i)-s(j)]^2-(sw/2)^2|+1}\right) + \beta \exp\left(\frac{-|y_i-y_j|}{256}\right) & if \quad x_i \neq x_j \end{cases} \tag{3.18}$$

where x_i and x_j are the hidden configuration (0 for foreground and 255 for background) for pixel i and j, $s(i)$ and $s(j)$ are edge potentials for pixel i and j using Equation 3.16, sw is the mean stroke width within the document image, and y_i and y_j are the gray-scale values for two neighboring pixels.

The underlying principle of Equation 3.18 is that if two neighboring pixels are from the same source, e.g. both of them are from foreground text, they should have similar edge potentials and gray-scale values which cause the pairwise energy to be low.

To achieve the global minimum energy corresponding to the optimal configuration of MRF, the graph cuts algorithm [51] is used in our experiment repeatedly until the binarized result is stable. The overall procedure of our algorithm is described in Fig. 11.

3.4 Experimental Results

In our experiment, we use a data set of 28 pages of document images which were captured using a hand-held cell-phone camera with a resolution of 3.8 mega pixels. The document pages in our data set are research papers in two column style and the text occupies at least 95% of the page area. All images were captured in an indoor office environment and our experiment was carried out using image portions with insufficient or uneven illumination along with out of focus blur. Fig. 12(a) shows an example image in our data set.

We used both qualitative judgement in the form of visual inspection as well as a quantitative metric based on OCR performance to evaluate the proposed binarization algorithm.

Fig. 12 shows an example of visual comparison of the binarization results with Otsu method, Niblack method, Sauvola method and the proposed MRF based method. Fig. 12(b) is the binarized result of Otsu method where strokes on the left portion of the image are thicker than those on the right portion because of uneven illumination. Although the stroke width is consistent in the result from

the Niblack method, a lot of noise is introduced in pure background areas as shown in Fig. 12(c). The Sauvola method has better performance as shown in Fig. 12(d) where noise is restrained and strokes are retained. The last Fig. 12(e) shows the result of the proposed MRF based binarization method which not only removes all isolated noise, but enhances the quality of strokes.

The goal of most binarization algorithms is to provide a reliable binarized image for further document processing such as Optical Character Recognition (OCR). Thus, we compared OCR results on the binarized images generated from the four different binarization methods considered in our experiments. The OCR experiment was carried out using the open source OCR software Tesseract [55] without deskew or other pre-processing procedures. Normalized mean edit distance between OCR results and ground truth was calculated and shown in Table. 1. Edit distance between two strings calculates the number of operations (insertion, deletion or substitution) to transform one of them into the other. It is obvious that the greater the edit distance, the more different the two strings are. There are several different definitions of the edit distance and we used the definition from Vladimir Levenshtein (Levenshtein distance) [56] which apply the bottom-up dynamic programming algorithm in our experiments.

As can be observed from Table 1, the proposed MRF based binarization algorithm has the shortest Levenshtein distance (Edit distance) to the ground truth

which means it provides more reliable OCR accuracy than the other three methods whereas the Niblack method has the worst OCR performance since it introduces more noise into the image.

Table 1: OCR Results from Tesseract on binarized images

Method	OCR Result(Levenshtein Distance)
Otsu [19]	0.561
Niblack [24]	1.165
Sauvola [25]	0.485
Proposed method	0.322

3.5 Conclusions

In this chapter, we have presented a binarization algorithm that focused on handheld devices captured document images with insufficient or uneven illumination. The algorithm mainly contains two key parts: initial segmentation/nomalization and MAP based binarization. Initially, a logistic based non-linear function is proposed to estimate the segmentation surface for entire document image and then the image is normalized by using segment surface and another non-linear function. The MAP based binarization procedure uses a MRF framework to segment the foreground text from background area. In this framework, the normalized document image is used as the observation, and a novel pairwise energy function

is defined in this paper to measure the relationship between neighboring pixels which preserves the stroke edge, removes the noises and smoothes the entire document image at the same time. Experiment results show that our method outperforms other approaches.

MRF Relabeling Algorithm

Input: Gray-scale document image I.

Initial Binarization:

1: Estimate adaptive threshold surface using Equation 3.11;

2: Segment text from background using adaptive threshold.

3: Initialize difference threshold t and maximum iteration number N.

Relabeling:

4: Extract edge potential feature on binarized document image for each pixel according to Equation 3.16;

5: Compute the unary energy of each pixel for the entire document image using Equation 3.17;

6: Calculate the pairwise energy of each pixel for the entire document image using Equation 3.18;

7: Use graph cuts algorithm to get the optimal relabeling of the binarized image;

8: Calculate the difference $\varepsilon^{(n)}$ between the relabeled image and the previous binarized image:

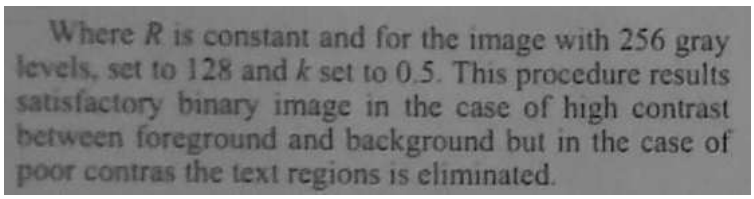
$\varepsilon^{(n)} = \sum_i \sqrt{(x_i^{(n)} - x_i^{(n-1)})^2}$

where $x^{(n)}$ is the configuration after relabeling for pixel i and $x^{(n-1)}$ is the order label for pixel i;

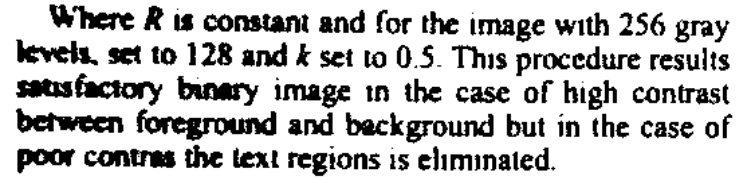
9: If $\varepsilon^{(n)} < t$ or $n > N$, go to next step, otherwise, $n = n+1$ and go back to step 4;

Output: binarized result.

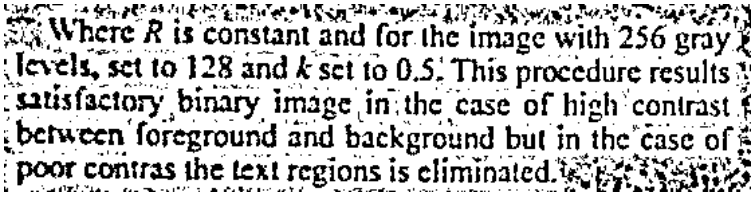
Figure 11: Overall Procedure of Binarization



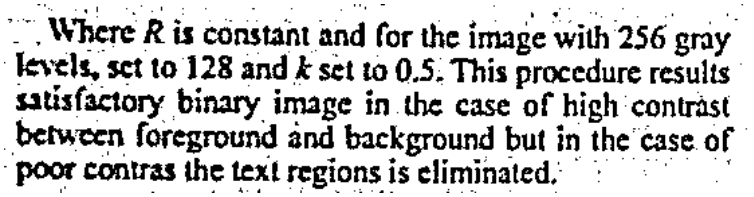
(a) Original



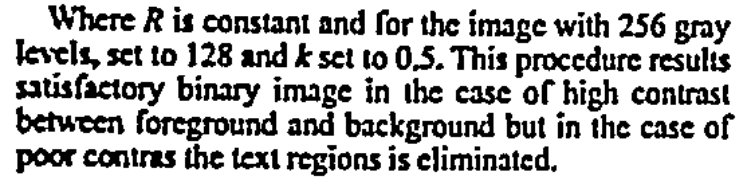
(b) Otsu



(c) Niblack



(d) Sauvola



(e) Proposed method

Figure 12: Binarization result of uneven illuminated document image using proposed algorithm compared with other methods. (a) The original grayscale document image, (b) Otsu binarization result, (c) Niblack binarization result with $k = -0.3$ and $s = 11$, (d) Sauvola binarization result with $k = 0.02$ and $s = 11$, (e) Proposed MRF based binarization result.

Chapter 4

Word Level Text Separation

4.1 System Structure

In this dissertation, we propose a two-module composite system to separate handwritten text, machine printed text and overlapped text from annotated documents.

As shown in Fig. 13, the first module (module I) of the system, which is the word level text separation, takes the entire document as its input and separates the text into three classes: machine printed text, handwritten text and overlapped text. This module has three components. The first component is a preprocessing procedure to extract patches of the document and extract features for each patch and is covered in section 4.2. Section 4.3 describes the second part of module I which is a G-means [57] based initial classification component. The last component of module I is to relabel the result of initial classification using on MRF based model. Section 4.5 describes the details of the relabeling. Section 4.6 presents

the experimental results of this module.

The second module (module II) (Fig. 13) which is for overlapped text separation takes the overlapped text patch as its input which is then further separated into machine printed text and handwritten text. The first step in module II is to extract shape-context based features for each pixel. The next step is an aggregation coarsening algorithm which extracts the basic element for the second module. Section 6.1 and section 6.2 describe these two steps. The last step is the separation of machine printed text and handwritten text using on MRF model and is described in section 6.3. Section 6.4 shows the experimental setup and results of module II. Our conclusions is presented in section 4.7.

4.2 Preprocessing

Our pre-processing consists of two steps: patch extraction and feature extraction.

4.2.1 Patch Extraction

Prior to classification, each binarized document is segmented into patches which are small snippets of the image as described in [58]. The patches are the basic units in our MRF based classification system for module I which models each document as a random field. The extraction of patches is done by using a $m \times n$ sized window based morphology closing operation on the original binarized image and the original content within the bounding box of each connected component is

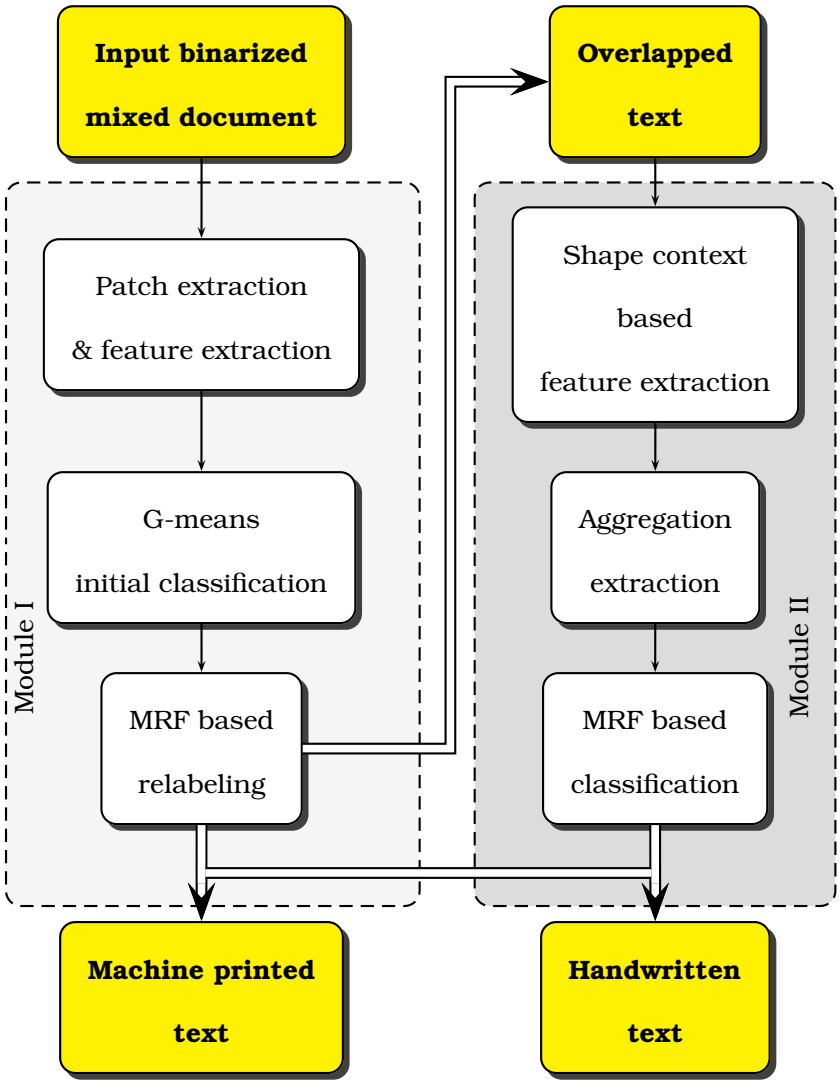


Figure 13: Overall structure of proposed system

Figure 14: The procedure to extract patches from a binarized document

defined as a single patch. The size of the window is empirically chosen such that the resultant patch typically represents a handwritten or machine-printed word. Patches are eliminated as noise if their size is smaller than a threshold t_l or larger than a threshold t_h. The procedure of patch extraction for a binarized document is shown in Fig 14.

4.2.2 Feature Extraction

Three different categories of features are considered for classification of a given patch into one of three classes viz., handwritten text, machine-printed text and overlapped text. These features are briefly listed in Table 2 and are described in greater detail in the following sections.

- **Patch Level Features:**

  Considering each patch as a single unit in our system, we extract several sets of features at the patch level. For a given patch, the first feature is its relative location x and y with respect to the entire document. The relative width w and

height h of the patch with respect to its nearest neighbor measures the size of this patch. The foreground density of the patch is measured by Equation:

$$d = \frac{\sum_{x,y} I(x,y)}{w \times h} \tag{4.19}$$

where $I(x,y)$ is the density of pixel (x,y) which represents black pixel as 0 and white pixel as 1 in our binary image.

