EXTRACTION OF TEXTUAL INFORMATION FROM IMAGES FOR INFORMATION RETRIEVAL

By

LIN-LIN LI

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
AT
DEPARTMENT OF COMPUTER SCIENCE
NATIONAL UNIVERSITY OF SINGAPORE
OCTOBER 2009

Acknowledgements

I would like to express my deep and sincere gratitude to my supervisor, Professor Chew Lim Tan for his valuable guidance and constant support through this thesis research, and his understanding and encouragement in the early years of chaos and confusion.

I would owe my warm and sincere thanks to Dr. Shi Jian Lu, who gave me important guidance during my first steps into this research area, and thanks for his detailed and constructive comments. I also sincerely appreciated the effort made by Mr. Peng Zhou, thanks for his valuable assistance to this thesis.

The episode of acknowledgement would not be complete without the mention of my colleagues in the Center of Information Mining and Extraction (CHIME) of School of Computing, National University of Singapore: Man Lan, Rui Zhe Liu, Tian Xia Gong, Li Zhang and Jie Wang. Thanks for their friendly help and social support during the period of my graduate study.

Last but not least, my special gratitude is due to my parents for their silent support throughout all these years, as well as to Mr. Yan Song for his continuous encouragement during my study.

Lin-Lin Li

March, 2009

Abstract

Traditional document image analysis relies on Optical Character Recognition (OCR) to obtain textual information from scanned documents. However, as the development of digitization technology, the current OCR technique is no longer sufficient for this purpose.

With the increasing availability of high performance scanners, many projects have been initiated to digitalize paper-based materials in bulk and build large multilingual document image databases. Two inherent shortcomings, namely, language dependency and slow speed, are the main obstacles for current OCR to fully access the textual information of such databases. We address both problems for clean and degraded scanned document images respectively. In particular, a word shape coding method has been proposed, which is 20 times faster than OCR. This method has been successfully employed in language identification and document filtering for clean scanned document image archives. Furthermore, a holistic word spotting method, invariant to geometric transformations of translation, scale, and rotation, is proposed to facilitate fast retrieval for degraded scanned document images. This method is optimized for the U.S. patent database, which have many degraded document images with severe skew.

The rapid development of camera technology has also challenged current OCR technique. The advancement of cameras has given people an alternative to traditional scanning for text image acquisition. However, because the image plane in a camera is not parallel to the document plane, camera-based images suffer from perspective distortion, leading to a failure when OCR or other textual information techniques are applied to them directly. In this thesis, this problem is addressed for camera-based document images and real scene images respectively. For camera-based document images, another word shape coding scheme, which is a variant of our holistic word spotting method, is proposed for language identification and fast retrieval. This method is Affine invariant, and thus is robust to moderate perspective deformation, which is sufficient for this image type. For real-scene images, which may have more severe perspective deformation, we propose a character recognition method based on a global descriptor called Cross Ratio Spectrum. With this descriptor, the perspective deformation of a character is compressed into a stretching deformation, and thus can be solved by Dynamic Time Warping. Besides characters, the method is also applicable to multi-component planar symbols.

Table of Contents

List of Tables

List of Figures

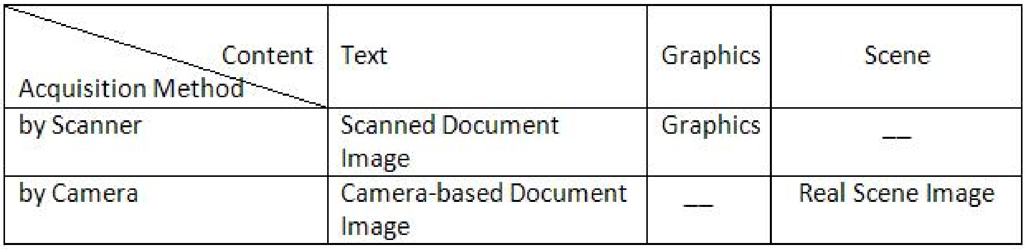
Chapter 1

Introduction

The history of communication dates back to the earliest signs of life. Communication can range from very subtle processes of exchange to full conversations and mass communication. Human communication was revolutionized with speech about 200,000 years ago. Symbols were developed about 30,000 years ago, and writing about 7,000 years. Although it emerged latest, writing is the most efficient and reliable way to communicate. Two aspects of writing are critically important in communication: content and format. In the world of computer, the former is called text and the latter is features other than text like color, size, and font.

Text is the core of writing. Many storage media have been used for writing in the early stage: stone, bones, bronze implements, turtle shells, papyrus, clay tablets, and bamboo pieces from the Warring States to Jing Dynasty in Chinese history. One of the most exciting technological innovations, improving the quality of text conservation, was the creation of paper by a Chinese inventor, Lun Cai, about 1800 years ago. Another essential innovation of text storage media took place when digitization devices came out into being since 1960.

Two types of digitized text can be found nowadays, namely, plain text and imaged



Content Acquisition Method	Text	Graphics	Scene
by Scanner	Scanned Document Image	Graphics	—
by Camera	Camera-based Document Image	—	Real Scene Image

Table 1.1: Categories of imaged text, classified by the acquisition method and content.

text. Plain text comprises of unformatted sequential code like ASCII. Many information retrieval techniques have been established for managing plain text. On the other hand, imaged text is stored as raw pixels. Table 1.1 shows several categories of imaged text, divided by their acquisition method and content. Images in different categories have their own characteristics and processing techniques.

Scanned document images are electronic images of documents produced by a scanner or photocopier. It is the most predominant image medium by which textual information is disseminated. The benefits of digitization are obvious. Information stored electronically consumes less space, and is much easier to duplicate and deliver. Besides, convenience of access is not tied to the physical proximity of materials any more. The content of **graphics** includes engineering drawings, maps, figures, and so forth. Text in graphics often functions as annotations, legends, or captions. It is particularly crucial, because it is useful for describing the semantic content of graphics, and it can be easily extracted compared to other semantic contents. The increasing availability of high performance, low-priced, portable digital imaging devices has created a tremendous opportunity for supplementing traditional scanning for document image acquisition. To differentiate from images captured by a scanner, we term images captured by a camera as **camera-based images**. A **camera-based document image** is camera-based image whose content is a text document. In this thesis, we use the term **real-scene image** to refer a scene photo which contains textual information such as a road sign. It worth noting that, cameras are also used to capture graphics images and videos, however, both of them will not be included in the scope of this thesis.

It is easy for humans to recognize textual information from images. However, with variations in size, font, orientation, resolution, and decoration, it is quite a difficult task for computers. In order to get machine-editable text from images, two steps are necessary, namely, text location and extraction. Text Location basically answers the question of where is the text present? Text Extraction is to extract content-level information, for example the identity of language using in an imaged text, the presence of a keyword in the image, or the exact text of the image.

For four types of text images introduced in table 1.1, scanned document images processing and graphics processing have been extensively studied. In contrast, the processing of images captured by cameras, including camera-based document images and real-scene images is at a rather preliminary stage.

Because information retrieval techniques, developed for plain text, cannot be directly applied to imaged text, **textual information extraction techniques** have been established to bridge the gap. **Optical Character Recognition** (usually abbreviated to OCR) is the predominant technique to translate images of typewritten or handwritten text into machine-readable text character by character. The state of the art commercial OCR software has been highly successful in recognizing standard business documents produced by modern photocopiers or scanners. In addition, there are two complementary techniques, which outperform OCR under certain conditions. One technique is **Word Shape Coding**, which maps the character set to a smaller symbol set other than the real character identities. For methods in this category, a word is represented by a sequence of symbols. These methods are much faster than OCR, and thus often are employed in document image processing applications which have critical time constraints. The other technique is **Holistic Word Spotting**. Different from OCR which recognizes each individual character, this technique recognizes a word as a whole entity. In this approach, a word image is represented by a feature vector of pixel-level features of the whole word image. Since no segmentation is needed, this technique is robust to the noise of poor-quality images, especially touching or broken characters. Therefore, this approach is particularly useful in word spotting application for degraded image documents.

1.1 Main Problem Statement

Many factors degrade the performance of textual information extraction techniques. For scanned document images, salt and pepper noise, touching and broken characters, and skew have long been the processing obstacles. For camera-based images, low resolution, blur, warping, as well as perspective distortion [LDL05] are the major challenges. Among these degradation factors, we are particular interested in geometric deformations, i.e. skew and perspective distortion. Skew may be generated in a scanned document image if the edge of the paper is not aligned correctly with the scanner during scanning. Perspective deformation of a camera-based document image is caused by the fact that the image plane in the camera is not parallel to the document plane, and manifests as severe skew, unpredictable orientation, non-parallel text-lines, and variable character sizes.

Existing textual information extraction techniques show little tolerance to geometric transformation. Skew degrades the speed and accuracy, and perspective deformation, especially in real-scene images of a sparse text context, is almost inaccessible for existing text extraction techniques. OCR, Word Shape Coding, and Holistic Word Spotting are all developed and optimized for images captured by scanners, which are produced from pseudo binary hardcopy paper manuscripts with a flatbed imaging device. Therefore these extraction techniques assume that the image to be processed is a parallel projection of the source document. However, the assumption does not hold when it comes to images taken by cameras. Because camera-based images are captured by a portable device in less constrained environments.

Given the presence of geometric deformation in a text image, a rectification step is indispensable. Skew detection for scanned document images has been extensively studied. On the contrary, the research on perspective rectification is at a preliminary stage. Only a few methods have been proposed to remove the perspective deformation of camera-based document images, and rectify the them into a fronto-parallel pose, using clues of the text format. Real-scene images pose a even greater challenge to recertification, because their text content may be sparse and could be any unpredictable format. To my knowledge, there is no rectification method generally applicable to real-scene images. Anyway, once a rectification is taken, it will take extra processing time and may cause errors which pass to downstream steps.

In view of this, a critical question is raised by us: **how can we directly access the content of a text image with geometric deformation without rectification**?

1.2 Solutions in this Thesis

In order to answer this question, we have proposed several content access methods for scanned document images, camera-based document images, and real scene images respectively. These methods requires no rectification. The benefits are obvious: extra processing time is saved, and possible errors introduced by the rectification are avoided. In particular, these methods are:

- A fast and reliable word shape coding method is proposed for clean document images without deformation. It is more than 20 times faster than OCR and thus is able to satisfy the requirement of time critical retrieval applications. It is employed in language identification and document image filtering applications for clean document images. This is a starting work for me to get familiar with this area.

- A word shape coding method is proposed for camera-based document, dealing with perspective deformation. It is invariant to affine deformation images, and thus robust to weak perspective deformation introduced by a camera. Language identification and document similarity estimation techniques are also established based on the coding method.

- A word spotting method is proposed for degraded document images, invariant to rotation transformation. This method is a variant of the word shape coding method for camera-based document images proposed above. It has been employed in a fast word spotting program for viewing U.S. patent documents.

- A character recognition technique, which is invariant to perspective deformation, is proposed. This method is also able to recognize more complex real-scene symbols like traffic signs. In addition, the point-level correspondence, given by this method concurrently when recognizing characters or symbols, can be used for restoring the fronto-parallel view if necessary.

1.3 Thesis Preview

This thesis is organized as follows. In Chapter 1, a preview of the whole thesis has been provided, including the scope of the thesis, the main problem and main contributions. In Chapter 2, I will introduce the background knowledge about textual information extraction, applications of text images, as well as linear geometric deformation theory. In Chapter 3, I will present a word shape coding method, and explain how to integrate it in language identification and document filtering for clean document image achieves. In Chapter 4, I will introduce a word shape coding method, and detail the way to employ it in language identification and document similarity estimation for camera-based document image achieves. In Chapter 5, a variant of the word shape coding method introduced in Chapter 4 is adapted to swiftly locate keywords in degraded patent images, regardless of the skew angle. In addition, a clustering based method to locate textual content in the drawings of patent documents will be present. In Chapter 6, I will detail a symbol recognition technique which is resistant to severe perspective deformation. Chapter 7 is a conclusion chapter.

Chapter 2

Background Knowledge

2.1 Textual Information Extraction Techniques for Scanned Document Images

Textual information extraction techniques for scanned images are divided into three categories: OCR, Word Shape Coding, and Holistic Word Spotting. The ultimate goal of extracting textual information is for information retrieval. The output of the extraction are passed to downstream retrieval applications.

First of all, I will make a very brief introduction about typical retrieval applications for scanned document images. **Language identification** is to determine which language the document image is written in. It is an important pre-processing step before document image indexing or retrieval can take place in a multilingual image archive. **Keyword spotting** is to locate the occurrence of certain keywords in one document image. It is a useful tool for viewing document images. **Document image retrieval** is to retrieve document images relevant to a query from a document image archive. Document image retrieval is further classified according to the query and the output. The query of **Boolean document image retrieval** comprises of a few keywords connected by Boolean operators. Keywords are considered to be either present or absent in a document and to provide equal evidence with respect to information needs. A Boolean retrieval model does not have a built-in way of ranking matched documents by some notion of relevance. On the contrary, **ranked document image retrieval**, which also takes a few keywords as the query, ranks the retrieved result according to their relevance to the query. The query of **document image similarity estimation** is a document image.

OCR, Word Shape Coding and Holistic Word Spotting techniques have different target applications that overlap a little. Table 2.1 is an overview of retrieval applications based on OCR, Word Shape Coding and Holistic Word Spotting respectively.

Table 2.1: An overview of applications that OCR, Word Shape Coding (WSC), and Holistic Word Spotting (HWS) are applied to.