Generally, machine printed text and handwritten text individually have constant stroke width. Therefore, we calculate the average stroke width using:

$$s = \frac{\sum_{x,y} I(x,y)}{l} \tag{4.20}$$

where l is the length of contour for a given patch. The crossing number within a patch measures the complexity of the stroke [58, 1] which counts the amount of pixels whose density differs from its direct neighbors in horizontal and vertical direction and is defined as:

$$\begin{aligned} c_x &= \frac{\sum_{x,y} I(x,y) \oplus I(x+1,y)}{h} \\ c_y &= \frac{\sum_{x,y} I(x,y) \oplus I(x,y+1)}{w} \end{aligned} \tag{4.21}$$

where $\oplus$ is the *exclusive and* operator.

The variances of horizontal and vertical projection of a patch and maximum runlength within the patch in these two directions are also used to measure the difference between the three different classes. Fig 15 and Fig 16 illustrate the process of computing these two sets of features.

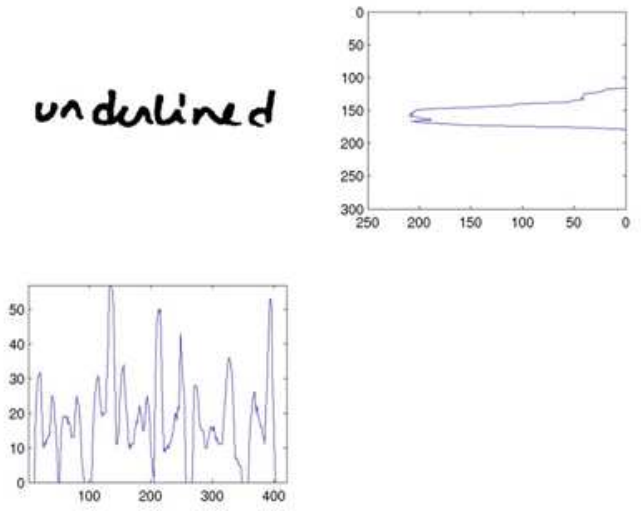


Figure 15: The projection of a patch

Figure 16: Maximum run-length of a patch

- **Connected Component (CC) Features:**

  A set of features is based on the statistical properties of connected components within a patch which is extracted from the original (non-dilated) binarized image.

  Assuming n is the number of connected components within a patch, we compute the mean and variance of width and height of each CC:

$$ave_w = \frac{1}{n}\sum_i w(i)$$
$$ave_h = \frac{1}{n}\sum_i h(i) \tag{4.22}$$

$$var_w = \sqrt{\frac{1}{n}\sum_i (w(i) - ave_w)^2}$$
$$var_h = \sqrt{\frac{1}{n}\sum_i (h(i) - ave_h)^2} \tag{4.23}$$

  where i runs over the number of CCs within a patch, $w(i)$ and $h(i)$ are the width and height of each CC respectively. Normally, machine printed character has a constant width and height in a certain range which results in a small value of variance. On the contrary, the width and height of the connected component within handwritten text are more likely to be scattered in the feature space and have larger variance value. The maximum width and height of CCs are also used as features.

  For a handwritten text patch, the area of loops within characters is typically larger than the area of loops for characters in a machine printed patch. So

the loop area normalized by the size of the patch is computed as a feature to distinguish handwritten text from the other two kinds of patches. Another CC feature is the overlap ratio which is the area of overlap between CCs within the patch divided by the total area of the patch. [58, 1].

- **Gabor Features:**

  Gabor filters can serve as directional band-pass filters which are modulations of a complex sinusoidal and Gaussian function. The 2-D Gabor filter is defined as Eq.4.24 in the space domain (details of parameters can be found in [59, 60, 61]):

$$g_{\lambda,\theta,\varphi,\delta,\gamma}(x,y) = K\exp\left(-\frac{x'^2+\gamma^2 y'^2}{2\delta^2}\right)\cos\left(2\pi\frac{x'}{\lambda}+\varphi\right) \tag{4.24}$$

  where

$$x' = x\cos\theta + y\sin\theta, \quad y' = -x\sin\theta + y\cos\theta$$

  and λ is the wavelength of the cosine factor of the Gabor filter kernel, θ is the orientation, φ is the phase offset, γ is the aspect ratio and δ is the squared deviation of the Gaussian function. A set of different θ and λ lead to the 8 gabor filters used in our experiments which are shown in Fig 17.

Table 2: Patch level and connected component level features

Patch Feature	**Description**	**# of Features**
Location	Relative location of patch.	2
Width and height	Relative width and height of patch.	2
Foreground density	# of foreground pixels divided by the size of the patch.	1
Average stroke width	# of foreground pixels divided by the length of the contour.	1
Crossing number	# of pixels whose intensity differs from its neighbor.	2
Variance of projection	Variance of horizontal and vertical projection.	2
Maximum run length	Maximum runlength within the patch in the horizontal and vertical directions.	2
CC Feature	**Description**	**# of Features**
Components number	# of CCs within a patch.	1
Maximum width and height	Maximum width and height of a connected component	2
Mean of width and height	Average width and height of components within the patch.	2
Variance of width and height	Width and height variation of components within the patch.	2
Hole ratio	Total hole area within patch divided by patch's size.	1
Overlap ratio	Overlap area between CCs divided by the size of the patch.	1

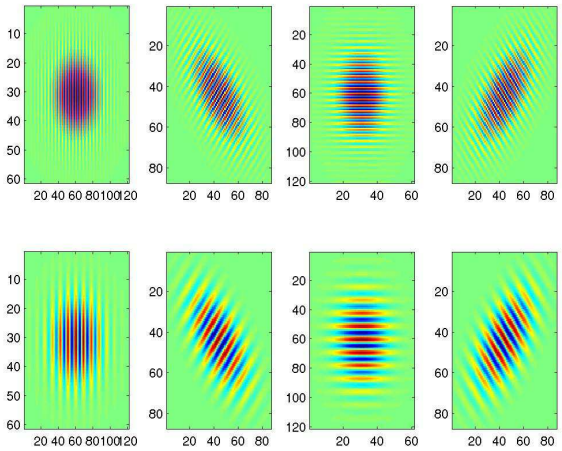


Figure 17: A set of Gabor filters

4.3 G-means Based Classification

Training on the three different kinds of patches is carried out using a modified K-means clustering algorithm known as G-means [57].

Unlike normal K-means which is widely used in clustering methods but where k has to be determined in advance, G-means estimates the number of clusters based on the distribution of the training data. The underlying principle of G-Means is to split the training set using K-means with $k = 2$, and if any sub-cluster dose not have a Gaussian distribution, then K-means with $k = 2$ is applied again for this sub-cluster until each cluster has a Gaussian distribution. Further details of G-means can be found in following figure and [57].

In our training phase, we run G-means clustering algorithm for machine printed patches, handwriting patches and overlapped patches individually to get three

G-Means Procedure

1: Initialize data set as a cluster $C_i = \{x | x \in class(i)\}$.

2: Project samples within cluster onto an optimal projection direction v to get corresponding one-dimensional data set $\hat{C}_i = \{y | y = \langle x, v \rangle / \|v\|^2\}$.

3: Estimate the confidence of statistic A to determine cluster's distribution using Anderson-Darling test [62]:

$$A^2 = -n - \sum_{k=1}^{n} \frac{2k-1}{n} [\ln G(y_k) + \ln(1 - G(y_{n+1-k}))]$$

where n is the size of the data set, y_k is the sorted sample from $\hat{C}_i$ and $G(x)$ is the normal distribution function.

4: If $A < \alpha$, where α is a pre-defined threshold, C_i is regarded as a Gaussian-like distribution and is stored as a qualified cluster with its center $\overline{c}_i = \sum x_k / n$. Otherwise, data set is split into 2 clusters C_{i+1} and C_{i+2} using normal K-Means clustering.

5: For each new cluster, go to step 2, until every cluster has a Gaussian-like distribution.

sets of clusters and corresponding centers $\Omega = \{c_1^1, c_2^1, \cdots, c_l^1 \cup c_1^2, c_2^2, \cdots, c_m^2 \cup c_1^3, c_2^3, \cdots, c_n^3\}$, where c^1 are the centers for machine printed patches, c^2 are the centers for handwriting patches and c^3 represents centers for overlapped patches. The total number of centers N satisfies $N = l + m + n$.

For each cluster, we calculate the co-variance:

$$\Sigma_i = \frac{1}{M} \sum_{k=1}^{M} (y_k - c_i)(y_k - c_i)^T \quad (c_i \in \Omega) \tag{4.25}$$

where M is the size of the cluster which indicates the number of feature points within the cluster, y_k is the sample belonging to this cluster and c_i is the center of the cluster.

During the classification phase, for each test feature point y which is an observation in the MRF framework, we define $L(\cdot)$ to be a function that calculates the cluster nearest to this feature point using Mahalanobis distance metric and maps the test feature point to the index of this cluster's center such that:

$$\begin{aligned} L(y) &= \arg\min_i D_m(y, c_i) \\ &= \arg\min_i \sqrt{(y-c_i)^T \Sigma_i^{-1} (y-c_i)} \quad (0 \leqslant i < N, c_i \in \Omega) \end{aligned} \tag{4.26}$$

This G-means based classification can be looked at as a nearest neighbor search (NNS) with a reduced search space. The label of the test feature point can be further mapped to one of the three classes because the class to which each center belongs (handwritten text, machine printed text or overlapped text) is already known during the training phase.

From the viewpoint of the MRF, our G-means based initial classification is a vector quantization (VQ) procedure which constructs the configuration set Ω for the MRF as described in section 2.1.

For convenience, we use the terminology *index* i of center c_i and *label* L_i interchangeably in the following sections.

4.4 Weighted Features

The individual predictive power of each feature is different and each feature varies in its influence on classification. Therefore, the features that have better discriminating ability should have a greater weight than other features. An easy approach

to evaluate the classification performance of each feature is to use a single feature with a simple classifier such as a Fisher classifier or a Nearest Neighbor classifier and distinguish them by their individual classification scores. Unfortunately, feature ranking based on individual classification score ignores any redundancy amongst the features. A feature selection algorithm based on conditional mutual information [63] can be used to weight features while accounting for any feature dependency. In this section, we will explore the method of feature ranking and modify our Mahalanobis distance measure of Equation 4.26.

Unlike mutual information whose value is an estimate of information shared between random variables X and Y:

$$I(X,Y) = H(X) - H(X|Y) \tag{4.27}$$

where $H(X)$ is the entropy of the random variable X and $H(X|Y)$ is the conditional entropy of the random variable X when Y is known. Conditional mutual information is defined as:

$$I(X,Y|Z) = H(X|Z) - H(X|Y,Z) \tag{4.28}$$

and represents the information shared between random variables X and Y when Z is known.

Then the mutual information gained by a feature f_i when a number of features f^n are already selected is:

$$I(X_i,T|X^n) - I(T|X^n) \tag{4.29}$$

where X_i is the classification test result by using a single feature f_i, T is the target

result (true class label) and X^n are classification results using selected features f^n.

So in the feature selection step, a feature which can provide more mutual information and less dependency on other selected features is the best candidate:

$$f = \arg\min_{f_i} (I(X_i, T|X^n) - I(T|X^n)) \tag{4.30}$$

In order to avoid the calculation of joint probability or joint entropy in Eq.4.30, a pair-wise probability is estimated and the previous feature selection step can be restated as: a feature whose minimum conditional mutual information gained with respect to each selected feature maximizes Eq.4.29 is the best candidate feature:

$$f = \arg\min_{f_i} \left(\min_{f_j \in f^n} I(X_i, T|X_j) - I(T|X_j) \right) \tag{4.31}$$

In our experiments, the predictive power or weight of individual feature f_i is denoted as the conditional mutual information gained by the feature and it is calculated between each pair of classes. Then the overall weight of each feature over three classes is:

$$w_i = P(m \cup h) w_{m,h} + P(m \cup n) w_{m,n} + P(h \cup n) w_{h,n} \tag{4.32}$$

where $P(m \cup h)$, $P(m \cup n)$ and $P(h \cup n)$ are prior probabilities of machine-printed and handwriting patches, machine-printed and noise patches, and handwriting and noise patches respectively. $w_{m,h}$, $w_{m,n}$ and $w_{h,n}$ are weights of feature f_i with respect to each pair of classes.

To classify test patches, weighted Mahalanobis distance from a patch feature point to each cluster center is calculated and this patch feature point is assigned to the closest center (label):

$$L(y) = \arg\min_i D_m(y, c_i)$$
$$= \arg\min_i \sqrt{[w \bullet (y - c_i)]^T [W \bullet \Sigma_i]^- 1 [w \bullet (y - c_i)]} \quad (0 \leq i \leq N, c_i \in \Omega) \tag{4.33}$$

where $W = ww^T$ and w is the weight vector of 29 features obtained from section 4.2.2. The experimental result of the using of weighted features is reported in [64]

4.5 MRF Based Relabeling

In practice, misclassification cannot be avoided using a single classifier due to overlaps in feature space. Therefore, postprocessing or relabeling is needed. The intuition for relabeling is that a patch surrounded by patches from a single different class has a high probability of belonging to that class. We use a Markov Random Field that describes the statistical dependency between observed patch features and their hidden states (labels) to model different patches in an annotated machine printed document and then relabel the patches in our scenario.