Technique	Applications	References
OCR	Ranked Document Image Retrieval	[CHTB94, TBC94] [HCW97, TNB01b, TBC96] [BSM95, OTA97, Tak97, OTA97]
	Document Image Categorization	[ILA95, TNB+01a, Vin05]
	POS Tagging	[Lin03]
WSC	Language Identification	[LT08, Spi97, NBSK97] [Nak94, LT06b]
	Document Similarity Estimation	[LT04, THS+03]
	Boolean Document Image Retrieval	[SS97]
	Fast Keyword Spotting	[Spi94, LT04]
HWS	Keyword Spotting in Degraded Images	[RM03, MMS06, KJM07, HHS92]

From Table 2.1, we can see that OCR has been mainly employed in ranked document image retrieval and document image categorization. Word Shape Coding technique has been mainly employed in language identification. Holistic Word Spotting technique mainly works for keyword spotting in degraded images. An illustration of

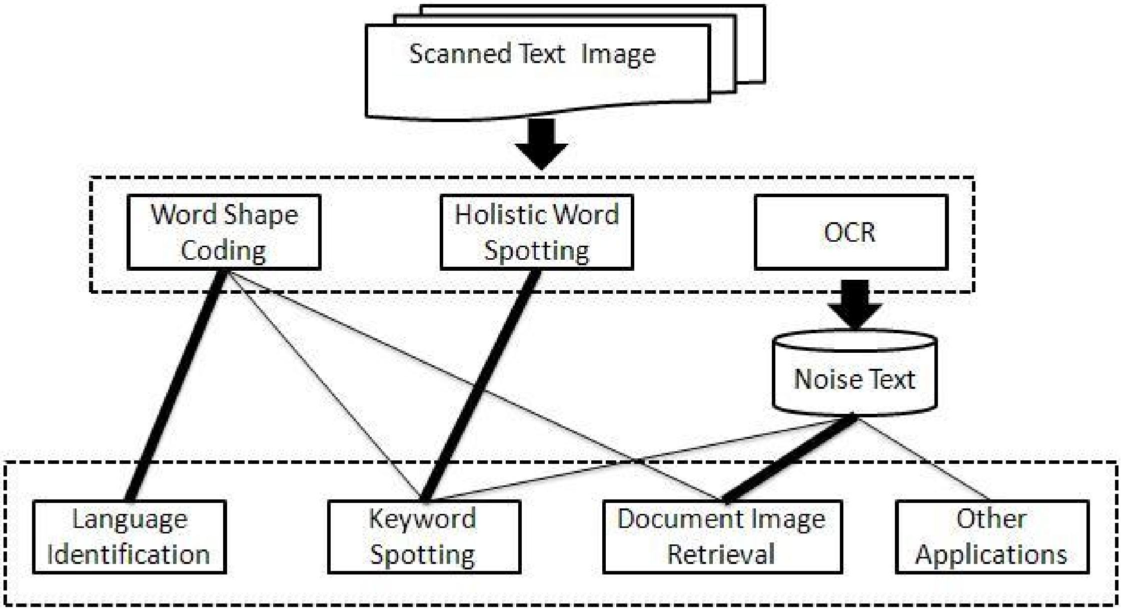


Figure 2.1: Textual information extraction techniques and document image retrieval applications.

this relationship is shown in Figure 2.1. This is caused by the fact that OCR has shortcomings of slow speed, language dependency, and fragility to degraded image quality, and thus is not suitable for certain applications. Therefore, both complementary techniques are proposed as alternatives to OCR for these applications. I will detail this point later in this section under topic “Why not OCR?”.

In the rest of this section, I will make a detailed explanation about these three techniques and their retrieval applications.

2.1.1 Optical Character Recognition

OCR is the mechanical or electronic translation of images of handwritten, typewritten or printed text (usually captured by a scanner) into machine-editable text. It is the dominant approach to extract character-level information from document images. OCR is a fundamental step to bridge the gap between imaged text and modern information retrieval technology. OCR is a complicated procedure, generally including the steps below:

- Character segmentation, which identifies the bounding box for each character in a document image. Touching characters lead to false bounding box detection, and hence false recognition results.

- Feature extraction, which extracts features from character images. Features can be divided into two groups: global features and local features. Global features include the number of holes in the character, the number of concavities on the contour, and the relative protrusion of character extremities. Local features include relative positions of line endings, crossovers and corners.

- Character classification, which assigns each character image with a character identity. A typical character classification based on the statistical classification is described below. Character image patterns are represented by points in a multidimensional feature space. A classifier partitions the feature space into regions associating with each class, labeling an observed pattern according to the class region into which it falls. Hence for each character class either a prototype or a set of character samples must be known.

- Post processing, which employs language knowledge to correct errors in recognition committed, for example, by comparing OCR output with a pre-defined lexicon.

The advantage of OCR is that, all information retrieval techniques designed for clean text are theoretically applicable to the output of OCR. As shown in Figure 2.1, the retrieval is conducted as the text level. However, errors may happen in each step of OCR, and thus many efforts have been made to narrow the gap between noisy text retrieval and traditional clean text retrieval.

Retrieval Applications based on OCR

Ranked Document Image Retrieval. Research on OCR text retrieval have blossomed since 1994. Many efforts have been made in preparation testing data. The information Science Research Institute established at the University of Nevada[1] has been making many contributions. Consequently in 2000, a collection of OCR-error-prone word images selected from testing images was published for researchers to get insights into the strengths and weaknesses of current OCR systems [NNR00]. Meanwhile, the text retrieval conference TREC-4 and TREC-5 held confusion tracks, and provided recognized text from scanning images in order to facilitate IR research on OCR text. In an attempt to avoid the tedious process of scanning and OCR involved in obtaining testing collection, Doermann et al. [DY95] developed a system which could generate simulated noisy text, which simulates errors in OCR output for evaluating the performance of various text analysis systems under varying, yet controlled conditions. The system presented a set of symbols and page models which are used to degrade an ideal text by introducing errors which typically occur during scanning, decomposition and recognition of document images.

Several researchers made quantitative studies on the impact of OCR errors on text retrieval performance. After experiments on simulated text, Croft et al. [CHTB94]

[1] http://www.isri.unlv.edu/ISRI

showed that high quality OCR text has little effect on the accuracy of retrieval, but short documents with low quality OCR significantly degrades result. Taghva et al. [TBC94] experimented with actual OCR text. They found that, for a Boolean system, the problem caused by OCR errors could be overcome by redundancy of the document text. However, for a statistical model, the impact of OCR errors becomes unimportant with insignificant retrieval performance degradation [TBC96, TNB01b]. Further, they observed that the ranking and feedback associated with IR models are not robust to deal with OCR errors. Besides, the OCR errors and garbage strings generated by the mistranslation of graphic objects increase the size of the index by a wide margin. An important conclusion drawn by Doermann [Doe98] after a complete review over applying text retrieval methods on OCR text is that: for OCR text whose character accuracy is higher than 80%, the average precision and recall of IR is not adversely affected by OCR errors; however, for character accuracy lower than 80%, the average precision and recall fall drastically. According to this guideline, some databases indexed OCR text for full text retrieval purpose. For example, in DIENST[2] which is a searchable database developed by Cornell University, image documents are transcribed into OCR text for indexing and full-text retrieval purpose. On the other hand, because the quality of OCR text is not adequate for display purpose, they only act as an invisible layer for search purpose. What users would see in their screens are documents in image format. This framework was also adapted by JSTOR later.

Although IR systems seem to be robust to OCR errors to a certain level, methods are explored to enhance the performance to fulfill higher requirements. Three important approaches are widely used, including query expansion [BSM95, OTA97],

[2] http://www.cs.cornell.edu/cdlrg/dienst/DienstOverview.htm

approximate matching [Tak97, OTA97], as well as N-gram matching [HCW97].

Document Image Categorization. Besides OCR text retrieval, investigation is carried on other retrieval-related applications including OCR text categorization, document clustering and information extraction. Ittner et al. [ILA95] conducted a categorization task to assign 63 overlapping categories to 1000 document images by a Rocchio classifier. They found that categorization accuracy decreases when OCR recognition accuracy decreases, and they also observed that the categorization performance on OCR text was better when the classifier was trained on OCR text other than noise-free text. A similar conclusion is drawn by Taghva et al. [TNB$^+$01a]. In addition, they also make some observations that dimensionality reduction improves categorization, and that OCR errors may have little effect on the categorization performance when OCR character accuracy is above 90%. Based on experiments on simulated text corpus whose word accuracy of a document spans from 10% to 50%, Vinciarelli [Vin05] observes that the categorization of noisy text has similar performance when the recall is less than 20%, however, performance decreases rapidly when a higher recall is required. Therefore, he proposed a new measure other than the word accuracy called information gain recall and information gain precision, which are expected to have a better linear relation between the noise estimation and the performance of categorization.

POS Tagging. Text processing tasks like information extraction which involve more complete access to the content of documents are more sensitive to OCR errors. Lin [Lin03] analyzes the performance of both individual POS taggers and combination systems on imperfect text. Experimental results show that a POS tagger's accuracy decreases linearly with the character error rate and the slope indicates a tagger's sensitivity to input text errors. Different from statistics-based applications like document image retrieval and categorization, POS tagging performance degrades linearly with the OCR accuracy.

Why not OCR?

After years of developing and improving, the state of the art commercial OCR software can achieves a 99% or more character recognition accuracy on standard business documents, and act as one necessary component of current imaged document retrieval projects. However, it is still necessary to establish alternative techniques to access content of document images. As a nutshell, current OCR techniques still have the following drawbacks:

Language dependence. In OCR process, unknown characters are compared with a set of trained templates. If the real identity of a character does not appear in the template set, the process fails. Additionally, if the template set comprises of many templates, the recognition speed will be slowed down. Therefore, in a general OCR process, users are required to manually choose the language of the input document, or the software itself assigns one or a few default languages to the document. However, either human intervention or setting default language is inadequate for a multilingual environment, and thus many language identification techniques are being developed.

Long execution time. Text generated by OCR is only suitable for a limited range of applications. There are many applications related to document image retrieval, and each of them has different requirements in terms of accuracy, storage and speed. Generally, the accuracy of OCR output is the most important aspect which is of great concern. In order to get higher transcription accuracy, a typical OCR software integrates many steps, leading to a slow program. The execution time of OCR makes it unacceptable for time-critical applications. For example, a document filtering system sifts through a steam of incoming information to find documents relevant to a set of user needs represented by profiles. This application emphasizes speed as well as indexing methods that enable very fast processing of documents against profiles. Besides, generally it takes tens of seconds to transcribe a document image by OCR. The time complexity also makes OCR unsuitable to document image archives of very large volume. For example, assuming it takes 20 seconds for an OCR software to process an scanned image of a A4 paper on my own PC, configured with 2.33GHz CPU and 3.25GB RAM. For a database with 5,000,000 images, it takes about 120 days to transcribe all images with 10 such PCs.

Susceptibility to images with poor quality, rare fonts. Current OCR software is only suitable for a limited range of images. Because characters of the alphabet are subject to many variations in terms of fonts, styles and size, OCR may fail when encountering a rare font or style which is different from these character prototypes employed in training. Furthermore, touching of adjacent characters, broken strokes due to poor binarization and noise in a real image all contribute to OCR errors. In fact, commercial OCR software work well on standard business documents generated from modern printers. But the OCR accuracy degrades with scanning photocopies, and small and highly stylized fonts such as those on business cards. When it comes to typesetting pages from books or newspapers (which are target content of projects like Google book Search and the Open Content Alliance), commercial OCR software give unacceptable results. Therefore, almost in every document image project, customization of OCR is inevitable.

2.1.2 Word Shape Coding

Due to the language dependency and long execution time of OCR, Word Shape Coding technique has been proposed in previous work. Word shape coding methods [Spi94, LLT08, THS$^+$03, LT04] take an individual character as input, and map character objects to a smaller symbol set. For example, 6 codes are employed in Spitz's method [Spi94], namely 'A','x','e','g','i', and 'j'. A word is represented by a code string. For example, the word shape coding representation for the word "left" is "AxAA" in Spitz's coding method [Spi94]. Because the encoding (mapping) process is based on a set of simple and universal image features, Word Shape Coding methods are fast computable and language independent. Therefore, they are widely employed in document image retrieval applications with speed constraints and language identification.

The detail of four important word shape coding methods [Spi94, LLT08, THS$^+$03, LT04] are introduced in Appendix A, with in an ascending order to the number of symbols used in coding method. We have used these four methods as comparative methods in our experiments in Chapters 3, 4, and 5. For simplicity, in the rest of this thesis, they will be referred as **TAN's** [THS$^+$03], **LU's** [LLT08], **SPITZ's** [Spi94], and **LV's** [LT04] respectively. These four coding methods have been widely employed in many retrieval applications, as shown in Table 2.2.

In word shape coding methods, each word maps uniquely to a corresponding symbol string, but one symbol string may be mapped to several real words because of the reduced symbol set, leading to **ambiguity**. The ambiguity is different from method to method. It is an important retrieval performance indicator.

Table 2.2: An overview of applications that these four coding schemes are applied to.

Coding Scheme	Applications	References
TAN'S	Document similarity estimation	[THS+03]
	Language identification	[LT06b, LT08]
LU's	Document similarity estimation	[LLT08]
	Keyword spotting	[LLT08]
SPITZ's	Language identification	[Spi94, Spi97, NBSK97, Nak94]
LV's	Document similarity estimation	[LT04]
	Keyword spotting	[LT04]

Retrieval Applications based on Word Shape Coding

Fast Document Image Retrieval. Word Shape Coding techniques has been employed in the ranked document image retrieval. Based on SPITZ's coding scheme, Smeaton et al. [SS97] apply vector space retrieval model to code strings. They use standard information retrieval steps of stopword removal and stemming to process the query, and score each document based on the $tf.idf$ weight of the processed query term. However, their experiment shows that multiple matches between each word shape surface form occurrences in the document text (i.e. ambiguity) and the skewed distribution of these lead to poor performance. On the contrary, with little ambiguity, LV's method shows a good performance in Boolean document image retrieval [LT04].

Due to ambiguity, word shape coding methods are widely employed in document similarity estimation, where they achieve a good performance. Because the query of this application is a whole document image, which compensates for the effect of ambiguity. Yu et al. [YT00] propose a character shape coding scheme. The encoding is based on the vertical traverse density (VTD) and horizontal traverse density (HTD) of the character object. HTD is a vector whose elements denote the numbers of line segments as scanning the character horizontally line by line from top to bottom. Similarly, VTD is another vector obtained from vertical scanning from left to right. After assigning each character object a code, they employe a N-gram model to evaluate the similarity between vectors of two document images. Similar works with different coding schemes could also been found in [LT04] (LV's coding) and [THS$^+$03] (TAN's coding). It has been proven in [LT04, THS$^+$03, YT00] that the document similarity estimation based on Word Shape Coding schemes is much faster than that based on OCR results, from several to twenty times , without any significant retrieval performance degradation.

Language Identification. Many methods have been reported for language identification of scanned document images. They are divided into three categories. The first category is component based [LK95, HKKT97]. These methods are proven to be accurate, but very slow and not training-free. The second category is texture based [BBS05]. Methods in this category are sensitive to the layout of the document image. A more detailed introduction of these two categories will be made in Section 4.1 of Chapter 4. The third category is Word Shape Coding based, such as [Spi97, Nak94, NBSK97] based on SPITZ's coding and [LT06b, LT08] based on TAN's coding.

A typical language identification procedure based on Word Shape Coding technique is as follows. A list of the most frequently-used words in each language (often stopwords) is encoded into some kind of word shape code strings. When a document image comes in, it is also encoded. The encoded document is then compared with the list. The language identity of the document is the one whose list has the most agreement with the document. Word Shape Coding based methods are free from training and layout constraints.