In module I of the system where we do patch level text separation, we model the annotated machine printed document I as the graph $\mathcal{G}$ defined in section 2.1. Each vertex in set $\mathcal{V}$ represents a patch within the document and the set of edges $\mathcal{E}$ connect vertices based on a neighbor system described in section 4.5.1. The random field x defined in section 2.1 is consistent with the graph $\mathcal{G}$ and we let

the set of centers Ω which is extracted using G-means from section 4.3 to be the set of all possible labels of x, and the patch features be the observations y.

As described in the previous section 2.2, the inference of the MRF model is implemented by using belief propagation algorithm which is shown in Fig. 7. Each grey node x_i in the hidden layer exclusively corresponds to a hidden configuration for a document patch and is assigned to a label $L_i(0 \leqslant L_i < N)$ after initial classification. Each white node y_i is a observed feature point for that patch. In real documents, the neighbor system of patches is determined by a Gaussian-like distance metric as defined in section 4.5.1 but may not necessarily be located as a grid as shown in Fig.7.

4.5.1 Definition of Neighbor System

As shown in Equation 2.7 and 2.8, we need to update the message and belief in the network based on a neighbor system to compute the optimal configuration for Markov Random Field x. Normally, the neighbor system of a Markov Random Field is based on a 4-neighbors lattice connectivity. However, if we consider each patch in the document as a single vertex in the graph G which may not be rigidly located as a grid, we need to define a flexible neighbor system to represent the spatial relationship between patches.

Firstly, we define a Gaussian-like distance metric to measure the spatial distance between each pair of patches in a document:

$$D_n(i,j) = \frac{(dx_{i,j} - \hat{x})^2}{2\hat{x}^2} + \frac{(dy_{i,j} - \hat{y})^2}{2\hat{y}^2} \tag{4.34}$$

where $[dx_{i,j}, dy_{i,j}]$ represent the convex-hull distance between patches i and j in the horizontal and vertical directions, $\hat{x}$ is the dominant gap between words and $\hat{y}$ is the dominant gap between text lines over the entire document. Dominant gaps $\hat{x}$ and $\hat{y}$ can be estimated using histograms. Based on spatial distance, the four closest neighbors are considered for each patch. The bottom part of Fig.18 shows the four nearest neighbors (which are represented by the four black rectangles) of the patch contained in the red rectangle.

By using Gaussian-like distance metric based neighbor system, we can measure the similarity between patches in spatial space by taking the distance as the variable of an exponential function used in our MRF model. In other words, the similarity to a given patch decreases exponentially like a Gaussian function as the distance increases and only patches which have the greatest similarity are considered as the neighbors of the center patch. The plot at the top of Fig. 18 shows the decrease of similarity with distance from the center patch.

4.5.2 Prior $P(x)$ and Likelihood $P(y|x)$

In order to compute an optimal configuration which maximizes the posterior as described in Equation 2.3, we use belief propagation which calculates the local

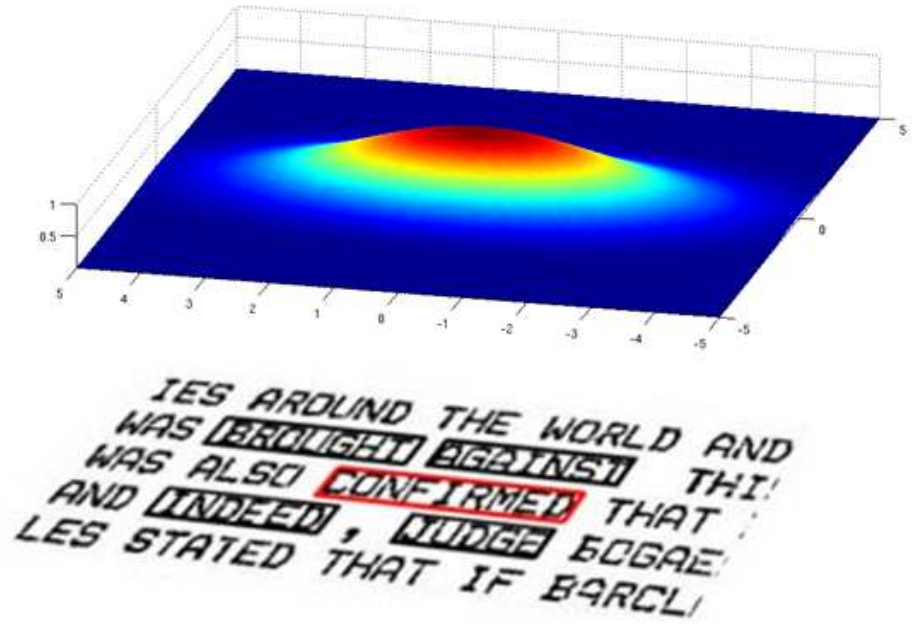


Figure 18: A patch and its nearest neighbors.

maximum messages and beliefs for each vertex to achieve global maximum. To use belief propagation for a given vertex $\mathcal{V}_i$, the prior $P(x_i|x_j)$ and likelihood $P(x_i|y_i)$ in Equation 2.7 and 2.8 are loosely replaced by similarity function $\Psi_h(x_i,x_j)$ (which measures the similarity of configuration between hidden node x_i and x_j) and dependency function $\Psi_o(x_i,y_i)$ (which describes the influence of the observation y_i on its hidden node x_i).

The similarity function is defined as:

$$\Psi_h(x_i,x_j) = 1 + \alpha e^{-D_n(i,j)} + \beta e^{-D_e(L_i,L_j)} \tag{4.35}$$

where $D_n(i,j)$ is the Gaussian-like distance between two neighboring patches (which corresponds to vertex $\mathcal{V}_i$ and $\mathcal{V}_j$ in graph $\mathcal{G}$) calculated from Equation 4.34 and

$D_e(L_i, L_j)$ is the Euclidean distance between the configurations of two patches which represent the assigned centers in the feature space as described in section 4.3. Fig. 19 shows the idea behind Equation 4.35. α and β are two parameters that control the influence from neighbors in the spatial space and the feature space respectively.

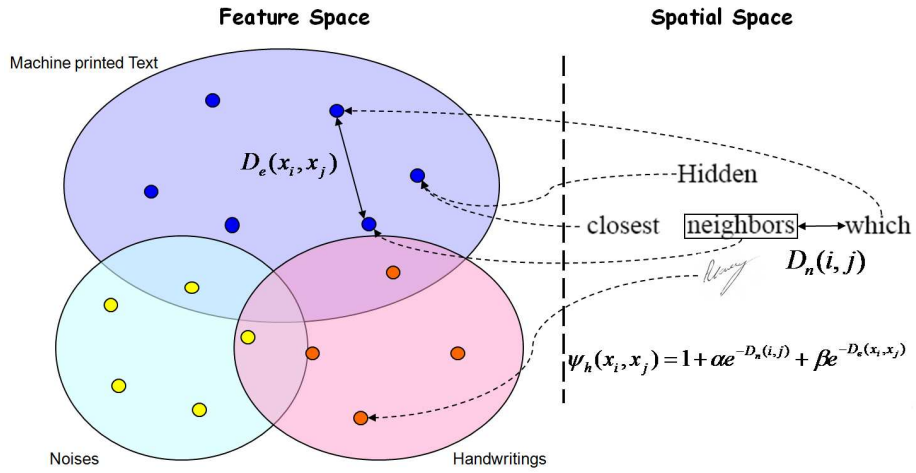


Figure 19: Distance in feature space and spatial space.

Equation 4.35 illustrates that the influence from the neighbors of a patch depends not only on their spatial distance but also on the distance of the two configurations in feature space. So, in our system, belief propagation is encouraged for those patches which are close to each other in both the spatial as well as the feature space.

Similarly, the dependency function $\Psi_o(x_i, y_i)$ is defined as:

$$\Psi_o(x_i, y_i) = e^{1/(\lambda D_m(c_i, o_i))} \tag{4.36}$$

where $D_m(c_i, o_i)$ is the Mahalanobis distance calculated from Equation 4.26, which measures the distance from the observation o_i of vertex $\mathcal{V}_i$ to its assigned center c_i (corresponding to a configuration) in feature space. Parameter λ controls the influence of observations and their hidden states.

Equation 4.36 shows the relationship between the configuration of a given patch and its corresponding observed features.

Optimal configuration for MRF is achieved by applying BP algorithm as shown in Equations 2.7 and 2.8. All messages $m_{i,j}$ in these two equations are initially set to be 1 and updating is terminated when there is no flipping occurring during belief propagation or when maximum iteration is achieved.

4.6 Experimental Results for Module I - Word Level Text Separation

We used two different data sets to train and test the proposed word level text separation system in our experiments. To estimate the performance of MRF based relabeling system, we use the *Precision (P)* and *recall (R)* metric in this chapter to compare proposed method with other classifiers.

$$P(i) = \frac{\#\ of\ patches\ correctly\ classified\ as\ class\ i}{\#\ of\ patches\ classified\ as\ class\ i} \tag{4.37}$$

$$R(i) = \frac{\#\ of\ patches\ correctly\ classified\ as\ class\ i}{\#\ of\ patches\ belonging\ to\ class\ i} \tag{4.38}$$

4.6.1 Tobacco Data Set

The first data set contains 94 binarized documents of the Tobacco industrial litigation archives from which we extracted patches using a 5×5 window based morphology closing and segment algorithm as described section 4.2.1.

To this data set, a total number of 19842 machine printed patches, 832 handwritten patches and 9011 noise patches were extracted. We randomly selected 48 documents which contain 15409 patches for training purpose and the remaining documents which have 14276 patches for testing.

The G-Means based clustering algorithm described in section 4.3 was used to extract 117 centers for machine printed patches, 5 centers for handwritten patches and 53 centers for noise patches on training data set and to construct our configuration set x.

The initial classification was carried out by assigning each test patch to the nearest center in the feature space and mapping them to one of three classes finally. The Markov Random Field based relabeling procedure was used to correct the misclassification for all patches as described in section 4.5.

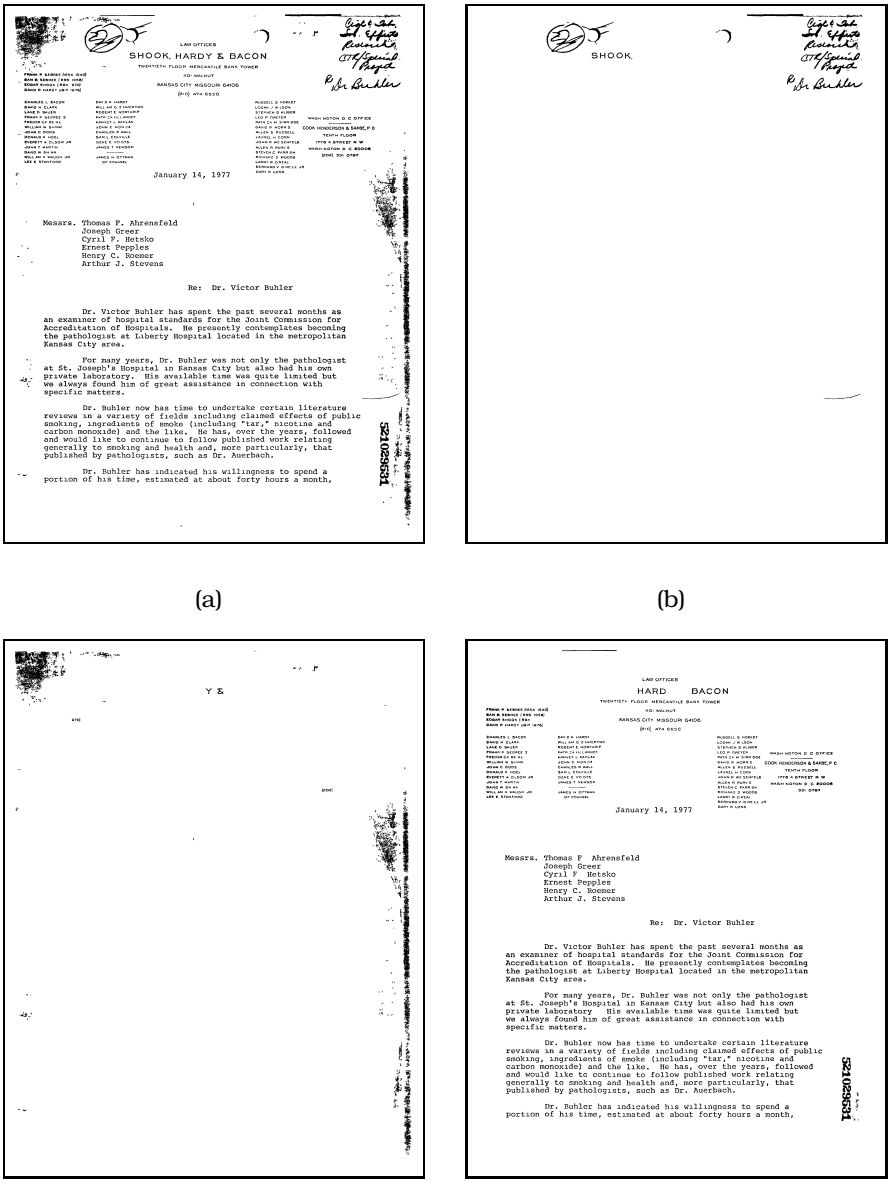
In table 3, we compared the precision and recall of using G-Means based classification and MRF based relabeling algorithm. From this table, we can see that both precision and recall increased for each class after using MRF based relabeling procedure, especially for handwritten patches. The overall recall increased around 3% in our system.