A hybrid method, integrating word shape coding scheme and other features, is presented by Tan et al. [TLH99], to identifying English, Chinese, Malay or Tamil in imaged document. They distinguish Chinese, Latin, and Tamil based on two attributes: bounding box elongation and the distribution of upward concavities. Then, they distinguish English and Malay based on the statistics of the most frequent word shape code strings. Finally, they choose AAx (the) and xxA for English and gxxg (yang) and Axx (dan) Malay language as the most frequent word shape code strings. Their experiment results show that word shape coding methods is the best way to distinguish among languages that share a similar character set.

2.1.3 Holistic Word Spotting

Holistic Word Spotting [WZH00, CB93, TLH99] treats each word as a whole entity and thus avoids the difficulty of character segmentation. This is different from the character-level processing strategy taken by OCR and Word Shape Coding, and is exactly why Holistic Word Spotting is robust to degraded image quality. In particular, broken and touching characters are one of the major document image degradation factors, and Holistic Word Spotting is naturally immune to them. Another important reason why Holistic Word Spotting presents an attractive alternative, lies in its the apparent similarity in the approach to how humans read text. Hull [Hul86] points out the fact that according to psychological experiments, humans do not read text character by character, rather they recognize words or even small groups of nearby words while they are reading. Holistic word spotting approaches are widely employed in keyword spotting application for degraded document images.

Retrieval Applications based on Holistic Word Spotting

Keyword Spotting in Degraded Images. Both Word Shape Coding [LLT08, LT04, Spi94], and Holistic Word Spotting [RM03, MMS06, KJM07, HHS92] techniques are employed in the keyword spotting application. In addition, a few document image retrieval systems [HHS90, YT00] take a hybrid approach, combining both Word Shape Coding and Holistic Word Spotting techniques.

For the word shape coding approaches [LLT08, LT04, Spi94], document images are firstly converted and stored as symbol strings. A query is translated into the same symbol representation by means of a table lookup. The keyword is spotted by strings matching. Particularly, word shape coding methods are employed mainly because of their fast processing speed; however, the performance drops rapidly when image are degraded. In contrast, Holistic Word Spotting is used for badly degraded images, such as handwritten documents [RM03], historical documents [KJM07, MMS06] , scanned images of envelopes [HHS90, HHS91, HHS92], and some document image archives in rare languages for which OCR is not available.

In Holistic Word Spotting, document images are firstly converted and stored as feature vectors. A word image synthesized from a set of character samples of various fonts is generated for the query. Feature vectors of the query are then computed. The matching is based on cosine similarity or more sophisticated classifiers. In particular, a method to index modern printed documents on word-level was proposed in [MMS06]. Each component image is scaled to fit an 8 × 10 grid, resulting in an 80-dimensional feature vector after concatenating the pixel density values in each grid item. The characters are then clustering by Self-Organizing Map (SOM). A word is represented by the cluster centroid of its character objects. Ho et al. [HHS90, HHS91, HHS92] proposed a method to spot keywords in degraded images. There are 35 descriptors used to represent the local features of a word image. The vector stores the normalized number of pixels belonging to four categories of strokes (east-west, northeast-southwest, north-south, and northwest-south). The matching between two word images is done by a nearest neighbor classifier. They applied the method on an image database of city names extracted from address block images, and were able to achieve a spotting accuracy at 90.57%. In [RML04], Fourier coefficients of upper and lower word profiles and projection profile are employed as a feature to represent a word image, because it is easy to get a fixed length representation.

2.2 Textual Information Extraction Techniques for camera-based images

High-end digital cameras[3] have long been used in large-scare book digitizing projects, mainly for dealing with paper materials that cannot be flattened, like thick rare books, fragile historical manuscripts or brittle paper [LDL05]. Document images produced by high-quality cameras have comparable quality to those produced by scanners, and thus the downstream processing are similar.

To date, there is a new trend that more and more low-end cameras, such as customer-grade digital cameras, PDAs, PC cams and cellophane cameras, are also employed as an acquisition tool to capture text images, because they are easy to carry around. We are particular interested in these images captured by low-end cameras with a casual manner (think about how a cashier uses a barcode scanner). **In the**

[3] http://www.4digitalbooks.com/scan2pages/Scan2Pages.htm

rest of this thesis, both camera-based document images and real-scene images refer to images taken by low-end cameras.

Imaged text captured by low-end cameras may suffer from many degradations: low resolution, uneven lighting, complex background[4], and motion blur, as well as various geometrical deformations: perspective distortion, warping, and wide angle lens distortion.

Because of the complex background of camera-based images, locating text is non-trivial. Figure 2.2 shows the text locating result of a real-scene image, where text regions are marked by dashed boxes. Text locating is a extensively studied topic. Text locating competitions were held in year 2003 [LPS+03] and 2005 [Luc05] by ICDAR.

For a camera-based image, a geometrical normalization step [ZTF04, ZYT07, MBLH05, CM04] to remove deformations is dispensable. Sometime an enhancement step [CSB01] is also necessary to improve the degradation. As a matter of fact, the prevailing research on camera-based image processing is to normalize and enhance the image, in order to make it acceptable for existing textual information extraction techniques. After these steps, the image is ready for textual information extraction.

As introduced in the last section, OCR, Word Shape Coding and Holistic Word Spotting are three textual information extraction techniques for scanned images. On the contrary, the only extraction technique that has been reported in literature for camera-based images is OCR. Because camera-based image processing has different downstream applications from scanned document image processing. For scanned

[4] More of the scene is imaged than the intended text.

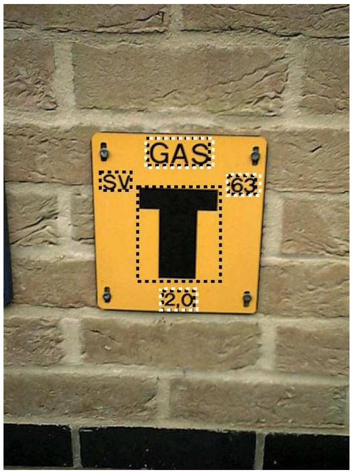


Figure 2.2: Locating text regions of a real scene image (the figure is from [LPS+03]).

images, textual information is extracted mainly for image archive management. However, for camera-based images, possible applications are much more diverse. Before a camera-based image can be utilized in an application, typical three steps are conducted, namely rectification, enhancement, and OCR.

Applications for Camera-base Image Processing

Low-end cameras are also employed to capture document images, for faxing, note taking, etc. Because a camera can be conveniently carried anywhere by a user to record interesting document pages instantly. A prototype based on PDA is presented[5]. In order to recognize text of camera-based document images, many methods have been proposed to remove the perspective distortion [Pil01, LCK05] and enhance the image quality [PP02].

The applications for real-scene images processing is even more various. Traffic sign recognition [dlEMSA97, LL95] is implemented in Driver Support Systems [ESS+94]

[5] http://www.hpl.hp.com/news/2002/apr-jun/translator.html

to recognize the traffic signs put on the road e.g. “slow”, “school ahead”, or “turn ahead”. The application improves road safety by informing the driver to go slow or take a turn. Another one application is license plate recognition [CFGS95, CCCC04, YMMN05], which is practically useful in parking lot billing, toll collecting monitoring, road law enforcement, and security management. Cargo container code recognition systems [LK95] are used in ports to automatically read cargo container codes for cargo tracking and allocation. Sign recognition systems or translation cameras recognize images of signs captured by a portable camera. They can help international tourists to overcome language barrier [WOKT98, YGZ$^+$01].

In these different applications, the degree of difficulty is reduced, because each of them deals with a different subset of degradation and deformation and extra non-shape information is available too. For example, traffic sign recognition makes use of a full range of features including color, shape, and texture; license plate recognition and cargo container recognition take the advantage of the limited alphabet as well as fixed format. Sign recognition is the one application that suffers most from perspective deformation, because shape is the only reliable feature. For this application, the key difficulty is in the concise nature of signs: a sign is often comprised of only a few words/characters. In the image processing phase, it will cause problems in sign detection and character recognition, because the prevailing technologies are designed for large text segments. More importantly, because of geographical constraints (such as when the sign hang high above) and the close distance between a sign and the camera, sign images often have a severe perspective distortion.

2.3 Linear Geometric Deformation of Images

In this thesis, we are particularly interested in imaged text with geometric deformations. When an image undergoes a geometric transformation, some or all of the pixels within the source image are relocated from their original spatial coordinates to a new position in the output image. In other words, the original image is distorted in some way. It is pointed out in [LDL05] that, for camera-based images, even small perspective distortion will cause significant trouble for OCR; and for flatbed scanners, rotation (skew) is the primary problem.

In this section, I will introduce the challenges of geometric deformed images, and existing rectification solutions, together with background knowledge of linear geometric deformation theory.

A two-dimensional linear geometric deformation has a basic form:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = T \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.3.1}$$

where (x, y) is the coordinate of a pixel in the source image, while (x', y') is the coordinate of the pixel in the deformed image, and T is a transformation matrix.

2.3.1 Skew of Scanned Document Images

A transformation of the plane that preserves shapes is called a Euclidean transformation. Under this transformation, lines transform to lines, planes transform to planes, circles transform to circles, and ellipsoids transform to ellipsoids. Euclidean transformation E has a form as:

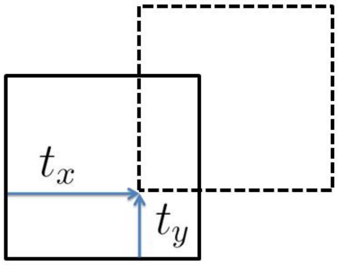


Figure 2.3: Translation.

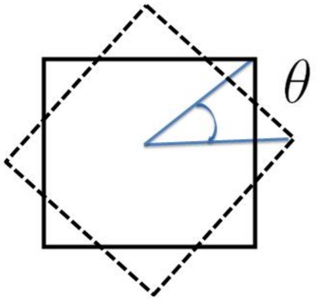


Figure 2.4: Rotation.

$$E = \begin{bmatrix} scos\theta & ssin\theta & t_x \\ -ssin\theta & scos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.3.2}$$

Transformation E can be treated as an accumulation of a sequence of sub-transformations: two translations along two axes by t_x and t_y respectively(Figure 2.3), a rotation by θ (Figure 2.4), and an isotropic scale by a factor s (Figure 2.5). In these figures, the dashed squares are transformed from the solid squares with corresponding parameters.

For a scanned document image, Euclidean deformation is caused when people fail to align the paper properly in a scanner. In particular, the rotation deformation is named as skew. Skew makes it more difficult to visualize of images by human users, increases the complexity of any sort of automatic image recognition, and degrades the performance of OCR tools, etc. A document image with skew is shown in Figure

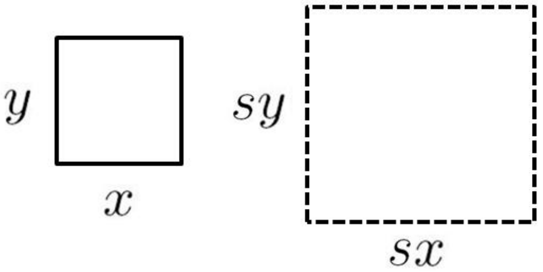
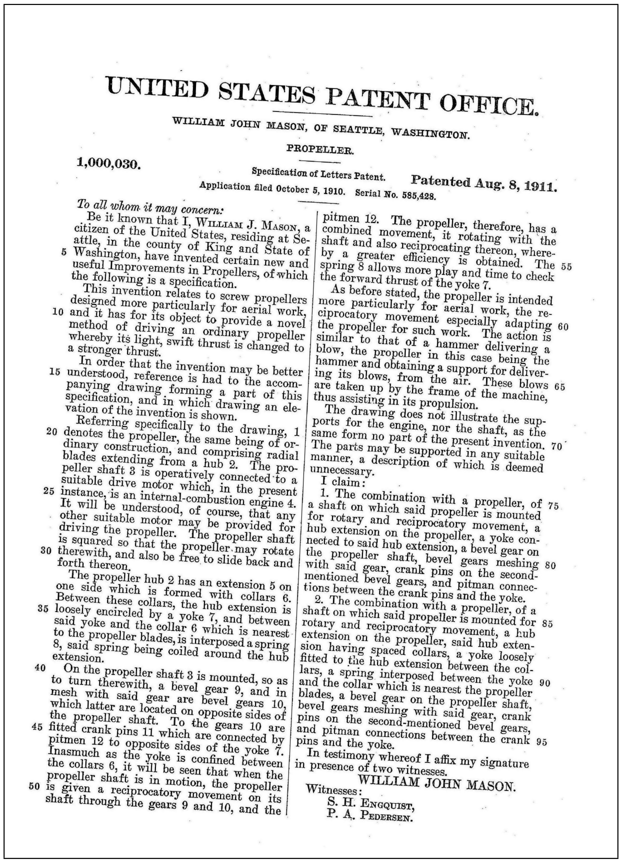


Figure 2.5: Scale.

5.5 (note that the text lines are not parallel to the edge of the image).

Skew Detection

The most popular strategy of handling skew is to remove it by skew detection methods. Skew detection methods can be mainly categorized into five groups: the ones based on Hough transformation, cross correlation, projection profile, Fourier transformation and nearest neighbors clustering. For approaches based on Hough Transform [SG89, Hin90, YJ96, AF00], the Hough Transform is computed at all angles between 0 and 180 degrees. A heuristic measures the rate of change in accumulator values at each degree. The skew angle is set to the degree that maximizes the heuristic. Approaches based on cross correlation [Yan93] use the cross correlation between the text lines at a fixed distance which is based on the fact that the correlation between vertical lines in an image is at a maximum for a skewed document. Approaches based on Fourier transformation [Pos86] take the skew angle as the angle of the direction in which the density of Fourier space is the maximum. For approaches based on projection profile [CSD$^+$88, SIR99], numbers of projections are obtained. The projection which has the minimum entropy gives the skew angle. Approaches based on nearest neighbors [YNF90] collect with a histogram the angle defined by two centroids of characters



UNITED STATES PATENT OFFICE.

WILLIAM JOHN MASON, OF SEATTLE, WASHINGTON.

PROPELLER.

1,000,030. Specification of Letters Patent. Patented Aug. 8, 1911.

Application filed October 5, 1910. Serial No. 585,428.

To all whom it may concern:

Be it known that I, WILLIAM J. MASON, a citizen of the United States, residing at Seattle, in the county of King and State of Washington, have invented certain new and useful Improvements in Propellers, of which the following is a specification.