Table 3: The analysis of system performance for tobacco set

	G-Means		Proposed method	
	Precision	Recall	Precision	Recall
Machine-printed	98.91%	94.11%	99.20%	96.59%
Handwritten	47.97%	90.91%	64.99%	96.01%
Noise	92.40%	95.19%	94.84%	96.40%
Overall	N/A	93.40%	N/A	96.33%

Fig. 20 shows an example of text identification result from Tobacco data set where we decomposed the original binarized document (Fig. 20a) into handwritten document(Fig. 20b), noise document (Fig. 20c) and machine printed document (Fig. 20d) respectively.

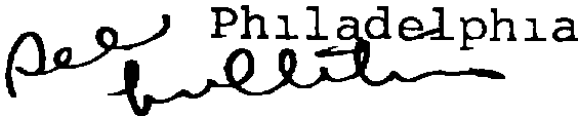
In the Tobacco data set, the misclassification was mainly from around 20 overlapped patches which contained both handwritten and machine printed text and over 300 patches whose resolution was very low as shown in Fig. 21. The overlapped patches were excluded from our evaluation since it contained two different classes in the same patch and should be considered as a separated class. Normally, the low resolution of machine printed patches caused low precision metric for the handwritten text in our system because they were easily classified as handwritten patches. Using more such training samples can solve this kind of problem.



(a) (b)

(c) (d)

Figure 20: A Tobacco data set example of text identification results from the system. (a) The original binarized document, (b) handwritten text, (c) noises, and (d) machine printed text.



(a)

(b)

Figure 21: Overlapped text and low resolution text

4.6.2 HP Labs Data Set

We extended our experiment on the HP Labs data set consisting of binarized images of annotated office documents scanned at a resolution of 300 dpi. Unlike Tobacco data set which has lots of noises, HP Labs data set has three different classes (machine printed text, overlapping text and handwritten text). We have collected 82 documents from the HP Labs data set. This extremely imbalanced data set contains over 25000 machine printed text patches, around 3200 handwritten text patches and less than 500 overlapped text patches. We selected 54 documents for training and 28 documents for testing.

Similarly to the previous experiment on Tobacco data set, prior to classification, morphology closing operation, patch extraction and feature extraction algorithms were applied as the preprocessing step. Initial classification was carried out by using G-means which was followed by a MRF based relabeling procedure.

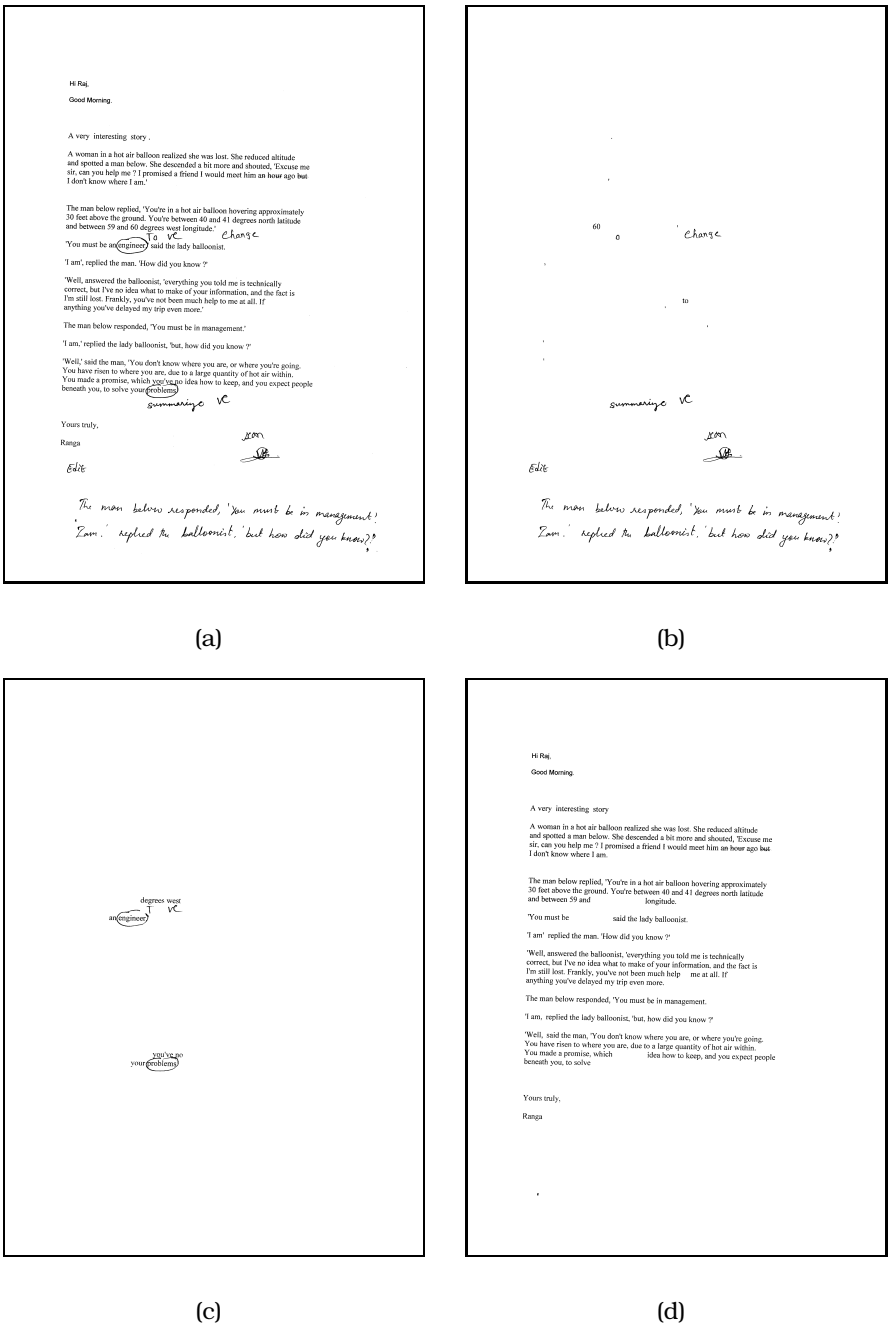
In Table 4, we compared the proposed method to a G-means based classifier

Table 4: The analysis of system performance for HP set

	BP Neural network		G-Means		Proposed method	
	Precision (%)	Recall (%)	Precision (%)	Recall (%)	Precision (%)	Recall (%)
Machine-printed	99.29	95.07	99.67	94.52	99.70	96.07
Handwritten	77.48	89.94	72.17	92.53	80.46	93.98
Overlapped	33.54	77.48	39.85	81.82	42.69	84.09
Overall	N/A	87.50	N/A	89.62	N/A	91.38

and a backpropagation Neural Network classifier which used a modified public ANN tool, FAAN. We see that our MRF based relabeling system had an overall recall of 91.38% which outperformed backpropagation Neural Network (87.5%) and G-means based classifier (89.62%) and slightly increased the precision, especially for two minor classes (handwritten text and overlapped text). Fig. 22 shows the three decomposed documents (Fig. 22d, Fig. 22c, Fig. 22d) from original documents (Fig. 22a) using proposed MRF based classification method.

The relative low precision for handwritten text and overlapped text is mainly due to the imbalanced property of HP Labs data set. The amount of machine printed text is much more than the handwritten annotations and overlapped text and dominates in the data set. If even a small proportion of machine printed text is misclassified as the other two classes, it is still a significant number comparison to handwritten samples and leads to low precision. To overcome the imbalanced data set problem, we suggest using a tree-structured initial classifier which is described in the following chapter.



(a) (b)

(c) (d)

Figure 22: A HP data set example of text identification results from the system. (a) The original binarized document, (b) handwritten text, (c) overlapped text, and (d) machine printed text.

4.7 Conclusions

In this chapter, we propose a novel Markov Random Field based algorithm to classifier three different kinds of text (machine printed text, handwritten text and overlapped/noise text). A Gaussian-like function is defined to measure the spatial distance which is used to construct the neighborhood system for MRF. Unlike other relabeling systems only consider the neighboring relationship in spatial space, our model uses distances from both feature space and spatial space to determine the similarity of two neighbors. The merit of proposed method is that it is easily to integrate other classifiers which can provide reliable distance measure in feature space into our system as an initial classifier. From the experiment on two different data set, it shows our method can improve the accuracy significantly.

Chapter 5

Boost Tree Classifier

5.1 Introduction

In our previous chapter, which separates the word level text into machine printed text, handwritten text, overlapped text or noises, one of the problem we encountered is the imbalanced data set problem. In the case of annotated documents, the imbalance between handwritten and machine printed text is a serious problem. Generally, the amount of machine printed text is much more than the handwritten annotations and will dominate in the data set. If even a small proportion of machine printed text is misclassified as handwritten text, it is still a significant number in comparison to handwritten samples and leads to relatively low precision [1] for identification of handwritten annotations. To overcome the imbalanced data set issue in a form classification problem, a tree-based classification algorithm was introduced in [65], which proposed a hierarchical classification model

to classify different tax forms using binary Regularized Least Square classifiers and K-nearest-neighbor classifiers in each tree node. Other boost tree classifiers that are attracting research interest in the pattern recognition and computer vision communities can be extended to the field of document analysis as well [66, 67].

The imbalanced data set problem is amplified even further when we consider annotations that overlap machine printed text. In order to classify machine printed text, handwritten text and overlapped text and overcome the imbalanced data set problem, we describe a tree-structured classifier which consists of binary weak classifiers. At each node, we convert a multi-class problem to a bi-class problem by merging all classes except the major class to a single class. Over-fitting is a potential drawback of a tree-structured classifier because it considers only a subset of training data at each node and loses the general distribution of the whole data set [68]. Inspired by the Adaboost algorithm which uses and updates the distribution of all training data in each round and is less subject to over-fitting , we propose a new mechanism to train the weak classifier for each node of the tree-structured classifier which is intended to replace the initial G-Means based classifier.

We organize the remaining of this chapter as the following: section 5.2 introduces the structure of our tree-structured classifier and the procedure used for learning and testing. Section 5.3 shows the details of experiments, including feature extraction, experimental set up and results. Section 5.4 presents our

conclusions.

5.2 Tree-Structured Classifier

The merit of the tree-structured classifier is that it balances the positive and negative samples by merging the classes in the lower level of the tree [65] and its training error can be made to decrease exponentially [68]. Normally, a divide-and-conquer strategy is used to build the tree. The underlying principle of this divide-and-conquer approach is that if current classifier cannot separate the training set with high accuracy, the set is separated into several smaller clusters and classifiers (may differ from previous classifier) are used for each new cluster. Thus, a tree-structured classifier recursively focuses locally on the data set as the tree grows deeper and can achieve high training accuracy. However, a node of a tree-structured classifier loses distribution information from the entire data set and is very susceptible to over-fitting.

5.2.1 Structure of the classifier

To preserve the advantage of tree-structured classifier and overcome the over-fitting problem, we design a boosting algorithm which has a tree structure and can be also viewed as a combination of cascade classifiers [69].

Assuming that there are n classes and a total number of m samples in the training data set Λ, each sample is associated with a normalized weight w_i which

measures the importance of the sample. Our tree-like classifier's job is to find a hypothesis $H : x_i \to y_i \in \Gamma = \{1, \cdots, n\}$, which maps sample x_i to a target label y_i in label set Γ.

Firstly, we define the structure of the tree-like classifier. Each node $N(i, j)$ of the tree corresponds to a weak classifier $h_{i,j}(\cdot)$, where i indicates the level of the node and j shows the index of the node in this level. Note that even an empty node is assigned an index number in the tree. Then the left child's index of a node $N(i, j)$ is $N(i+1, 2j)$ and the right child's index of node $N(i, j)$ is $N(i+1, 2j+1)$. Each leaf node in the tree corresponds to a target label.

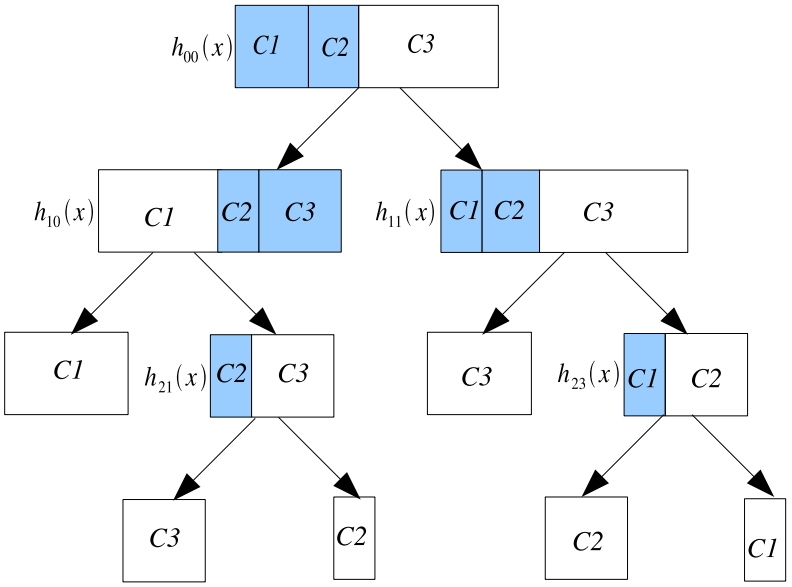


Figure 23: An example of tree-like classifier. In each node, minority classes are merged to a single class and represented as a blue rectangle. Majority class is represented as a white rectangle.