This invention relates to screw propellers designed more particularly for aerial work, and it has for its object to provide a novel method of driving an ordinary propeller whereby its light, swift thrust is changed to a stronger thrust.

In order that the invention may be better understood, reference is had to the accompanying drawing forming a part of this specification, and in which drawing an elevation of the invention is shown.

Referring specifically to the drawing, 1 denotes the propeller, the same being of ordinary construction, and comprising radial blades extending from a hub 2. The propeller shaft 3 is operatively connected to a suitable drive motor which, in the present instance, is an internal-combustion engine 4. It will be understood, of course, that any other suitable motor may be provided for driving the propeller. The propeller shaft is squared so that the propeller may rotate therewith, and also be free to slide back and forth thereon.

The propeller hub 2 has an extension 5 on one side which is formed with collars 6. Between these collars, the hub extension is loosely encircled by a yoke 7, and between said yoke and the collar 6 which is nearest to the propeller blades, is interposed a spring 8, said spring being coiled around the hub extension.

On the propeller shaft 3 is mounted, so as to turn therewith, a bevel gear 9, and in mesh with said gear are bevel gears 10, which latter are located on opposite sides of the propeller shaft. To the gears 10 are fitted crank pins 11 which are connected by pitmen 12 to opposite sides of the yoke 7. Inasmuch as the yoke is confined between the collars 6, it will be seen that when the propeller shaft is in motion, the propeller is given a reciprocatory movement on its shaft through the gears 9 and 10, and the pitmen 12. The propeller, therefore, has a combined movement, it rotating with the shaft and also reciprocating thereon, whereby a greater efficiency is obtained. The spring 8 allows more play and time to check the forward thrust of the yoke 7.

As before stated, the propeller is intended more particularly for aerial work, the reciprocatory movement especially adapting the propeller for such work. The action is similar to that of a hammer delivering a blow, the propeller in this case being the hammer and obtaining a support for delivering its blows, from the air. These blows are taken up by the frame of the machine, thus assisting in its propulsion.

The drawing does not illustrate the supports for the engine, nor the shaft, as the same form no part of the present invention. The parts may be supported in any suitable manner, a description of which is deemed unnecessary.

I claim:

1. The combination with a propeller, of a shaft on which said propeller is mounted for rotary and reciprocatory movement, a hub extension on the propeller, a yoke connected to said hub extension, a bevel gear on the propeller shaft, bevel gears meshing with said gear, crank pins on the second-mentioned bevel gears, and pitman connections between the crank pins and the yoke.

2. The combination with a propeller, of a shaft on which said propeller is mounted for rotary and reciprocatory movement, a hub extension on the propeller, said hub extension having spaced collars, a yoke loosely fitted to the hub extension between the collars, a spring interposed between the yoke and the collar which is nearest the propeller blades, a bevel gear on the propeller shaft, bevel gears meshing with said gear, crank pins on the second-mentioned bevel gears, and pitman connections between the crank pins and the yoke.

In testimony whereof I affix my signature in presence of two witnesses.

WILLIAM JOHN MASON.

Witnesses:
S. H. ENGQUIST,
P. A. PEDERSEN.

Figure 2.6: A document image with skew.

which are nearest neighbors, and the main peak of the histogram indicates the skew angle.

2.3.2 Perspective Deformation of Camera-based Images

Affine transformations are generalizations of Euclidean transformations. An affine transformation can be further decomposed into an Euclidean transformation E, a shear k (Figure 2.7), and a non-isotropic scaling of one axis, which means that there is no scaling of the other axis, by a scaling a factor b, as shown in Equation 2.3.3.

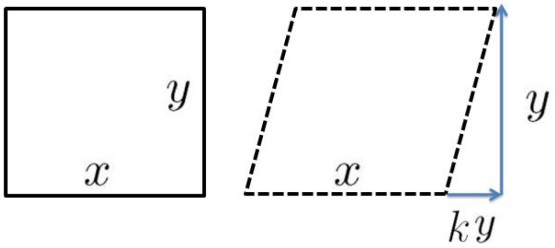


Figure 2.7: Shear.

$$A = E \begin{bmatrix} 1/b & -k/b & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.3.3}$$

Under an Euclidean transformation, the shape of a geometric object will not change. Only the position and orientation of the object will change. Under an affine transformation, however, a shape will change, i.e. a square to a diamond. Because although parallelism[6] is preserved under affine transformation, angles are not preserved here.

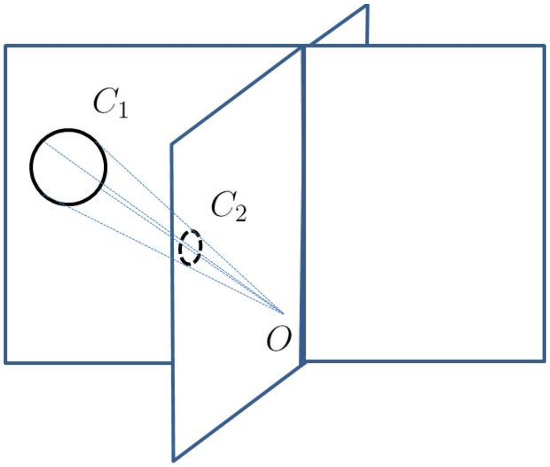


Figure 2.8: A perspective transformation with center O, mapping the circle C_1 on a plane to the ellipse C_2 on another plane.

[6] Parallelism means to map parallel lines to parallel lines.

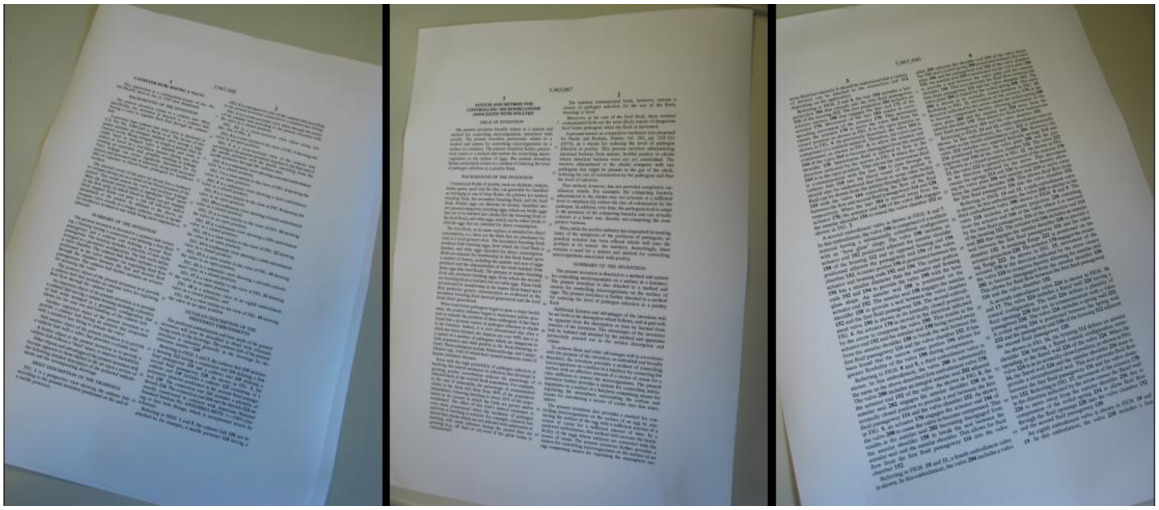


Figure 2.9: A document image with perspective deformation.

Perspective transformations, as shown in figure 2.8, are the most general linear transformations, under which parallelism is not preserved. Theoretically there is always a perspective transformation which can project one closed shape to another. A perspective transformation matrix P can be further decomposed into an affine transformation A and perspective foreshortening along the two axes l_x, l_y:

$$P = A \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_x & l_y & 1 \end{bmatrix} \tag{2.3.4}$$

When l_x and l_y is small, such as when the distance between the object and the camera is greater than the size of the object itself, the perspective transformation can be approximated by an affine transformation.

Perspective deformation in a camera-based image is introduced in by the fact that the image plane in a camera is not parallel to the object plane. It is one the major

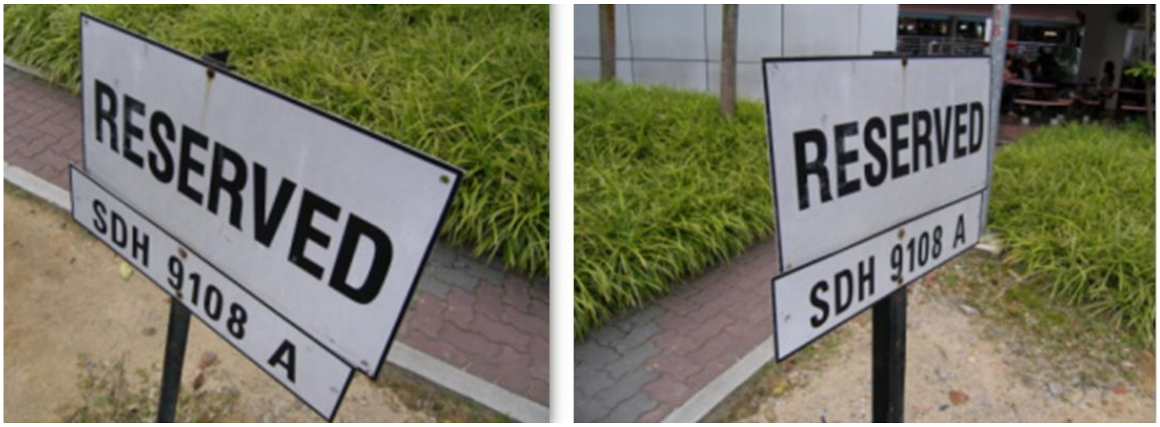


Figure 2.10: Perspective deformation in real scene images.

problems which prevents adapting existing textual information extraction techniques for scanned images to camera-based images. As shown in Figure 2.9, if we consider the deformation in a more straight forward way, the challenges of perspective transformation for camera-based document image processing are as follows:

- The skew of a camera-based image is more severe and unpredictable than that of a scanned image.
- Text lines in an image are no longer parallel to each other, and thus the skew angle of each text line is different from each other.
- Characters within the same text-line no longer remain the same height, namely, characters near the camera lens are larger than those further away.

For real scene images, the deformation become even worse. A character itself is perspectively distorted. Some parts of the character expand, while some parts shrink, as shown in Figure 2.10.

Perspective Rectification

In order to resolve the problem of perspective deformation, a popular solution is to reconstruct the fronto-parallel view, and then recognize symbols. This methodology is often used for camera-based document images. Jelinek et al. [JT01] proposed a rectification method making use of vanishing points to estimate the perspective projection matrix, which has been widely adopted. In order to estimate the vanishing point, methods of finding clues were proposed. Pilu [Pil01] proposed a method to find the vanishing points based on illusory horizontal clues (i.e. text lines) and vertical clues (i.e. column edges). Besides, Clark et al. [CM04] found that the quadrilateral edges of the text area were also useful features to locate the vanishing points. In Lu's method [LCK05], tip points and the vertical stroke boundary of a character were used. However, the presence of clues is highly application-dependent, and thus these methods are difficult to be applied to imaged text other than denouements. For example, these methods assume that the text body has sufficient number of text lines and that the layout is highly formatted, and thus they are not suitable for real-scene word recognition, whose text may comprise of only a few text lines, or even only a few characters. For images with less text present, a solution is to approximate the perspective transformation by an affine transformation, because an affine transformation has fewer free parameters. Mayer et al. [MBLH05] proposed a method to rectify individual text lines, assuming a weak perspective distortion in the vertical direction. However, note that the approximation only holds when the distance between the object and the camera is much greater than the size of the object.

Besides slowing down speed and introducing errors, another disadvantage of the rectification strategy is its format dependence. For example, methods employed in camera-based document image processing assume that the text body has sufficient number of text lines and that the layout is highly formatted, and thus they are not suitable for real-scene word recognition, whose text may comprise of only a few text lines, or even only a few characters.

Chapter 3

A Word Shape Coding Scheme for Scanned Document Images

Motivated by the rapid development of electronic information technologies, many projects have been initiated to digitalize paper-based materials. For example, the U.S. patent database [1] has scanned more than 4,000,000 imaged patent documents in the period 1790-1975. The European patent database has scanned images from patents before 1970, estimated to be several terabytes in size. Another category of paper materials which draws interests is historical books and journals. ProQuest Historical Newspapers (PQHN), which offers a full-image archive of the newspapers New York Times (1851-2001), The Wall Street Journal (1889-1987), The Washington Post (1887-1988), and The Christian Science Monitor (1908-1991). The Los Angeles Times and Chicago Tribune are currently going through the digitization process. JSTOR provides an interdisciplinary image-document archive of over 600 journals in the arts, humanities, and social sciences. American Periodical Series Online (APS Online) contains digitized images of the pages of American magazines and journals that originated between 1741 and 1900. Early English Book Online (EEBO) stores

[1] http://www.uspto.gov/patft/help/contents.htm

the full images of 125,000 early English books from 1475 to 1700. The benefits are obvious: information stored electronically requires less space, and it is much easier to be duplicated and delivered. Convenience of access is also not tied to the physical proximity of materials any more. Consequently, document image indexing and retrieval has become a growing and challenging problem, because traditional full-text information retrieval techniques totally fails when documents are simply presented as raw bit-maps.

One drawback of OCR is that it is computationally intensive, and thus is not suitable when the application itself is time critical. Another drawback is that OCR is language dependent, which makes it fail when the language is unknown. An alternative technique, namely Word Shape Coding, was proposed to fill up the deficiency. In this chapter, I will introduce a word shape coding method, and then explain how to employ it in language identification, Boolean document image retrieval, and document image filtering applications respectively. Related work can be found in Section 2.1.2 of Chapter 2.

3.1 A Fast Word Shape Coding Scheme

In this section, we propose a word shape coding scheme which can be swiftly extracted from scanned document images. The coding scheme is stroke-based, and has 8 codes in total. A word image is firstly decomposed into a sequence of strokes. A shape code is assigned to each stroke. The decomposition of word image into strokes depends on the pixels lying on the middle line as shown in Figure 3.1(a). If a pixel on the middle line is OFF, all pixels in the column are turned OFF, and otherwise pixels remain the same state. Figure 3.1(b) is an example where the phrase “keyword spotting” is decomposed into strokes. Each stroke is separated by several blank columns. In real applications, imperfect printing or scanning causes word images to have blurred intersection, and therefore the strokes are too wide to be decomposed by pixels on the middle line. In this case, we use a middle zone instead of the middle line. The middle zone is defined as a rectangular zone lying on the middle line, with a width equal to half an average vertical stroke width. Vertical stroke widths are collected by horizontal run length analysis, and the average vertical stroke width is estimated by finding the peak in the histogram.

In order to detect ascender and descender features, the top line, x-line, baseline and bottom line (shown in Figure 3.1(a)) are extracted for each text line by examining the horizontal projection. These four lines define the boundaries of three significant zones on each text line. The area between the bottom and the baseline is the descender zone; the area between the baseline and the x-line is the x-zone; and the area above the x-line is the ascender zone. The coding of strokes is based on the presence of straight vertical/non-straight vertical strokes, ascender and descender and the number of components, as shown in Table 3.1 (“1+” in the column “Number of Components” means “more than one component”). Examples of encoded strokes are shown in Figure 1(b).

Each Latin character is represented by a code string. Table 3.2 shows a list of Latin characters and their corresponding word shape code strings. It is worth noting that, in different font character ‘g’ may have different code strings. For example, character ‘g’ in Times New Roman is coded as 11, while ‘g’ in Arial it is coded as 14. Since no valid codes for any word contain 11 except words with character ‘g’ of

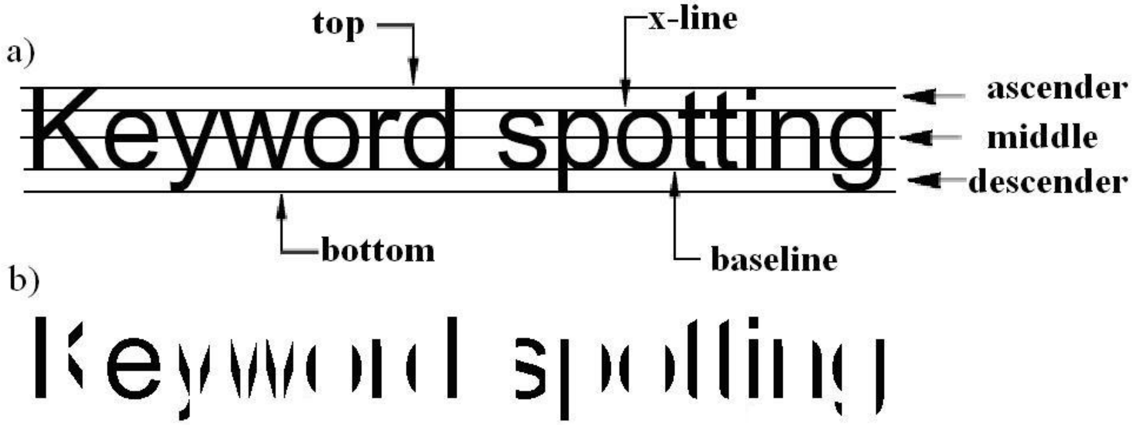


Figure 3.1: (a) A word image showing the text line parameter positions: top, x-height, baseline, and bottom, and the zones defined by them. (b) Decompose "keyword spotting" into strokes and encoded them.

Roman, it will not affect the final coding performance. The code string for a word comprise of code strings for each character in order. For example, the code string for "keyword spotting" is "62212222222526 24226675514".

3.1.1 Collision Rates

Since word shape coding methods maps characters to a reduced symbol set, ambiguity occurs, which means that several words share the same word shape coding string. SPITZ's coding method fails in Boolean retrieval application, while LV's method has a good performance. The key is the ambiguity level of the coding method.

Here we quantify the ambiguity of a coding method by a measure called collision rate developed by us. The collision rate is defined as the difference between the

Table 3.1: The mapping of strokes to shape codes Codes.

Code	Vertical straight	Ascender	Descender	# of Components	The example in figure
1	NO	YES	NO	1+	(1)
2	NO	NO	NO	1+	(2)
3	NO	NO	YES	1+	(3)
4	YES	YES	NO	1	(4)
5	YES	NO	NO	1	(5)
6	YES	NO	YES	1	(6)
7	YES	YES	NO	2	(7)
8	YES	YES	YES	2	NA

number of words and the number of corresponding identical code strings over the number of words as the formula below:

$$collision\ rate = \frac{\#\ of\ words - \#\ of\ code\ strings}{\#\ of\ word\ strings} \tag{3.1.1}$$

The collision rate of 4 Latin languages are shown in Tables 3.3 and 3.4. In particular, Table 3.3 is evaluated based on stops words of these 4 languages. Stop-word lists are provided by CLEF[2]. The off-diagonal items of Table 3.3 show the pair-wise “overlapping” between stop-word lists of these languages. It shows that collision rates of different language pairs range from 0.2% to 5.0%. Note that language identification methods based on word shape coding techniques identify the language identity by checking the frequency of stop words of a specific language in the document. Therefore, the result indicates that the coding scheme will work well for language identification.

Table 3.4 is evaluated based on non-stop words, which are obtained from filtering

[2] http://www.unine.ch/info/clef

Table 3.2: The codes for characters in Latin-1.

Characters	Codes	Characters	Codes	Characters	Codes
a	25	h	65	H	66
iìíî	7	m	555	ceszYç	2
nu	55	d	26	CSZ	3
fltEFLPT	6	BDKR	63	p	42
M	6226	q	24	J	3
âä	37	r	5	w	222
N	626	GOQUXÔôéèê	33	W	2222
bk	62	j	i	OvAVÄ	22
g	14/11	USS	33	y	12
ûü	77				

stop words away from a lexicon. Diagonal items of the table indicates the potential performance of Boolean retrieval with the code scheme. The collision rates for these 4 languages lie between 1% and 4%. This ensures our word shape coding scheme is to adequate for Boolean retrieval because it causes only a little ambiguity.

Table 3.3: The collision rate of the proposed word shape coding scheme between stop words of the same and different languages.

	English	French	German	Italian
English (571 words)	0.0861	0.0566	0.0216	0.0164
French (463 words)	0.0566	0.1554	0.0117	0.0147
German (603 words)	0.0216	0.0117	0.0336	0.0029
Italian (430 words)	0.0164	0.0147	0.0029	0.0507

For comparison purpose, the collision rates for three other word shape coding schemes, namely LV's, TAN's, and SPITZ's are shown in Tabel 3.5. The inherited ambiguity makes TAN's and SPITZ's coding schemes not suitable for Boolean retrieval. Our coding scheme has a collision rate of 0.0619%, and LV's coding scheme

Table 3.4: The collision rate of the proposed word shape coding scheme between non-stop words of the same and different languages.

	English	French	German	Italian
English (55900 words)	0.0096	0.0255	0.0085	0.0045
French (34696 words)	0.0255	0.0418	0.0065	0.0145
German (72815 words)	0.0085	0.0065	0.0107	0.0025
Italian (9127 words)	0.0045	0.0145	0.0025	0.0115

has even lesser ambiguity.

Table 3.5: The collision rate for four word shape coding schemes.

Coding Schemes	Stop Words (120)	Non Stop Words(54880)	n times faster than OCR
TAN's	0.4622	0.4231	2-6
SPITZ's	0.4351	0.2366	2-3 orders of magnitude
Ours	0.2304	0.0619	20-40
LV's	0.0083	0.0023	3

3.2 Applications

3.2.1 Language identification

A template vector is built for each candidate language. When a new query document image comes in, a query vector of the document image is created. We calculate the similarity between the query and each template. The language identity of the query is the one whose template has the highest similarity with it.

In order to construct the template and query vector, a subset of the most frequently-used 200 words of each language are chosen. The template and query vectors are built in a way that each dimension of a vector represents a unique word shape code string, and the value is the normalized string frequency. Language templates are trained on 40 documents. Similarity between the query and a template is defined as:

$$sim(Q,T) = \frac{\sum_{i=1}^{200} Q_i \times T_i}{\sqrt{\sum_{i=1}^{200} Q_i^2} \times \sqrt{\sum_{i=1}^{200} T_i^2}} \tag{3.2.1}$$

where Q represents the query vector, and T represents a template vector.

The performance of the proposed language identification method have been tested. 80 documents are prepared. We have documents in English, French, Italian and German (20 documents for each). Each document contains at least 15 text lines each and texts within them, and is printed in a font chosen from Arial, Roman, or Verdana, which are popular fonts.

In our experiments, languages of all testing documents are correctly determined. Table 3.6 shows the average similarity between document vectors and the template vectors of the same and different languages. The similarities between the same language (diagonal items) are much higher than those of different languages (off diagonal items). For these 80 testing documents, three document set with image degradation are created, namely Gaussian noise set ($\sigma = 0.08$), pepper-salt noise set (corruption percentage $= 0.06$) and low scanning resolution set (150 ppi).We have gotten an identification accuracy of 97.50%, 90%, and 87.5% for the 3 sets respectively. Table 3.7 shows the encoding accuracy. Over 90% word images are correctly encoded in the presence of various types of noise and document degradation.

Table 3.6: The similarity between document vectors of same and different languages.

	English	French	German	Italian
English	0.8802	0.2728	0.2005	0.2148
French	0.3638	0.8233	0.1952	0.3501
German	0.2696	0.2454	0.8444	0.2146
Italian	0.3076	0.3273	0.2218	0.9181

Table 3.7: The coding accuracy of the proposed word shape with image degradation.

Gaussian	Pepper-Salt	Low Resolution
0.9527	0.9056	0.9224

3.2.2 Boolean Document Image Retrieval based on Single Keyword Spotting

We found that encoding errors do occur when the text line is too short, which will degrade the retrieval performance. In order to address this problem, a method to estimate the similarity between two code strings based on an approximate matching algorithm is proposed as below:

$$sim(query, str) = 1 - \frac{MinEditDis(query, str)}{10 \times \lg(query.length)} >= \delta \tag{3.2.2}$$

where $MinEditDis$ is the function to calculate Levenshtein distance [Gus97]. Levenshtein distance between two strings is given by the minimum number of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a single character. In our experiment, cost for each operation equals to 1, and hence the return value of $MinEditDis(query, str)$ is a non-negative integer. The formulation indicates that a higher δ allows smaller edit distance, which means strings are more similar and vice versa.

The testing dataset has 300-dpi binary images scanned from 23 printed pages, each of which has a randomly chosen character size from 10 to 18 points, and a font chosen from Times New Roman, Arial, Dotum or Century Gothic. This experiment demonstrates whether the coding scheme will work well on different fonts and sizes under a controlled situation. Because the coding is sensitive to skew, images must first be de-skewed. Manual zone information is provided. In case there are headlines, footnotes or drawings zones, they are removed from the images. In each text zone, word bounding boxes are extracted by examining the projection profile. Each query will be translated into two code strings: all lowercase and capital initial.

There are 23 pages, having 174221 words in total. After all pages are tokenized, the approximate matching algorithm is used to compare queries and strings in the document. 50 keywords were generated, which appear at least 15 times in the dataset. There are a total of 1845 appearances for all the keywords. The precision of spotting one keyword is defined as the number of correctly detected words over the number of detected words, while the recall of spotting one keyword is defined as the number of correctly detected words over the number of actual keyword occurrence.

Table 3.8: Keyword spotting performance.

δ	0.7	0.88	0.9	0.92	1
Precision	0.3020	0.9294	0.9399	0.9622	0.9994
Recall	0.9601	0.9223	0.9133	0.9008	0.7205
F1	0.4100	0.9169	0.9173	0.9205	0.7612

The average recall and precision measure of the keyword spotting for several thresholds δ is shown in Table 3.8. It shows that the proposed word spotting method achieves the best precision/recall value as 96.22%/90.08%. On the other hand, OCR has a performance as 98.33%/96.08%. The result suggests that the proposed word spotting is robust to different fonts and character sizes. Besides, the table also shows it is very easy to adjust the precision/recall pair according to different application requirements. A lower threshold brings high recall but low precision, because the coding scheme itself causes ambiguity between different words. Approximate matching with a loose constraint will aggravate the situation. On the contrary, higher threshold brings high precision but low recall. When δ is around 0.9, the average F1 achieves the highest value. This may be because the threshold is low enough to compensate coding mistake when the coding scheme assigned a stroke a wrong code, while high enough to avoid bringing more ambiguity.

An important issue of keyword spotting is speed. As shown in Table 3.9, in the experiments it took 0.73 seconds to encode all images, while a commercial OCR spent 3.06 seconds to transcript these images. Hence, the keyword spotting is more than 20 times faster than OCR.

Table 3.9: Running time comparison for OCR and coding.

Dataset	# of pages	OCR time	Coding time
Self-prepared	23	3.06s	0.73s

3.2.3 Document Image Filtering

We set up an experiment to simulate a real document filtering task as follows. 50 profiles are built, each of which comprises of 20 to 30 keywords connected by Boolean operators. As the document stream proceeds, the incoming image is encoded by the proposed word shape coding scheme, and a binary decision is made based on keyword matching. If the keyword appears in a document, this document is considered as relevant. Similarity estimation between code strings is shown in Equation 3.2.2.

The testing data is ISRI DOE2&3 collections. ISRI DOE2&3 are public image collections from ISRI[3], which contains 1670 scanned pages as well as the associated ground truth text. These images are generated from various ways including directly scanning from journal pages, scanning from first or later generation photocopies. Therefore the dataset has unexpected fonts, character sizes, noise as well as skew. Also, a preprocessing to remove skew is necessary.

The average recall and precision of the document image filtering task for several thresholds δ is shown in table 3.10. The document filtering performance based on the proposed method achieves the best precision/recall value at 91.00%/77.01%, while the one based on OCR has a best performance at 93.7%/88%. The proposed method archives comparable accuracy, yet is 20 times faster than OCR.

Two factors that may affect the keyword spotting performance are the ambiguity and coding errors. The former means that a code string may map to several words and the later means that a wrong code is assigned to a stroke. Ambiguity brings higher recall but lower precision; on the contrary, encoding errors bring lower recall.

[3] http://www.isri.unlv.edu/ISRI

Table 3.10: The document filtering performance based on keyword spotting for ISIR DOE dataset.

δ	0.80	0.80	0.88	0.9	1
Precision	0.5928	0.8889	0.9100	1.0000	1.0000
Recall	0.8048	0.7804	0.7701	0.5612	0.5612
F1	0.6827	0.8289	0.8341	0.7189	0.7189

Table 3.11: Running time comparison for OCR and coding.

Dataset	No. of pages	OCR time	Coding time
ISRI DOE	1245	62min	3min

In Table 3.10, when the performance (F1) is best, precision (91%) is much higher than recall (77%). This phenomenon may indicate that the coding errors dominate the performance other than ambiguity.

As shown in table 3.11, it took 3 minutes to process all document images by the proposed method, while it took 62 minutes by OCR.