Unlike other algorithms that handle multi-class classification using multi-class weak classifiers at each node or each level, we propose to convert our multi-class classification problem to a two-class classification problem at each node. We will determine which class is the major class, and all other classes are considered as the minor class.

For a given node $N(i,j)$, the summation of the sample weights from each class measures the majority of this node.

$$m(i,j) = \arg\max_{t\in\Gamma} \sum_{k=0,y_k=t}^{m} w_k \tag{5.39}$$

Fig.23 shows an example of the tree structure and how to convert multi-class to bi-class problem.

5.2.2 Inspired from Adaboost

Adaboost which was formulated by Freund and Schapire [70] uses other classifiers as its weak classifier as a cascade to achieve better performance. One merit of Adaboost is it can be less susceptible to the overfitting problem than most learning algorithms.

During the training phase of Adaboost, a weak classifier is called repeatedly in total of T rounds. On each round, the distribution or weight of training samples D_t is updated according to the classification test. The algorithm of Adaboost for bi-classification can be briefly described as in Fig. 24 and detailed in [71, 72, 73]:

Algorithm of Adaboost

Input: Total number of n training samples $x_0, x_1, \cdots, x_n$ with their targets. $y_0, y_1, \cdots, y_n$ where $x_i \in \Lambda$ and $y_i \in \Gamma = \{-1, +1\}$

Distribution initialization: $D^1(i) = 1/n$

During iteration, for $t = 1, \cdots, T$:

1: Train weak classifier h_t using distribution D^t;

2: Run classification test on $h_t : \Lambda \to \Gamma$ with error $\varepsilon_t = P_{i \sim D_t}[h_t(x_i) \neq y_i]$;

3: Calculate weight updated parameter $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$;

4: Update distribution $D_t : D_t = \frac{D_t(i) exp(-\alpha_t y_t h_t(x_i))}{Z_t}$, where Z_t is a normalization factor;

Output the final classifier: $H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$.

Figure 24: Algorithm of Adaboost

As shown in the third step of Fig.24, the parameter α_t measures the importance of the weak classifier h_t where smaller error ε_t causes larger value of α_t. The distribution D_t is updated according to the classification test of h_t where the weight $D_t(x_i)$ is increased if x_i is misclassified by h_t. It is easy to observe that the final classifier of Adaboost is a weighted T weak classifiers which focuses more on harder samples.

It was proved by Freund and Schapire [74, 71] that the training error rate of

Adaboost is bounded by:

$$\frac{1}{n}\sum_i [H(x_i) \neq y_i] \leq \frac{1}{n}\sum_i exp(-y_i h(x_i))$$
$$= \prod_t Z_t \tag{5.40}$$

and the final classifier $H(x)$ can achieve an arbitrarily low error rate by repeatedly using a weak classifier with sufficient training data. Although it is shown that boost algorithm will overfit as T becomes large, in real applications, Adaboost often does *not* suffer from overfitting even with large T. In [72], Freund and Schapire show the strong connection between Adaboost and SVM (support vector machine).

Based on the merits of properties described above, Adaboost gained great success in machine learning, pattern recognition and computer vision areas. For example, Viola and Jones used Adaboost to build a feature extractor and face detector in [75]. Based on boosted decision stumps, Torralba et al. proposed an algorithm to detect multiclass and multiview objects [76]. To detect face with rotation invariant, Huang et al developed a vector boosting algorithm which is inherited from Adaboost [77, 78].

To extend adaboost from bi-class problem to multi-class problem, Schapire and Singer converted multi-class problem to several binary problems and then used the general Adaboost algorithm [71]. In [74], Freund and Schapire proposed a multi-class adaboost method which can be considered as a special case of Schapire's method. To improve the classification performance of imbalanced

data involving multiple classes, Sun et al. developed a cost-sensitive boosting algorithm in [79].

Considered decision tree as a special boosting mechanism [80], researchers tend to combine Adaboost with decision tree to overcome the imbalanced data set problem and over-fitting problem. Grossmann suggested an algorithm to boost a weak classifier into a tree which is called AdaTree in [68]. In [67], Tu proposed a decision tree classifier where each node of the tree is adaboost. Similarly, Wu and Nevatia proposed a tree-structured classifier for multi-class detection where each path of the tree is a boosted classifier [66].

To inherit the merits from Adaboost which is less susceptible to over-fitting, we suggest a boosting mechanism to construct a tree-structured classifier which is described in the following section.

5.2.3 Learning & Testing Procedure

The normal recursive learning procedure is to split the source set of the parent node into subsets for child nodes based on the parent node's classification test. Given the training set $\hat{\chi}$ for a parent node $N(i,j)$, each child node has a subset:

$$N(i+1,2j) : x \in \hat{\chi}^{+} \mid h_{i,j}(x) < t$$

$$N(i+1,2j+1) : x \in \hat{\chi}^{-} \mid h_{i,j}(x) > t$$

where t is a threshold for weak classifier $h_{i,j}(x)$ and $\hat{\chi} = \hat{\chi}^{+} \cup \hat{\chi}^{-}$ is satisfied.

As mentioned earlier, this splitting method is prone to over-fitting. Since Adaboost is less susceptible to over-fitting, we integrate a cascade decision mechanism which is similar to Adaboost's boosting algorithm into the tree-structured classifier to overcome the over-fitting problem. As noted in [72], the effect of updating the distribution of training data in Adaboost algorithm is to increase the weight of examples misclassified by h_t, and to decrease the weight of correctly classified examples. Thus, the weight tends to concentrate on "hard" examples.

So, rather than split the training data set and assign a subset to a child node, we propose to use all the training data at each node during recursive learning but assign different weights to them according to their attributes.

For a given parent node $N(i,j)$, we train the weak classifier $h_{i,j}(\cdot)$ using weighted training samples $\hat{\chi}$ with their weights $\hat{w}$ and calculate the threshold t for this node.

The weight of training samples at the left child node $N(i+1,2j)$ is updated as:

$$w_i = \begin{cases} \hat{w}_i & \mid h_{i,j}(x_i) < t \\ \alpha \cdot \hat{w}_i \cdot exp\{-(\frac{h_{i,j}(x_i)-t}{o_{max}-h_{i,j}(x_i)})^2\} & \mid h_{i,j}(x_i) > t \end{cases} \tag{5.41}$$

Similarly, the weight of training samples for the right child node is updated as:

$$w_i = \begin{cases} \hat{w}_i & \mid h_{i,j}(x_i) > t \\ \alpha \cdot \hat{w}_i \cdot exp\{-(\frac{t-h_{i,j}(x_i)}{h_{i,j}(x_i)-o_{min}})^2\} & \mid h_{i,j}(x_i) < t \end{cases} \tag{5.42}$$

where o_{max} and o_{min} are maximum and minimum output of $h_{i,j}(\cdot)$ of the parent node given the training samples. The weight for all training samples are initially set to be $\frac{1}{m}$ in our experiment.

Fig.25 shows an example of splitting a parent node to two child nodes. Initially, a weak classifier is trained using all training samples with the same weights as shown in the upper figure. The distances of misclassified training samples to decision boundary are calculated based on the classification test. Both child nodes inherit all training samples from the parent node but with different weights associated with them. As shown in the figure on the bottom-left, the misclassified yellow samples in the left child node gain more weights while all other samples remain the same weights. These updated samples tend to force decision boundary to move leftwards. Similarly, the updated green samples in the right node force the decision boundary to move rightwards as shown in the figure on the bottom-right.

The purity p of the node is used as our stopping criteria for each node $N(i,j)$ and updated according to the level of the node to avoid over-fitting as described in following equation:

$$\sum_{k=0,y_k=m(i,j)}^{m} w_k > p \cdot e^{-\frac{i}{\lambda}} \tag{5.43}$$

where i is the level of the node $N(i,j)$, p is a pre-defined constant purity threshold (may differ for different classes), λ is the parameter to control the convergence of training procedure and $m(i,j)$ is the majority of current node as defined in Equation 5.39. Equation 5.43 illustrates that if the ratio of majority is higher than the purity threshold, a leaf is achieved and its majority id is assigned as the target label of this leaf node.

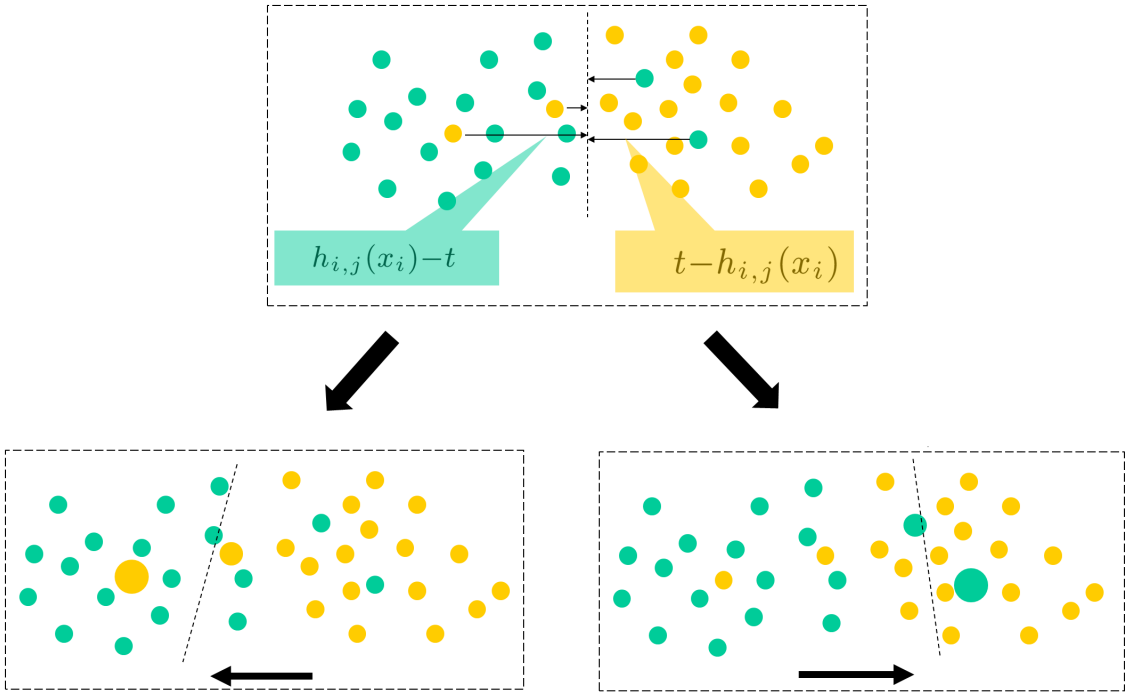


Figure 25: An example of splitting a node in the tree.

The testing procedure is similar to the training phase which starts from the root node and achieves one of the child nodes according to the classification test at each parent node. The truth label of the test sample is assigned as the label of the leaf node which is achieved.

5.3 Experiments

The data used for the experiments is a set of 82 annotated office documents from the HP Labs data set which consists of binarized images scanned at a resolution of 300 dpi. This extremely imbalanced data set contains over 25000 machine

printed text patches, about 3200 handwritten text patches and less than 400 overlapped text patches. We used 54 documents for training and the remaining for testing.

Prior to classification, a morphology closing operation was applied to merge small characters to patches which approximately represent words. These patches are the basic unit in our system. At each patch, we extracted patch level features, connected component features and Gabor features as described in [58]. The tree-structured classifier was built as described in Section 5.2. A modified public ANN tool, FANN [81], was used in our experiments using one hidden layer to train the binary weak learner for each node. The test samples were labeled using this classifier in the same manner.

We measured the performance of the proposed system using precision and recall metrics. Precision for machine printed text in our system is the ratio of patches which are correctly classified as machine printed text to all patches which are classified as machine print. Recall for machine printed text is the ratio of patches which are correctly classified as machine printed text to all machine printed patches in the test set. The same metrics were applied to handwritten text and overlapped text.

In table 5, we compared the proposed tree-structured classifier to a normal decision tree classifier which has a similar training phase but without using updated distribution for training samples, and a backpropagation Neural Network

Table 5: The analysis of performance of tree-structured classifier

	BP Neural network		Normal DT		Proposed method	
	Precision (%)	Recall (%)	Precision (%)	Recall (%)	Precision (%)	Recall (%)
Machine-printed	99.29	95.07	99.40	97.84	99.61	98.74
Handwritten	77.48	89.94	91.04	90.66	92.89	93.67
Overlapped	33.54	77.48	40.61	80.30	57.29	83.33
Overall	N/A	87.50	N/A	89.60	N/A	91.91

classifier which is implemented using FANN. We see that the Neural Network cannot identify handwritten text and overlapped text very well on this imbalanced set because it tends to focus on the majority class (machine printed text) and loses information about the minority classes (handwritten text and overlapped text). Normal decision tree can achieve slightly better identification performance because as the tree grows deeper, the possibility of focusing on minority classes increases as well. However, our proposed tree-structured classifier had an overall recall of 91.91% which outperformed backpropagation neural network (87.5%) and DT (89.6%) and also significantly increased the precision, especially for overlapped text.

The mis-classification in our system is mainly from machine printed symbols and punctuation marks such as comma and period which are classified as handwritten dots. The imbalanced data set problem which causes the low precision of overlapped text can be overcome by adding more such samples into the training set.

5.4 Conclusions and future work

In this chapter, we present a tree-structured classifier that can take advantage of the global distribution of training samples. The distributions (weights) of training samples are updated based on the classification test at each node during the training phase. Experiments show that the proposed method is more reliable for extremely imbalanced data sets when compared to normal decision tree and other classifiers.

The remaining problem for our proposed boost tree classifier is that although the classifier divides the entire feature space to several small subsets where each leaf is corresponding to one of subsets and provides better initial classification performance than normal decision tree and G-Means based classifier, it only outputs the label of each test sample without the probability estimation which is needed by our MRF based re-labeling framework. Our future work will focus on developing a method to calculate probability or ranking of the boost tree and integrating the tree-structured classifier into MRF.

Chapter 6

Overlapped Text Separation

As shown in Fig. 13, the second module of our text separation system takes as its input the overlapped text from module I. Since the overlapped text is considered as a single unit in module I, its basic element (patch) and the features for patches are not suitable to further segment the overlapped text into machine printed and handwritten text. In this chapter, we propose a method to decompose the overlapped text image into smaller units using a coarsening procedure which aggregates foreground pixels of the overlapped text image according to their coherence properties [82]. Shape context, a pixel level feature, is used to measure the coherence or variance between pixels or aggregations. The MRF based segmentation of overlapped text is based on these small aggregations which are the basic element in module II.

6.1 Shape Context Features

Prior to coarsening pixels to aggregations, we first extract features at each pixel. We use shape context features to characterize each pixel or aggregation. Shape context features were first used by Belongie et al. [83] to compare the similarity between two shapes.

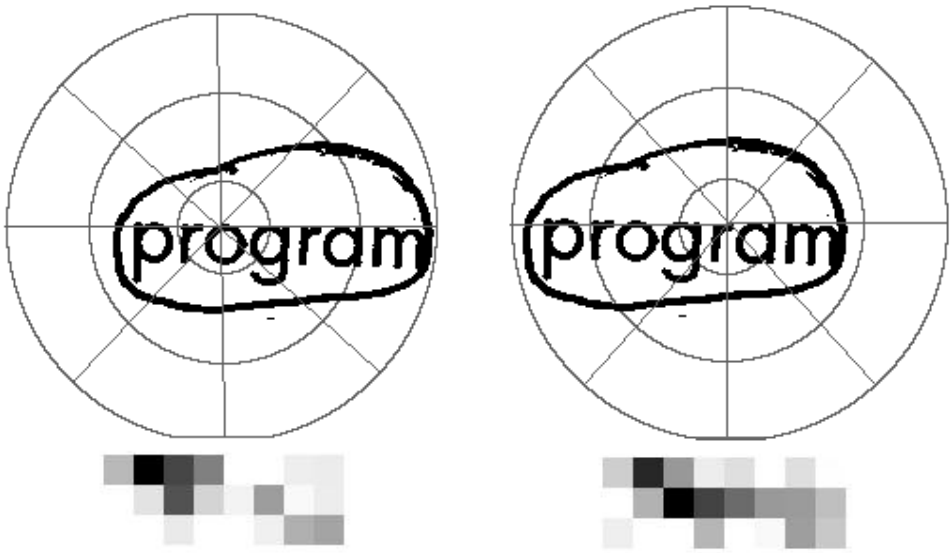
Given a shape which contains a set of points, we can draw vectors from a given point on this shape to each of the other points and the shape can be exactly represented by these vectors. Shape context features were inspired by this notion but instead of using vectors to characterize each point on the shape, shape context features tend to capture the coarse distribution of the rest of the points on a shape with respect to a given point.

To compute the shape context feature $\overline{v}_i$ for a pixel p_i, a polar system is centered on this pixel to calculate a histogram h_i of the remaining pixels on the shape using Equation 6.44:

$$h_i(k) = \#\{q \neq p_i \cap q \in bin(k)\} \tag{6.44}$$

where $h_i(k)$ is the kth element of histogram which counts the number of foreground pixels falling into the kth bin of the polar system and $\overline{v}_i = [h_i(0), h_i(1), \cdots, h_i(n)]$, n is the number of bins in the polar system.

Fig. 26 illustrates two examples of computing histograms for two different points on the same shape by centering the polar coordinate system on the points. The bottom part of each sub-figure shows the extracted shape context features



(a) (b)

Figure 26: The polar system and shape context corresponding to two different points.

where each colored rectangle corresponds to a bin in the polar system and darker color represents a larger value. In our experiments, we use 3 circles and 8 quadrants to divide the overlapped text patch into 24 bins as shown in Fig. 26.

6.2 Aggregation Coarsening

In order to separate overlapped text, we propose a coarsening procedure by which foreground pixels in the overlapped text are grouped to aggregations which have coherent attributes and can be used as the basic element for training and classification.

We assume an overlapped text image I is modeled by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each vertex corresponds to an aggregation which contains a set of foreground pixels $\mathcal{P}_i = \{p_{i,1}, p_{i,2}, \cdots, p_{i_m}\}$ from the image and edges $\mathcal{E}$ connect vertices based on 4-neighbors lattice connectivity.

For each vertex $\mathcal{V}_i$, we measure its coherence by calculating the variance δ_i of its corresponding aggregation:

$$\delta_i^2 = \frac{1}{m} \sum_{j=1}^{m} (\overline{v}_{i,j} - \mu_i)^2 \tag{6.45}$$

where m is the size of the aggregation which indicates the number of pixels in it, $\overline{v}_{i,j}$ are the shape context features extracted from section 6.1 for pixel $p_{i,j}$ in vertex $\mathcal{V}_i$ and μ_i is the average of the features for this vertex:

$$\mu_i = \frac{1}{m} \sum_{j=1}^{m} \overline{v}_{i,j} \tag{6.46}$$

Then the diversity between two adjacent vertices $\mathcal{V}_i$ and $\mathcal{V}_j$ is measured by the difference in their average values:

$$\eta_{i,j} = \mu_i - \mu_j \tag{6.47}$$

By using variance δ_i within a vertex and difference $\eta_{i,j}$ between two adjacent vertices, we define a criteria as presented in Equation 6.48 to determine whether to merge two adjacent vertices to a new single vertex (aggregation) or not.

$$\hat{j} = \arg\max_{j \in N(i)} \sum_{k \in N(i) \setminus j} \eta_{i,k} e^{-\delta_{i+j}} \tag{6.48}$$

where $j \in N(i)$ represents the neighbors of vertex $\mathcal{V}_i$, $k \in N(i) \setminus j$ represents all neighbors of the vertex $\mathcal{V}_i$ but excludes vertex $\mathcal{V}_j$, and δ_{i+j} is the variance of the features within the new merged vertex ($\mathcal{V}_i$ merged with $\mathcal{V}_j$).

The underlying principle of Equation 6.48, as shown in Fig.27, is that a vertex $\mathcal{V}_i$ is temporarily merged with its immediate neighbors individually to create a set of candidate vertices $\{\mathcal{V}_{i+j}\}$ whose variances are $\{\delta_{i+j}\}$. Then, the vertex $\hat{\mathcal{V}}_j$ whose average feature value is close to $\mathcal{V}_i$'s average feature value and minimizes the new merged vertex $\mathcal{V}_{i+j}$'s variance is the optimal vertex which should be merged with $\mathcal{V}_i$.

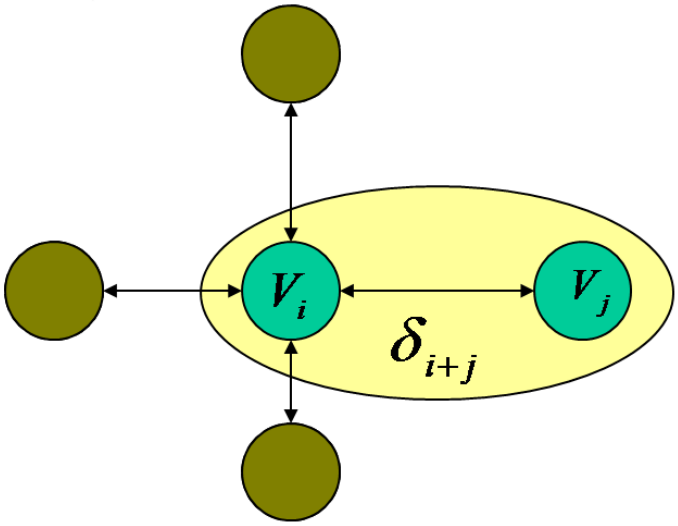


Figure 27: Merge two neighbor nodes.

Based on this criterion, we can extract aggregations starting from a single pixel to produce larger coherent coarsened regions. Fig. 28 shows the procedure of coarsening. Initially, each aggregation for the vertex $\mathcal{V}_i$ contains only one pixel

and all vertices $\mathcal{V}$ in graph $\mathcal{G}$ are pushed in a queue $\mathcal{Q}$. Prior to executing the main loop of the coarsening procedure, the size m of all vertices is checked and the first vertex whose size is smaller than a pre-defined threshold τ is picked as the working vertex. The working vertex $\mathcal{V}_i$ is merged with an optimal immediate neighbor according to Equation 6.48 to produce a new vertex which is stored at the end of queue $\mathcal{Q}$ and its neighbors are inherited from its two parent vertices. The two merging vertices are removed from the queue $\mathcal{Q}$ and the neighbor system for graph $\mathcal{G}$ is also updated accordingly. The iteration of coarsening is stopped when every vertex's size is bigger than threshold τ.

Fig. 29 shows an example of coarsening for an overlapped text image where machine printed text is circled and touched by handwriting. Fig. 29(a) is the original binarized overlapped text image. Fig. 29(b) shows the coarsening result after first iteration when every pixel is coarsened with its neighbor at least once. Fig. 29(c) shows the aggregations after the third round of iteration and Fig. 29(d) is the final coarsening result. The aggregations are randomly colored in these figures.

6.3 MRF Based Classification

6.3.1 Modeling Overlapped Text Using MRF

Markov Random Field model is reused in module II to separate handwritten text from an overlapped text image. As shown in section 2.1, a key step in the usage of

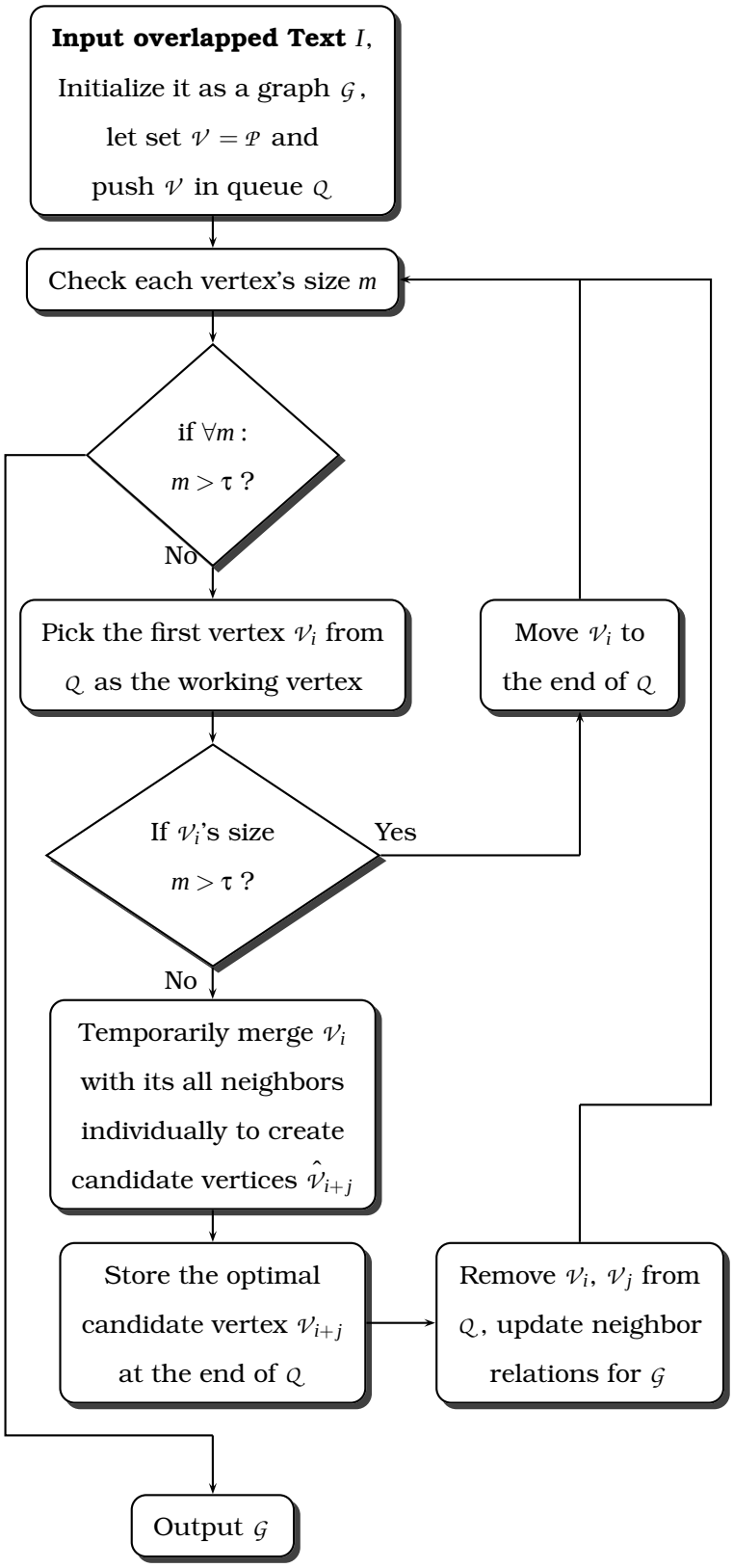


Figure 28: The coarsening procedure

(a) (b)

(c) (d)

Figure 29: Example of coarsening procedure and aggregation

MRF is the definition of the random field x, the hidden configuration set Ω and the observations $\mathcal{Y}$. Unlike the definition of an MRF for the entire document in section 4.5 where each vertex in the MRF corresponds to a text patch, the overlapped text image is modeled as a random field whose nodes represent aggregations extracted using the coarsening algorithm from section 6.2. To build the configuration set Ω, G-means clustering algorithm is used as a vector quantization (VQ) procedure to separate the training set into several small clusters and the center of each cluster is used to build a codebook to represent the configuration set Ω. The observations $\mathcal{Y}$ for each vertex is obtained by normalizing each aggregation using a 16×16 grid. The neighbor system for the MRF of module II is inherited from the coarsening algorithm where the neighbors of each vertex are its direct connected vertices based on the 4-neighbors lattice connectivity relation.