3.3 Summary

I have introduced a new word shape coding scheme in this chapter, which avoids the difficulties of separating touching adjacent characters in a word image and extensive computation during recognition. The experiment results show that it is very fast and yet generates comparable accuracy to OCR. It is promising method to act as an alternative to full-scale OCR in some document image retrieval applications.

However, there is still one issue which needs further exploration. It is the sensitivity of skew. In the experiment, a deskew process was employed before encoding document image. Otherwise, it will be difficult to detect the baseline and topline. Consequently, the position of the middle line will be wrong, and characters like 'a' and 'e' may have wrong code strings. In view of this, we proposed a word shape coding method which is invariant to rotation transformation. I will detail this method in Chapter 4, and explain its application to camera-based document image retrieval, where skew occurs more frequently than scanned document images.

Chapter 4

A Word Shape Coding for Camera-based Document Images

Recently, we have seen an increasing interest in adapting digital cameras to tasks related to document image analysis. Digital camcorders, digital cameras, PC-cams, and even cell phone cameras are becoming increasingly popular and they have shown their potential as an alternative imaging device. Although they cannot replace scanners, they are small, light, easily integrated with various networks, and are more suitable for many document capturing tasks in less constrained environments. These advantages are leading to a natural extension of the document processing community where cameras are used to image hardcopy documents, or natural scenes containing textual content [DLL03]. Camera-based document image processing introduces many new requirements that are not common with images acquired by scanner, including dealing with perspective. Hence, document image processing techniques developed specifically for scanned images may not work on camera-based images. In this thesis, we are particularly interested in handling perspective deformation.

One important strategy to deal with the perspective deformation is to recover the fronto-parallel text plane by estimating the transformation matrix. A review about the perspective rectification method has been made in Section 2.3.2 of Chapter 2. However, the disadvantage of the rectification strategy is that it is very slow, taking more time than text extraction itself, as well as being format-dependent.

In this chapter, I will introduce a word shape coding scheme proposed by us, which directly accesses the textual information of a camera-based document image. This method is invariant to affine transformation and thus is robust when the perspective deformation is moderate. The coding scheme will be applied to script identification and document image filtering respectively.

4.1 Related Work

A detailed review about language identification and fast document image retrieval based on word shape coding methods, designed for scanned document images, has been made in Section 2.1.2 of Chapter 2. Word shape coding methods are optimized for images produced by a flat-bed scanner, and thus feature extraction steps are based on an assumption that there is only Euclidean Transformation in an image. For example, ascenders and descenders, as shown in Figure 4.1, are two important features employed in many word shape coding methods, including TAN's, LU's, LV's, and SPITZ's. They are detected by checking peaks and valley of a horizontal projection profile of the text line. This method is only valid when parallelism is preserved. However, due to the perspective deformation, text lines in a camera-based document image may not be parallel to each other, and the sizes of characters in the same text line may not be homogeneous. Therefore, it is difficult to detect both features.

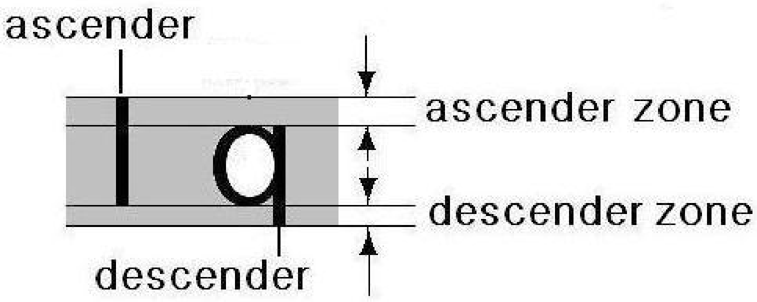


Figure 4.1: Ascender and descender.

Consequently, word shape coding methods based on these features will fail.

Besides word shape coding based methods, there are two other categories of language identification methods, namely texture based [BBS05] and component based [HKKT97]. In particular, Busch et al. [BBS05] investigated the use of texture like gray-level co-occurrence, energy, and wavelets to differentiate languages. However, since these features are not perspective invariant, it is obvious that the texture based approaches fail for camera-based document images. Hochberg et al. [HKKT97] proposed a method to identify the language by comparing characters of an unknown image to an exhaustive list of characters of a certain language, which is gotten from training document images by clustering. In this method, connected components were first extracted, and were scaled to a 30 × 30 pixel size. A hierarchy clustering algorithm, with Hamming distance as the distance measure, was employed to classify component images into clusters. The template comprised of the centroid of all clusters. The similarity between a template and a query was estimated by averaging the smallest Hamming distance between each components in the query document and the template. Based on a training process, Hochberg's method may stand a chance to

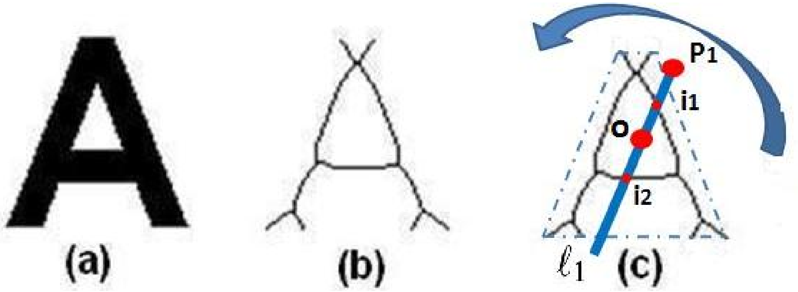


Figure 4.2: Signature generating process.

work, and thus it is employed as a benchmark in our script identification experiment.

4.2 A Word Coding Scheme for Camera-based Document Images

Theoretically, when the size of the perceived object depth is much smaller than the distance between the camera lens and the object, the perspective transformation can be approximated by an affine transformation. Now let's look at how a document image is captured by a camera. English characters printed on an A4 sheet is within a 2×2 mm^2 bounding box. In order to take a photo of the whole sheet, the distance between the camera and the projection center on the sheet is at least 30 mm. Because the image is taken for reading purpose, the camera projection angle is nearly perpendicular to the sheet plane, thus the object depth is very small. Hence, the affine assumption holds in this case. In fact, a similar assumption has been employed for rectifying camera-based document images under in previous studies [CM04, YMMN05].

The encoding process has two steps: signature generation and code assignment.

Signature Generation. Assume there is a component C, such as character 'A' shown in Figure 4.2(a), and the pixel sequence of the convex hull of C is $\{p_1, p_2, ..., p_n\}$ (shown as the dashed line in Figure 4.2(c)), where p_1 is an arbitrary pixel on the convex hull, and p_2 is the anti-clock-wise neighbor pixel of p_1, etc. The centroid of the convex-image of C is denoted by o. The signature of C is constructed as follows:

1. The centroid o of the convex image of C and the convex hull pixel sequence are first located. The skeleton of C, as shown in Figure 4.2(b), is obtained by a thinning operation.

2. The line ℓ_1 defined by o and p_1 is found, shown as the bar in Figure 4.2(c). There are two intersections between ℓ_1 and the skeleton of C, denoted by i_1 and i_2. Of course, there may be more than two intersections since p_1 is arbitrary. For each pair of these intersections, denoted by i_u and i_v, the length ratio λ_{uv} is calculated as:

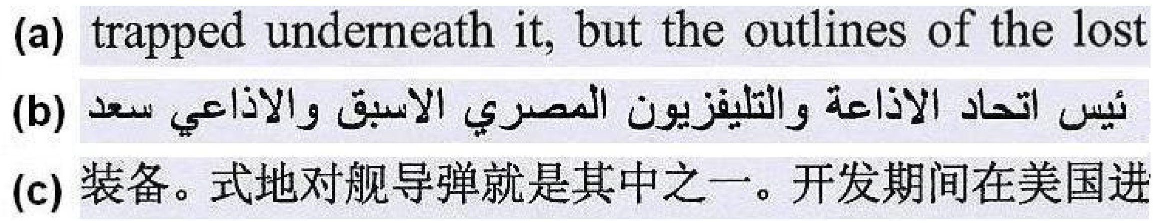
$$\begin{cases} \lambda_{uv} = \dfrac{oi_u}{oi_v},\ i_u\ and\ i_v\ are\ at\ different\ sides\ of\ o \\ \lambda_{uv} = -\dfrac{oi_u}{oi_v},\ i_u\ and\ i_v\ are\ at\ the\ same\ side\ of\ o \end{cases} \tag{4.2.1}$$

where oi_u and oi_v are the Euclidean distances between o, i_u, and i_v respectively.

3. Repeat step 2 on the remaining pixels $\{p_2, ..., p_n\}$. Actually, the bar is rotated 360 degrees around o. Length ratios are collected in the meantime.

4. A histogram is constructed for C to record the number of occurrences of length ratios λ_{uv}, if $|\lambda_{uv}| > 1$. In the experiment, we used a histogram starting with -5 and ending with 5, with n bins. In particular, bin i keeps a record of the number of length ratios within the range $(-5 + i \times \frac{10}{n}, -5 + (i + 1) \times \frac{10}{n}]$. The signature of C is gotten by normalizing the histogram.

Code Assignment. Training images are prepared, and signatures are extracted and classified into clusters by a hierarchy clustering algorithm, with cosine distance as the distance measure and a maximum radius of a cluster as clustering criterion.



(a) trapped underneath it, but the outlines of the lost

(b) ئيس اتحاد الاذاعة والتليفزيون المصري الاسبق والاذاعي سعد

(c) 装备。式地对舰导弹就是其中之一。开发期间在美国进

Figure 4.3: Examples of three languages: (a) English (b) Arabic (c) Chinese.

The distance can be empirically decided. The κ largest clusters are chosen. Each of them is then assigned a representative code, and the centroids of chosen clusters are referred as templates in the following paragraphs. Unknown characters are encoded by comparing their signatures with each template. The codes for the characters are given by the template which has the nearest distance with it.

According to affine geometry, the number of intersections defined by the projection of ℓ_i and the projections of the skeleton keeps constant, and the length ratio of line segments on a given line remains constant, when C is under affine distortion. Also, it has been proved that the centroid of a convex polygon preserves under affine transformation [GK07], namely, the affine projection of o remains the centroid of the affine projection of the convex image. As a result, the variation of the signature under different affine transformation becomes trivial.

4.3 Applications

4.3.1 Script Identification

In this subsection, I will explain how to apply our word shape coding method to script identification. In order to further differentiate languages using the same script, a methodology that has been adapted by many language identification methods can be employed in future. A typical language identification procedure based on Word Shape Coding technique is as follows. A list of the most frequently-used words in each language (often stopwords) is encoded into some kind of word shape code strings. When a document image comes in, it is also encoded. The encoded document is then compared with the list. The language identity of the document is the one whose list has the most agreement with the document.

Script Template Generating and Script Identification

We believe that our signature is effective in differentiating different scripts, because strokes of characters of different scripts have different complexity levels, which can be quantified by the number of intersections. Assume there is a vertical line passing thought the centroid of a character, and the number of intersections between the line and strokes of the character is t. t of English characters, shown in Figure 4.3(a), often ranges from 1 to 4; t of Arabic characters (Figure 4.3(b)) is often equal to 1; Chinese characters (Figure 4.3(c)) often have a larger t.

One template is generated for each candidate script. The script template is a frequency vector of signatures. For each script, a few training document images in fronto-parallel view are prepared. When clustering, 0.02 is chosen as the maximum radius of a cluster. A histogram of $n = 20$ bins is used. The $\kappa = 30$ biggest clusters are chosen, and a template comprises of the centroid and the size of the chosen clusters (the number of members in the cluster, denoted by $freq$). Both thresholds used here are empirically decided. An automatic learning procedure may be introduced in future. The format of a template is as follows:

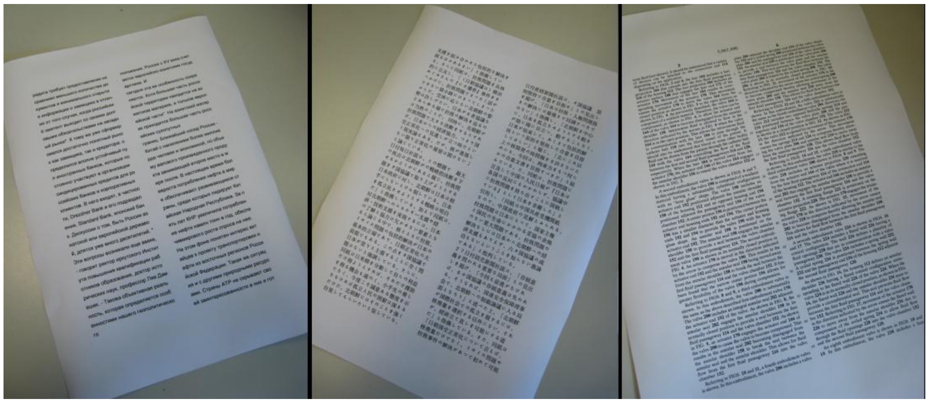


Figure 4.4: These photos are taken in a very casual manner. Some of them are with perspective deformation that is considered quite severe for this application. Our goal is to show that our method is robust to these extreme conditions.

$$\begin{bmatrix} signature_1 & freq_1 \\ signature_2 & freq_2 \\ ... & ... \\ signature_{30} & freq_{30} \end{bmatrix}$$

The similarity between a query document Q and a script template T is defined by:

$$sim(Q,T) = \frac{\Sigma freq(signature_i, Q) \times freq_i}{\Sigma freq_i} \tag{4.3.1}$$

$freq(signature_i, Q)$ is a function to find the frequency of $signature_i$ in Q. It works as follows: for each signature q_j of Q, if $signature_i$ is the nearest template signature of q_j, and the cosine distance between them is smaller than 0.02, the frequency of $signature_i$ in Q increases by 1. The script template which has the highest similarity with Q gives its script identity.

Experimental Setup

Ten scripts under study were 1: Arabic, 2: Chinese(simplified), 3: Cyrillic, 4: Greek, 5: Hebrew, 6: Japanese, 7: Korean, 8: Roman, 9: Thai, and 10: Bengali. These scripts are the most widely used scripts in the world. In the training data set, ten synthetic images in fronto-parallel view in each script were generated, in order to show that the method provides possibilities to train on only frontal-parallel images and construct a classifier which is able to identify the script of camera-based images. In the testing data set, photos (3072×2304 pixels) of ten printed images in each script were taken by a camera. Since it is natural that a printed paper has some warping, this distortion was also kept in the photo. The criterion of taking the photos is that, all characters in the photo should be recognizable to people. Examples of the testing data is shown in Figure 4.4.

Script Identification Results

Tables 4.1(a) and (b) show the confusion matrices of script identification results of our method and Hochberg's [HKKT97] method, respectively. An item in either tables is the number of documents in script i (ground truth) which were identified as script j (output). Correctly identified documents are not shown in this table. The proposed method was able to determine the scripts of testing images with 91% accuracy, while the baseline method was not able to deal with many of these images. We found that performance of Hochberg's method highly depended on the skew: it worked good on those images with small skew (within ±5°), but failed on those with severe skew. The performance of our method seems to be more independent of skew, but it made more mistakes on certain pairs of scripts.

Table 4.1: Confusion matrixes of ours and Hochberg's method.

(a) Our method

	Ground Truth									
Output	Arabic	Chinese	Cyrillic	Greek	Hebrew	Japanese	Korean	Roman	Thai	Bengali
Arabic										
Chinese						2				
Cyrillic										
Greek		1			1					
Hebrew								2		
Japanese		3								
Korean										
Roman										
Thai										
Bengali										0
Errors in total	0	4	0	0	1	2	0	2	0	0

(b) Hochberg's method

	Ground Truth									
Output	Arabic	Chinese	Cyrillic	Greek	Hebrew	Japanese	Korean	Roman	Thai	Bengali
Arabic		1								
Chinese	2		1			3			1	
Cyrillic				2						
Greek		1	2		1					2
Hebrew		3		1		1		3		
Japanese			1							
Korean										
Roman		1			3				2	2
Thai	2						1			
Bengali										2
Errors in total	4	6	4	3	4	4	1	3	3	6

Table 4.2: Cosine distances between pairs of script templates.

	Arabic	Chinese	Cyrillic	Greek	Hebrew	Japanese	Korean	Roman	Thai	Bengali
Arabic	0.0000	0.6680	0.4661	0.8437	0.7364	0.8008	0.9381	0.9840	0.6318	0.9218
Chinese	0.6773	0.0000	0.7051	0.5948	0.8042	**0.3611**	0.8678	0.9563	0.6330	0.9759
Cyrillic	0.4661	0.7051	0.0000	0.8723	0.7524	0.9366	0.9787	0.5166	0.7224	0.9678
Greek	0.9102	0.5948	0.8723	0.0000	0.9297	0.9297	0.6533	0.9942	0.6953	0.9747
Hebrew	0.7364	0.8042	0.7524	0.9297	0.0000	0.9777	0.9903	**0.2550**	0.4355	0.8896
Japanese	0.8008	**0.3806**	0.9366	0.6200	0.9777	0.0000	0.9841	0.8348	0.6838	0.9734
Korean	0.9381	0.8472	0.9787	0.5580	0.9773	0.9841	0.0000	0.9992	0.9830	0.9955
Roman	0.9816	0.9563	0.5412	0.9942	**0.1473**	0.8348	0.9992	0.0000	**0.3385**	0.9052
Thai	0.6258	0.7647	0.7224	0.6953	0.4658	0.7059	0.9830	**0.3385**	0.0000	0.9260
Bengali	0.9218	0.9759	0.9678	0.9747	0.8896	0.9734	0.9955	0.9421	0.9260	0.0000

The identification performance highly depends on the templates. If two templates are similar to each other, it is very likely that documents in one script are mistaken for another. Therefore, the similarity of templates are compared. Table 4.2 shows the cosine distance between each pair of template i and j. A diagonal item is the distance between a template and itself, thus equal to 0.

Table 4.2 shows that two groups of templates are similar to each other: the first group is 2: Chinese and 6: Japanese; the second one is 5: Hebrew, 8: Roman, and 9: Thai. This explains why errors often occur within both groups. The table also indicates that the performance can be improved by preparing more discriminating templates of these scripts.

Although the templates in the second group are closer to each other than those in the first group, errors occurred more frequently in the first group in the experiment. A possible reason is that Chinese and Japanese both have thousands of frequently used characters, and a template with 30 signatures is not enough to incorporate them. Increasing the size of templates for Chinese and Japanese may help with this problem.

It is worth noting that the number of bins is an essential parameter for a better performance. The binarization process may suffer from pixel quantization; the centroid computation and skeletonizing steps may be affected by noise. Hence length ratios may fluctuate accordingly. A signature with wide bins will be more tolerant to the fluctuation, but consequently it may have less discriminating power among characters. On the contrary, a signature with narrow bins is more discriminating, but it is more susceptible to noise.

4.3.2 Document Similarity Estimation

Document Image Representation

When an unknown document image is presented, each connected component in the image is compared with the templates trained. A histogram of $n = 20$ bins and $\kappa = 23$ biggest clusters are used. Its identity is assigned as the code of the most similar template. In the experiment, components smaller than 50 pixels were thrown away in order to avoid pixel quantization errors.

Word boundaries are found by Document-Spectrum analysis [O'G93]. Then "words" comprising of code string are formed. Thereafter, traditional vector space model with tf.idf representation is applied to those "words". Similarity between two document images is formulated as:

$$sim(D_u, D_v) = \frac{\sum D_{u,i} D_{v,i}}{\sqrt{\sum D_{u,i}^2 \sum D_{v,i}^2}} \tag{4.3.2}$$

where D_u and D_v are two documents, and i is the dimension index of the "word" space.

Experimental Setup

In order to test the efficiency of the proposed method, the experiment was set as follows: 100 pages were selected from 6 documents of the U.S. patent database. These documents were in different categories assigned by the database. We assumed that pages from the same document were similar to each other in content, and different otherwise. These pages were printed out, and 2304×3072 pixel photos were taken by a camera. Since it is natural that a printed paper has some warping distortion, this distortion was kept in these photos. We made sure that all characters in a image were readable. Examples of the testing data are shown in Figure 4.5. 10 photos were selected as queries. Each query was compared to other 90 photos by the proposed method. When the similarity between the query and an image was greater than a threshold θ, the image is retained, otherwise filtered. The training set comprised of 10 raw images from the same database.

Experiment Result

In the experiment, an average precision of 93.43% and an average recall of 94.22% were achieved, with $\theta = 0.3$. Table 4.3 shows the similarity across pages of the same document and that of different documents. Scores were calculated by averaging the similarity between each pair of pages from document i and j. Particularly, the cells on the diagonal are the similarity of pages within the same document. These items are much greater than those off-diagonal items.

An exhaustive study of this method will be done in the future. The high recall and precision in our experiment indicate that the technique is also promising for duplication detection.

4.4 Summary

I have presented a word shape coding method based on an affine invariant signature, which is able to directly access textual information of camera-based document images. To best of my knowledge, this method is the first to directly access the textual content of camera-based document images without geometric rectification.

In this method, we make an assumption that the perspective deformation is weak

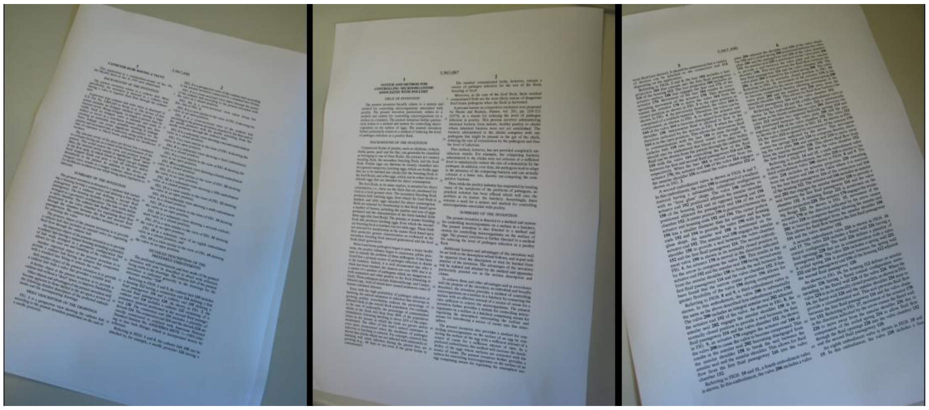


Figure 4.5: Samples of testing images.

Table 4.3: Similarity of the same and different documents. Items on the diagonal are average similarity among pages of the same document.

Doc.	1	2	3	4	5	6
1	**0.425**	0.079	0.062	0.106	0.117	0.050
2	0.079	**0.453**	0.108	0.192	0.023	0.151
3	0.062	0.108	**0.528**	0.088	0.117	0.175
4	0.106	0.192	0.088	**0.378**	0.066	0.188
5	0.117	0.023	0.117	0.066	**0.422**	0.138
6	0.050	0.151	0.175	0.188	0.138	**0.511**

and can be approximated by an affine deformation, because the character size is much smaller than the distance between the camera and the character. However, this assumption will not hold, if the character size is comparable to the camera distance. For example, people only take a small portion of the document. In this case, we have proposed a method to recognize character or symbol under severe perspective deformation. I will present this recognition method in Chapter 6.

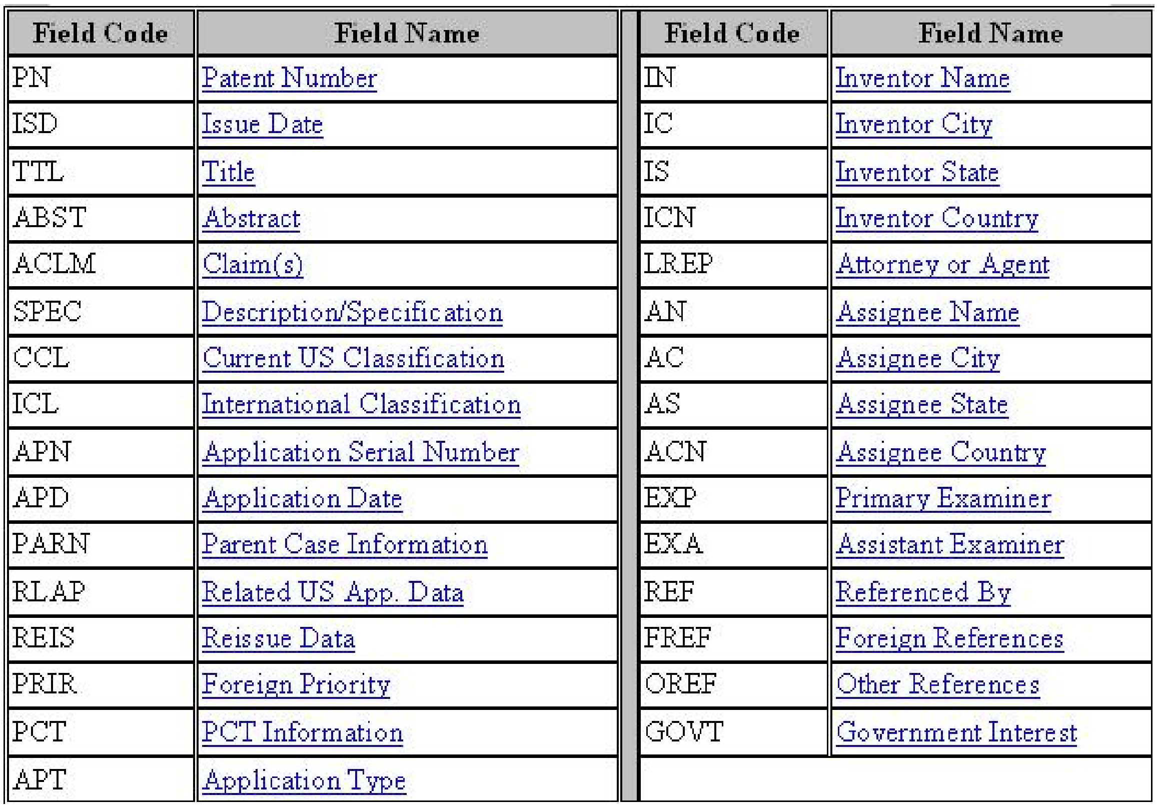
Two issues have not been addressed by this thesis and remain open for this coding method. First, two parameters of the proposed method, namely κ and n were empirically decided to maximize the performance for diffident applications in our experiments. The effect of both parameters on the performance of the proposed word shape coding method needs further evaluation. Also, because the ambiguity of the proposed method is unknown, we only employ the method in script identification and document similarity estimation applications, which are resilient to high ambiguity. A way to estimate the ambiguity may be further explored.

Chapter 5

Viewing Patent Images

In this chapter, I will present a fast rotation-invariant keyword spotting method and a method to locate text content of drawing especially for, but not constrained to, the U.S. patent database.

The U.S. patent database, maintained by the United States Patent and Trademark Office(USPTO), stores both patent text and patent images separately. The Web Patent Full-Text Database (PatFT) contains the full-text of over 3,000,000 patents from 1976 to the present, plus limited bibliographic data for over 4,000,000 patents from 1790 to 1975. The Web Patent Full-Page Images Database (PatImg) contains over 70,000,000 images, including every page of over 7,000,000 patents from 1790 to the most recent issue week. The Web Patent Databases now serves over 25,000,000 pages of text (over 150,000,000 hits) per month to over 350,000 customers each month. The Web Patent Databases serve over 36,000,000 full-page images each month. In order to facilitate better Intellectual Property managing, patents are divided into many fields (metadata) which are listed in Figure 5.1. Full-text retrieval is allowed for patents after 1976. However, due to unavailability of text version, patents from 1790 through 1975, whose total size is estimated at more than 20,000,000 pages, are



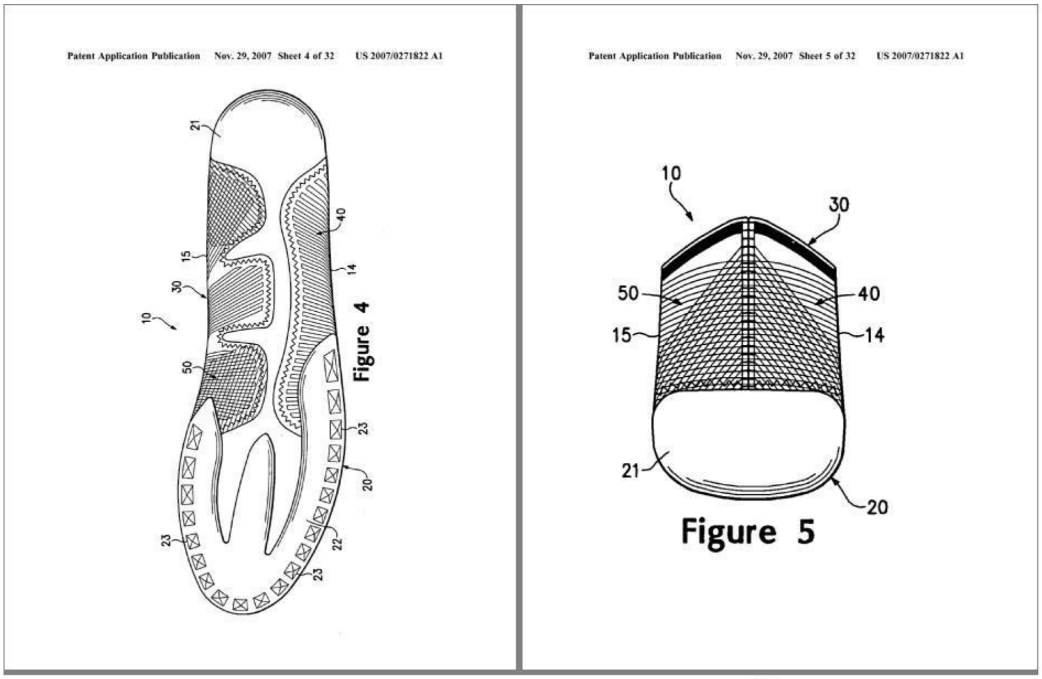
Field Code	Field Name	Field Code	Field Name
PN	Patent Number	IN	Inventor Name
ISD	Issue Date	IC	Inventor City
TTL	Title	IS	Inventor State
ABST	Abstract	ICN	Inventor Country
ACLM	Claim(s)	LREP	Attorney or Agent
SPEC	Description/Specification	AN	Assignee Name
CCL	Current US Classification	AC	Assignee City
ICL	International Classification	AS	Assignee State
APN	Application Serial Number	ACN	Assignee Country
APD	Application Date	EXP	Primary Examiner
PARN	Parent Case Information	EXA	Assistant Examiner
RLAP	Related US App. Data	REF	Referenced By
REIS	Reissue Data	FREF	Foreign References
PRIR	Foreign Priority	OREF	Other References
PCT	PCT Information	GOVT	Government Interest
APT	Application Type		

Figure 5.1: The list of all fields for each patent used in the U.S. patent database. The figure is taken from the homepage of the United States Patent and Trademark Office.

searchable only by Issue Date, Patent Number, and Current US Classification.