6.3.2 Prior $P(x)$ and Likelihood $P(y|x)$

Similar to module I which uses Belief Propagation (BP) algorithm to obtain the optimal configuration $\overline{x}$ for MRF in section 4.5, our MRF based classification for overlapped text also achieves the optimal configuration based on Equations 2.4 but re-defines the prior $P(x_i|x_j)$ and likelihood $P(y_i|x_i)$.

The prior $P(x_i|x_j)$ which is replaced by a similarity function $\Psi_h(x_i,x_j)$ in our MRF based overlapped text separation module is defined as:

$$\begin{aligned}&\Psi_h(x_i = c^{'}, x_j = c^{''}) \\ &\quad = 1 + m\left\{\alpha f(c^{'},c^{''}) + \beta e^{-d_h(c^{'},c^{''})}\right\}\end{aligned} \tag{6.49}$$

where $c^{'},c^{''} \in \Omega$ are two configurations assigned to vertex $\mathcal{V}_i$ and $\mathcal{V}_j$ respectively, m is the size of vertex $\mathcal{V}_i$ which indicates the number of pixels contained in the corresponding aggregation represented by the vertex $\mathcal{V}_i$. $f(c^{'},c^{''})$ and $d_h(c^{'},c^{''})$ are two functions that count the pairwise occurrence frequency of configurations in the training set (defined in Equation 6.50) and measure the distance between two configurations (corresponding cluster centers) in the feature space (defined in Equation 6.51) respectively. Parameters α and β control the strength of the influence from these two terms on vertex $\mathcal{V}_i$.

The pairwise frequency is defined as:

$$f(c^{'},c^{''}) = \frac{\#\{i \in N(j) | g(y_i) = c^{'}, g(y_j) = c^{''}\}}{\#\{i \in N(j)\}} \tag{6.50}$$

where j runs over all vertices in the graph $\mathcal{G}$, function $g(y_i)$ calculates the corresponding center for vertex $\mathcal{V}_i$ given its observation y_i. The numerator of Equation 6.50 represents the number of neighboring aggregations whose corresponding centers are $c^{'}$ and $c^{''}$ in the training set, and the denominator represents the total number of pairwise neighboring aggregations for the MRF model in the training set.

The definition of distance between two configurations in the feature space is:

$$d_h(c^{'},c^{''}) = \left\{ \frac{1}{T} \sum_{t=1}^{T} (c^{'}(t) - c^{''}(t))^2 \right\}^{-1/2} \tag{6.51}$$

where T is the dimension of features (normalized grid for aggregation) which is 256 in our experiment.

Equation 6.49 indicates that for two neighboring vertices $\mathcal{V}_i$ and $\mathcal{V}_j$, two aggregations which co-occur frequently in the training set and whose configurations are close to each other in the feature space are encouraged by our criterion. The size m in this equation shows that a larger vertex has more influence on its neighbors.

The likelihood $P(y_i|x_i)$ in Equation 2.4 is re-defined as a dependency function:

$$\Psi_o(x_i = c, y_i) = \frac{1}{1 + e^{\lambda d_o(c,y_i)}} \tag{6.52}$$

where $c \in \Omega$ is a configuration assigned to vertex $\mathcal{V}_i$, $d_o(c,y_i)$ is an Euclidean distance function to measure the distance from an observation to a configuration c for vertex $\mathcal{V}_i$. This equation represents the probability of a configuration for a vertex given its observation y_i.

The same belief propagation algorithm as described in section 2.1 is used again for module II to get the optimal configuration $\overline{x}$. The final classification result is obtained by mapping the optimal configuration to two classes (machine printed text and handwritten text) as their relationship is already known during the G-means clustering procedure.

6.4 Experimental Results for Module II - Overlapped Text Separation

To estimate the performance of MRF based classification algorithm for overlapped text separation, we used the same morphology closing operation as module I to collect a total of 331 overlapped patches (including crosses, edit marks and other annotations within a single patch) from the HP Labs data set. The set of overlapped patches was randomly split into a training set of 204 patches and a test set of 127 patches.

In the training phase, the coarsening method described in section 6.2 was used to extract aggregations for training data followed by a vector quantization procedure based on G-means clustering to construct the configuration set Ω with 42 centers from machine printed text and 35 centers from handwritten text.

A similar coarsening procedure was applied on the test data samples to extract aggregations for each vertex v_i to build the MRF to model the overlapped text patch.

Prior to MRF based classification, isolated characters were filtered out based on an estimation of their size. The MRF classification described in section 6.3.2 was implemented to segment overlapped text patch into machine printed and handwritten text.

We measured the performance of the MRF classification algorithm for module II using the recall metric. Recall for machine printed text is defined as the ratio of the amount of pixels which are correctly classified as machine printed text to all machine printed pixels in the test set. The same metric was applied to handwritten text.

Fig. 30 shows the recall curves of machine printed text and handwritten text during BP iteration of classification. The recall for handwritten text increased 11.57% and the overall recall increased 4.72% from the start of BP.

We compared the proposed MRF based segmentation algorithm with an artificial neural network classifier which was implemented using FANN. Table 6 shows that our MRF based method achieved higher overall accuracy as well as higher accuracy on handwritten text than FANN.

Fig. 31 shows sample segmentation results from module II. Red strokes represent handwriting and black strokes represent machine printed text. We see that characters which touch the handwritten brackets or cross, such as character **y** in (a), character **e a n** in b) and characters **m** and **a** in (c) are correctly classified as machine printed. Fig. 31 (d) shows the overlapped patch which is shown in Fig. 29 is also correctly segmented. The broken machine printed text can be restored

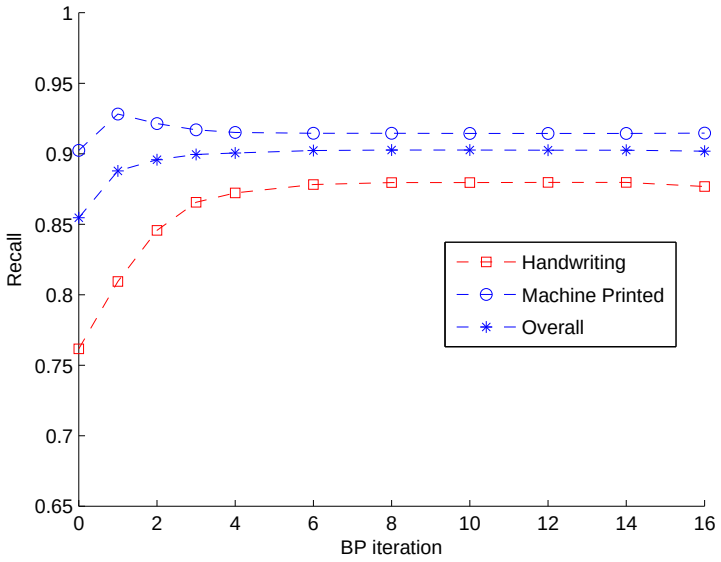


Figure 30: Recall curves during BP iteration

using the method reported in [16].

The main reason for the misclassification of overlapped text in module II is that although the coarsened aggregations have coherent attributes, their sizes are determined by a predefined threshold τ which causes them to not be strictly along the edges of areas where the machine printed text and handwriting intersect. Using a heuristic method to calculate an optimal threshold for each overlapped text segment and applying a shape-driven coarsening algorithm to restrict the growth of aggregates along certain directions should improve performance.

(a) (b)

(c) (d)

Figure 31: Examples with the labeled results for overlapped text from the system.

6.5 Conclusions

In this chapter, we present a novel aggregation procedure to obtain a basic element from overlapped text to separate machine printed text and handwriting.

Table 6: Performance of module II (separation of overlapped text into handwritten and machine printed text) on HP data set (Recall)

	ANN	MRF (Proposed method)
Machine-printed	96.53%	91.44%
Handwriting	61.20%	87.97%
Overall	84.54%	90.26%

We use a MRF based model which inherits its neighbor system from the aggregation algorithm for classification. Experiments show that our method is reliable in segmenting overlapped handwriting from machine printed text. We propose to investigate knowledge-driven aggregation and classification techniques to further improve classification performance.

Chapter 7

Contributions and Conclusions

7.1 Summary

Preprocessing and text separation from mixed documents is one of the key and most challenging tasks in the document analysis and retrieval areas, especially for the situation where handwritten text is surrounded by machine printed text or they are touched together. As most text separation approaches rely on the binarized image, the pre-processing step, document image binarization also plays an important role for the document processing and analysis. These two research topics attract great interests from researchers.

In this dissertation, we have presented a system that has two main components: document image binarization and text separation for annotated machine printed document. We summarize the major design problems by listing their main objectives and approaches as follows:

- **Document image binarization:**

  We focus our document binarization research on those images captured by using hand-held devices, such as cell-phone camera, in in-door office environments.

  1. To overcome the uneven or bad illumination problem, and remove noise from document images as well as keep the text stroke, two non-linear transformation functions are proposed in this dissertation. The first non-linear function defines an adaptive segment surface according to the local information from each single pixels and the second non-linear function maps the original image into a new domain based on the adaptive surface. Initial binarization is done on the mapped document image.
  2. A MAP-MRF based framework is applied to relabel the initial binarized document image where a novel edge potential features are proposed and used. The method of graph cut is used to calculate the optimal labels (configurations) for all the pixels.

- **Text separation:**

  The proposed text separation algorithm aims to separate word level text to machine printed text, handwritten text and noise or overlapped text initially and separate the overlapped text further.

  1. The entire document image is initially separated using a G-means based

NNS (nearest neighbor search) classifier. The G-means based classifier can also be considered as a vector quantization (VQ) procedure in the framework of MRF.

2. A Markov random field based relabeling procedure is implemented to relabel the initial separation result.
3. To overlapped text, extract shape context features for each single pixel and merge them according to their properties. The merged aggregations compose the basic elements for the module II of the proposed text separation system.
4. A G-means VQ procedure is carried out followed by a MRF based relabel approach.

7.2 Major contributions

In this dissertation, various novel features, classifiers, and algorithms were proposed and studied to provide reliable performance of binarization and text separation.

1. In chapter 3, two non-linear transformation functions were proposed to provide an initial binarization result for uneven or bad illumination document images which were captured by cheap hand-held cell-phone camera.
2. To relabel the initial binarization result, we proposed a new edge potential feature which computes the probability of any single pixel be on the stroke

and we used this feature in a novel energy function of MRF. Experimental results show this feature along with the new designed MRF energy function effectively removes the noises and preserves the stroke of the text.

3. In chapter 4, we proposed a novel Markov Random Field based framework to classify three different kinds of text (machine printed text, handwritten text and overlapped text/noises). A Gaussian-like function was defined to measure the spatial distance between patches which is used to construct the irregular neighbor system for MRF. Unlike other relabeling systems that only consider the neighboring relationship in spatial space, our model used distances from both feature space and spatial space to determine the similarity of two neighbors.

4. In order to overcome the imbalanced data set problem for test separation, and intend to replace G-means based initial classification in our framework, we proposed a boost tree classifier which was inspired from Adaboost in chapter 5. During the training phase of the tree-structured classifier, unlike the traditional scheme of the decision tree which uses only a subset of all training samples, we suggested to use all training samples with different weight for each node of the tree.

5. In the module II of our text separation system, which is for the overlapped text separation, a similar MRF framework as previous chapter 4 was used to

further separate the overlapped text into machine printed text and handwritten text in chapter 6. In this chapter, we proposed a coarsening procedure to extract the basic element for classification. A novel posterior probability which considers the relationship of neighboring elements both in feature space and spatial space was designed. The experimental results show that the proposed method significantly improves upon the accuracy of other methods.

6. One merit of the proposed MRF framework is that it is easy to integrate other classifiers which can provide a reliable distance measure in feature space into our system as an initial classifier.

7.3 Future work

Our future research includes the integration of tree-structured classifier to overcome the issue of imbalanced data sets, and development of shape-driven or knowledge-driven coarsening algorithms for overlapped text separation.

Bibliography

[1] Y. Zheng, H. Li, and D. Doermann, "Text identification in noisy document images using markov random model," in *Proc. Seventh International Conference on Document Analysis and Recognition*, vol. 1, pp. 599–603, 2003.

[2] S. N. Srihari, V. Govindaraju, and A. Shekhawat, "Interpretation of handwritten addresses in u. s. mailstream," in *Proc. IEEE 2nd International Conference on Document Analysis and Recognition*, pp. 291–294, 1993.

[3] G. Nagy, S. Seth, and S. Stoddard, "Document analysis with an expert system," in *Pattern Recognition in Practice II*, pp. 149–155.

[4] L. O'Gorman, "The document spectrum for page layout analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1162–1173, 1993.

[5] J. Eduardo Bastos Dos Santos, B. Dubuisson, and F. Bortolozzi, "Characterizing and distinguishing text in bank cheque images," in *Proc. XV Brazilian Symposium on Computer Graphics and Image Processing*, pp. 203–209, 2002.

[6] S. I. Jang, S. H. Jeong, and Y.-S. Nam, "Classification of machine-printed and handwritten addresses on korean mail piece images using geometric features," in *Proc. 17th International Conference on Pattern Recognition ICPR 2004*, vol. 2, pp. 383–386, 2004.

[7] J. Guo and M. Ma, "Separating handwritten material from machine printed text using hidden markov models," in *Proc. Sixth International Conference on Document Analysis and Recognition*, pp. 439–443, 2001.

[8] F. Farooq, K. Sridharan, and V. Govindaraju, "Identifying handwritten text in mixed documents," in *Proc. 18th International Conference on Pattern Recognition ICPR 2006*, vol. 2, pp. 1142–1145, 2006.

[9] L. Zhao and L. Davis, "Iterative figure-ground discrimination," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 1, pp. 67–70, Aug. 2004.

[10] E. Borenstein and S. Ullman, "Combined top-down/bottom-up segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 2109–2125, Dec. 2008.

[11] J. Corso, "Discriminative modeling by boosting on multilevel aggregates," in *Computer Vision and Pattern Recognition, 2008. IEEE Conference on*, pp. 1–8, June 2008.

[12] S. Alpert, M. Galun, R. Basri, and A. Brandt, "Image segmentation by probabilistic bottom-up aggregation and cue integration," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pp. 1–8, June 2007.

[13] W. Freeman and E. Pasztor, "Learning low-level vision," in *Proc. Seventh IEEE International Conference on Computer Vision The*, vol. 2, pp. 1182–1189, 1999.

[14] W. Freeman, T. Jones, and E. Pasztor, "Example-based super-resolution," *IEEE Trans. Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002.

[15] H. Cao and V. Govindaraju, "Preprocessing of low-quality handwritten documents using markov random fields," *IEEE Tran. Pattern Analysis and Machine Intelligence*, vol. 31, pp. 1184–1194, 2009.

[16] H. Cao and V. Govindaraju, "Handwritten carbon form preprocessing based on markov random field," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition CVPR '07*, pp. 1–7, 2007.

[17] J. Banerjee, A. Namboodiri, and C. Jawahar, "Contextual restoration of severely degraded document images," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 517 –524, 2009.

[18] S. Shetty, H. Srinivasan, S. Srihari, M. Beal, and S. Srihari, "Segmentation and labeling of documents using conditional random fields," in *Proc. Document Recognition and Retrieval IV, Proceedings of SPIE*, pp. 6500U–1–11, 2007.

[19] N. Otsu, "A threshold selection method from gray level histograms," *IEEE Trans. Systems, Man and Cybernetics*, vol. 9, pp. 62–66, Mar 1979.

[20] M. Valizadeh, N. Armanfard, M. Komeili, and E. Kabir, "A novel hybrid algorithm for binarization of badly illuminated document images," in *Proc. IEEE 14th International CSI Computer Conference*, pp. 121–126, 2009.

[21] S. Lu and C. L. Tan, "Binarization of badly illuminated document images through shading estimation and compensation," in *Proc. IEEE 9th ICDAR*, vol. 1, pp. 312–316, Sept. 2007.

[22] Z. Shi and V. Govindaraju, "Historical document image segmentation using background light intensity normalization," in *Document Recognition and Retrieval XII*, vol. 5676, pp. 167–174, SPIE, 2005.

[23] M. Pilu and S. Pollard, "A light-weight text image processing method for handheld embedded cameras," in *The 13th British Machine Vision Conference*, 2002.

[24] W. Niblack, *An Introduction to digital image processing*. NJ, USA: Prentice Hall, Englewood Cliffs, 1986.

[25] J. Sauvola and M. Pietikainen, "Adaptive document image binarization," *Pattern Recognition*, vol. 33, pp. 225–236, 2000.

[26] S. S. Bukhari, F. Shafait, and T. M. Breuel, "Adaptive binarization of unconstrained hand-held camera-captured document images," *Journal of Universal Computer Science*, vol. 15, no. 18, pp. 3343–3363, 2009.

[27] R. F. Moghaddam and M. Cheriet, "A multi-scale framework for adaptive binarization of degraded document images," *Pattern Recognition*, vol. 43, pp. 2186–2198, 2010.

[28] M. J. Taylor and C. R. Dance, "Enhancement of document images from cameras," in *Document Recognition V*, vol. 3305, pp. 230–241, SPIE, 1998.

[29] A. Zandifar, R. Duraiswami, A. Chahine, and L. S. Davis, "A video based interface to textual information for the visually impaired," in *Multimodal Interfaces, 2002. Proceedings. Fourth IEEE International Conference on*, pp. 325 – 330, 2002.

[30] T. A. Mahmoud and S. Marshall, "Document image sharpening using a new extension of the aperture filter," *Signal, Image and Video Processing*, vol. 3, no. 4, pp. 403–419, 2009.

[31] D. Doermann, J. Liang, and H. Li, "Progress in camera-based document image analysis," in *Proc. IEEE 7th ICDAR*, vol. 1, pp. 606–616, 2003.

[32] S. Z. Li, *Markov Random Field Modeling in Image Analysis*. Springer, third ed., 2009.

[33] J. M. Hammersley and P. Clifford, "Markov fields on finite graphs and lattices." 1971.

[34] T. Lelore and F. Bouchara, "Document image binarisation using markov field model," in *Proc. IEEE 10th ICDAR*, pp. 551–555, July 2009.

[35] M. Lettner and R. Sablatnig, "Spatial and spectral based segmentation of text in multispectral images of ancient documents," in *Proc. IEEE 10th ICDAR*, pp. 813 –817, July 2009.

[36] J. G. Kuk and N. I. Cho, "Feature based binarization of document images degraded by uneven light condition," in *Proc. IEEE 10th ICDAR*, pp. 748–752, July 2009.

[37] R. R. Schultz and R. L. Stevenson, "Extraction of high-resolution frames from video sequences," *IEEE Transactions on Image Processing*, no. 6, pp. 996–1011, 1996.

[38] R. R. Schultz and R. L. Stevenson, "A bayesian approach to image expansion for improved definition," *IEEE Transactions on Image Processing*, no. 3, pp. 233–242, 1994.

[39] M. Gupta, S. Rajaram, N. Petrovic, and T. Huang, "Models for patch based image restoration," in *Proc. Conference on Computer Vision and Pattern Recognition Workshop*, 2006.

[40] M. Gupta, S. Rajaram, N. Petrovic, and T. Huang, "Non-parametric image super-resolution using multiple images," in *Proc. IEEE International Conference on Image Processing ICIP 2005*, vol. 2, pp. 89–92, 2005.

[41] T. J. Burns and J. J. Corso, "Robust unsupervised segmentation of degraded document images with topic models," in *Computer Vision and Pattern Recognition, 2009. IEEE Conference on*, pp. 1287–1294, 2009.

[42] S. Z. Li, "Markov random field models in computer vision," vol. B, pp. 361–370, 1994.

[43] S. Z. Li, "A markov random field model for object matching under contextual constraints," in *In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 866–869, 1994.

[44] Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection," *PAMI*, vol. 27, pp. 1778–1792, 2005.

[45] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, 1983.

[46] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equations of state calculations by fast computing machines," *Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[47] S. Geman and D. Geman, "Stochastic relaxation, gibbs distribution and the bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721–741, 1984.

[48] D. Greig, B. Porteous, and A. Seheult, "Exact maximum a posteriori estimation for binary images," *Journal of the Royal Statistical Society*, no. 2, pp. 271–279, 1989.

[49] L. Ford and D. Fulkerson. Princeton Univ. Press, 1962.

[50] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. MIT Press and McGraw-Hill, second edition ed., 2001.

[51] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Recognition and Machine Intelligence*, vol. 26, pp. 1124–1137, Sept 2004.

[52] L. Xiong, F. Wang, and C. Zhang, "Multilevel belief propagation for fast inference on markov random fields," in *Proc. Seventh IEEE International Conference on Data Mining ICDM 2007*, pp. 371–380, 28-31 Oct. 2007.

[53] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[54] A. Raj and R. Zabih, "A graph cut algorithm for generalized image deconvolution," in *Proc. IEEE 10th ICCV*, vol. 2, pp. 1048 –1054, Oct 2005.

[55] TESSERACT-OCR:, "http://code.google.com/p/tesseract-ocr/,"

[56] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[57] G. Hamerly and C. Elkan, "Learning the k in k-means," in *InProc. 17th NIPS*, 2003.

[58] X. Peng, S. Setlur, V. Govindaraju, R. Sitaram, and K. Bhuvanagiri, "Markov random field based text identification from annotated machine printed documents," in *Proc. IEEE 10th International Conference on Document Analysis and Recognition*, pp. 431–435, 2009.

[59] N. Petkov, "Biologically motivated computationally intensive approaches to image pattern recognition," *Future Generation Computer Systems*, vol. 11, no. 4-5, pp. 451–465, 1995.

[60] N. Petkov and P. Kruizinga, "Computational models of visual neurons specialised in the detection of periodic and aperiodic oriented visual stimuli: bar and grating cells," *Biological Cybernetics*, vol. 76, no. 2, pp. 83–96, 1997.

[61] N. P. S.E. Grigorescu and P. Kruizinga, "Comparison of texture features based on gabor filters," *IEEE Trans. on Image Processing*, vol. 11, no. 10, pp. 1160–1167, 2002.

[62] T. Anderson and D. Darling *Annals of Mathematical Statistics*, pp. 193–212, 1952.

[63] F. Fleuret and I. Guyon, "Fast binary feature selection with conditional mutual information," *Journal of Machine Learning Research*, 2004.

[64] X. Peng, S. Setlur, V. Govindaraju, and R. Sitaram, "Text identification from mixed documents using weighted features," in *Proceedings of 14th Conference of the International Graphonomics Society*, 2009.

[65] J. Xu, V. Singh, V. Govindaraju, and D. Neogi, "A hierarchical classification model for document categorization," in *Proc. 10th IEEE International Conference on Document Analysis and Recognition*, pp. 486–490, Aug 2009.

[66] B. Wu and R. Nevatia, "Cluster boosted tree classifier for multi-view, multi-pose object detection," in *Computer Vision, ICCV, IEEE 11th International Conference on*, pp. 1–8, Oct 2007.

[67] Z. Tu, "Probabilistic boosting-tree: learning discriminative models for classification, recognition, and clustering," *Proc. 10th IEEE International Conference on Computer Vision ICCV*, vol. 2, pp. 1589–1596, 17-21 Oct 2005.

[68] E. Grossmann, "Adatree: boosting a weak classifier into a decision tree," in *Computer Vision and Pattern Recognition Workshop, CVPRW04*, June 2004.

[69] X. Peng, S. Setlur, V. Govindaraju, and R. Sitaram, "Text separation from annotated documents using a tree-structured classifier," in *Proc. IEEE 20th ICPR*, 2010.

[70] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Proceedings of the Second European Conference on Computational Learning Theory*, pp. 23–37, 1995.

[71] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning*, vol. 37, no. 3, pp. 297–336, 1999.

[72] Y. Freund and R. E. Schapire, "A short introduction to boosting," in *In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pp. 1401–1406, Morgan Kaufmann, 1999.

[73] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *Annals of Statistics*, vol. 28, p. 2000, 2000.

[74] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[75] P. Viola and M. Jones, "Robust real-time object detection," in *2nd International Workshop on Statistical and Computational Theories of Vision - Modeling, Learning, Computing, and Sampling*, 2001.

[76] A. Torralba, K. Murphy, and W. Freeman, "Sharing visual features for multiclass and multiview object detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, pp. 854 –869, may. 2007.

[77] C. Huang, H. Ai, Y. Li, and S. Lao, "Vector boosting for rotation invariant multi-view face detection," vol. 1, pp. 446–453, Oct. 2005.

[78] C. Huang, H. Ai, Y. Li, and S. Lao, "High-performance rotation invariant multiview face detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, pp. 671–686, April 2007.

[79] Y. Sun, M. Kamel, and Y. Wang, "Boosting for learning multiple classes with imbalanced class distribution," in *Proc. Sixth International Conference on Data Mining ICDM '06*, pp. 592–602, 2006.

[80] M. Kearns and Y. Mansour, "On the boosting ability of top-down decision tree learning algorithms," in *In ACM Symp. on the Theory of Computing*, pp. 459–468, 1996.

[81] FANN:, "http://leenissen.dk/fann/."

[82] X. Peng, S. Setlur, V. Govindaraju, and R. Sitaram, "Overlapped text segmentation using markov random field and aggregation," 2010.

[83] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, pp. 509–522, Apr 2002.