In the patent database, the format of a patent document is constrained: it is divided into five sections, including abstract, drawing, description, claim and reference sections, each of which occupies a few consecutive pages. For Each patent, except for those before 1970, a text version and an image version are available. The text version includes abstract, description, claim and reference sections, while the image version has all five sections.

Drawing pages have two types of orientation, namely landscape and portrait, as


Patent Application Publication Nov. 29, 2007 Sheet 4 of 32 US 2007/0271822 A1
21
15
30
10
50
40
14
Figure 4
23
20
22
23
23
Patent Application Publication Nov. 29, 2007 Sheet 5 of 32 US 2007/0271822 A1
10
30
50
40
15
14
21
20
Figure 5

(a) (b)

Figure 5.2: Two drawing pages: (a) a landscape drawing page. (b) a portrait drawing page.

shown in Figure 5.2. Figure 5.3 shows a typical drawing image. It has several figures, each of which has a caption and many labels. A label represents a particular part of the invention. An important characteristic of US patent database is that text pages, including abstract, description, and claim sections, are almost in the same font. However, the fonts of text content in the drawing parts are diverse, sometimes even written by hand.

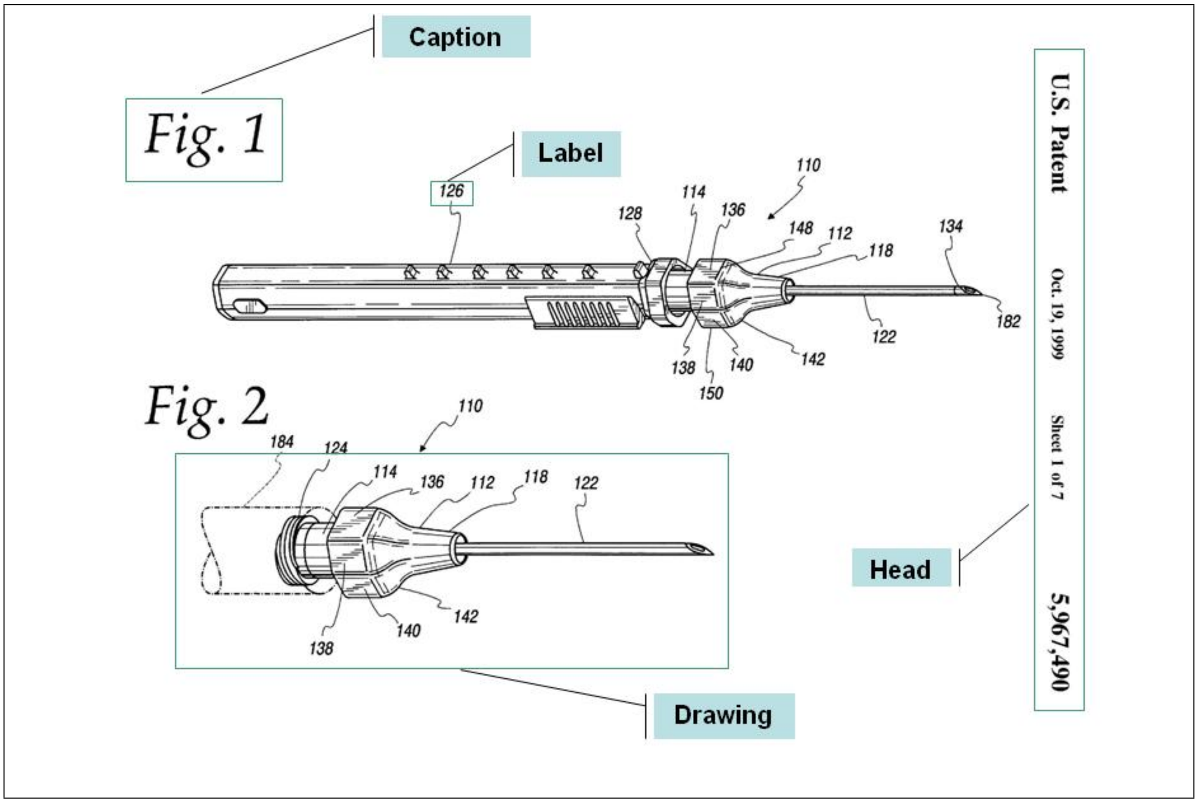


Figure 5.3: A drawing image of a patent document with several figures. A typical figure has a caption, drawings and several labels.

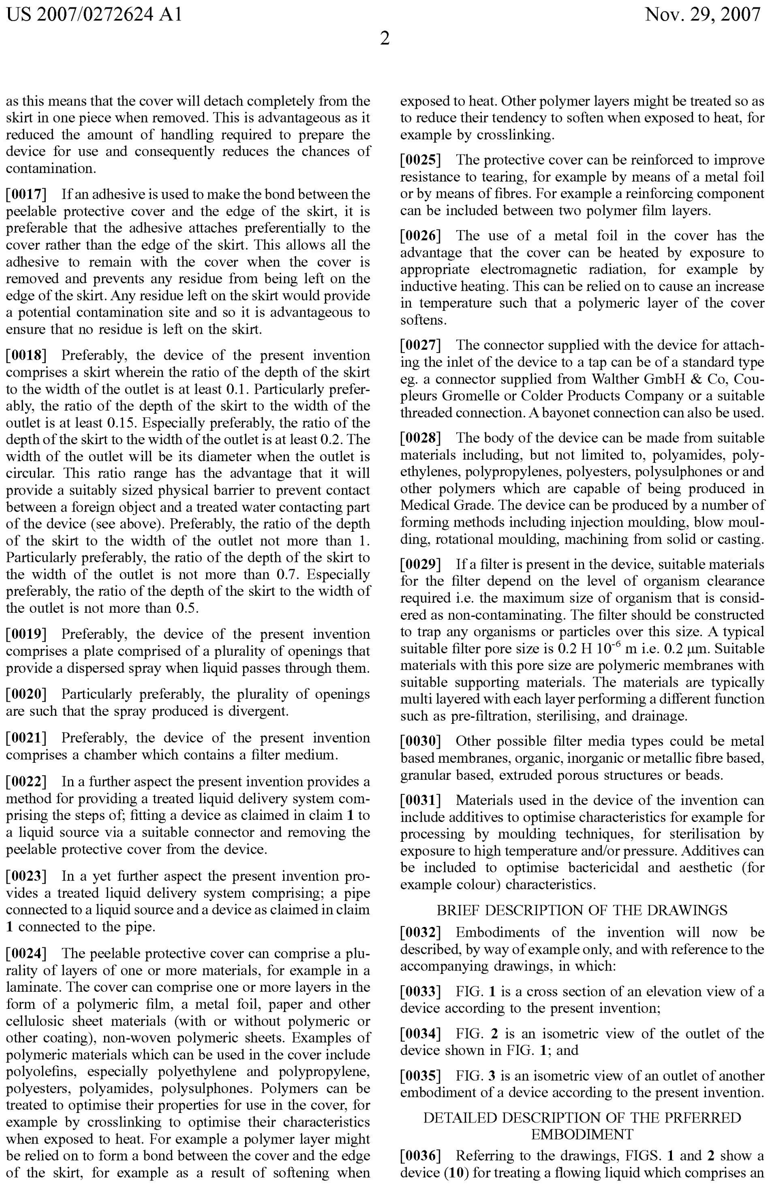
5.1 Problem Statement

Patents downloaded from almost all databases are not searchable, even for databases like Google Patents, which provides free online keyword spotting functions. Because the source of patents, namely the USPTO, maintains patent images and text separately. However, no connecting information between images and text, i.e. the mapping between the word image and its correspond word, is kept. Getting the connecting information by OCR is really expensive, and thus it is not publicly downloadable. This causes difficulty in local viewing for a user when he has downloaded the target patents to his PC, and then tries to find useful information in these particular patents. The local viewing step is also an essential step of the whole information retrieval process. The predominant searching tool for viewing is keyword spotting functions embedded in image viewing software, based on extra OCR programs. There are two common problems which make the patent viewing step difficult for users:

- Many patent documents have more than 50 pages (the longest patent document has as many as 3,000 pages), which leads to a very long OCR time. For example, we tried a patent with 61 pages on a normal PC with 2.33GH CUP and 3.25 GB RAM using the OCR plug-in provided by Acrobat 8, it will take 6 minutes to process it.

- The layout of patent images is not optimal for reading purpose. Firstly, all figures in a patent document appear together in the drawing section before the description section. Therefore, when a user is reading a description paragraph and wants to refer to the relevant figures, he has to scroll back to the drawing section. Secondly, in order to locate the exact sentences about a label, a user may have to go through a whole description paragraph, which will definitely slow down the reading speed.

Also, challenges of the poor image quality have to be overcome at the same time. Except for drawing pages, patent documents after 1970 are of good quality, and a sample is shown in Figure 5.4. However, those patents before 1970 may suffer from touching and broken characters, salt and pepper noise, as well as skew. A page from a patent document dated of 1911 is shown in Figure 5.5. In particular, skew appears



US 2007/0272624 A1 Nov. 29, 2007

2

as this means that the cover will detach completely from the skirt in one piece when removed. This is advantageous as it reduced the amount of handling required to prepare the device for use and consequently reduces the chances of contamination.

[**0017**] If an adhesive is used to make the bond between the peelable protective cover and the edge of the skirt, it is preferable that the adhesive attaches preferentially to the cover rather than the edge of the skirt. This allows all the adhesive to remain with the cover when the cover is removed and prevents any residue from being left on the edge of the skirt. Any residue left on the skirt would provide a potential contamination site and so it is advantageous to ensure that no residue is left on the skirt.

[**0018**] Preferably, the device of the present invention comprises a skirt wherein the ratio of the depth of the skirt to the width of the outlet is at least 0.1. Particularly preferably, the ratio of the depth of the skirt to the width of the outlet is at least 0.15. Especially preferably, the ratio of the depth of the skirt to the width of the outlet is at least 0.2. The width of the outlet will be its diameter when the outlet is circular. This ratio range has the advantage that it will provide a suitably sized physical barrier to prevent contact between a foreign object and a treated water contacting part of the device (see above). Preferably, the ratio of the depth of the skirt to the width of the outlet not more than 1. Particularly preferably, the ratio of the depth of the skirt to the width of the outlet is not more than 0.7. Especially preferably, the ratio of the depth of the skirt to the width of the outlet is not more than 0.5.

[**0019**] Preferably, the device of the present invention comprises a plate comprised of a plurality of openings that provide a dispersed spray when liquid passes through them.

[**0020**] Particularly preferably, the plurality of openings are such that the spray produced is divergent.

[**0021**] Preferably, the device of the present invention comprises a chamber which contains a filter medium.

[**0022**] In a further aspect the present invention provides a method for providing a treated liquid delivery system comprising the steps of; fitting a device as claimed in claim **1** to a liquid source via a suitable connector and removing the peelable protective cover from the device.

[**0023**] In a yet further aspect the present invention provides a treated liquid delivery system comprising; a pipe connected to a liquid source and a device as claimed in claim **1** connected to the pipe.

[**0024**] The peelable protective cover can comprise a plurality of layers of one or more materials, for example in a laminate. The cover can comprise one or more layers in the form of a polymeric film, a metal foil, paper and other cellulosic sheet materials (with or without polymeric or other coating), non-woven polymeric sheets. Examples of polymeric materials which can be used in the cover include polyolefins, especially polyethylene and polypropylene, polyesters, polyamides, polysulphones. Polymers can be treated to optimise their properties for use in the cover, for example by crosslinking to optimise their characteristics when exposed to heat. For example a polymer layer might be relied on to form a bond between the cover and the edge of the skirt, for example as a result of softening when exposed to heat. Other polymer layers might be treated so as to reduce their tendency to soften when exposed to heat, for example by crosslinking.

[**0025**] The protective cover can be reinforced to improve resistance to tearing, for example by means of a metal foil or by means of fibres. For example a reinforcing component can be included between two polymer film layers.

[**0026**] The use of a metal foil in the cover has the advantage that the cover can be heated by exposure to appropriate electromagnetic radiation, for example by inductive heating. This can be relied on to cause an increase in temperature such that a polymeric layer of the cover softens.

[**0027**] The connector supplied with the device for attaching the inlet of the device to a tap can be of a standard type eg. a connector supplied from Walther GmbH & Co, Coupleurs Gromelle or Colder Products Company or a suitable threaded connection. A bayonet connection can also be used.

[**0028**] The body of the device can be made from suitable materials including, but not limited to, polyamides, polyethylenes, polypropylenes, polyesters, polysulphones or and other polymers which are capable of being produced in Medical Grade. The device can be produced by a number of forming methods including injection moulding, blow moulding, rotational moulding, machining from solid or casting.

[**0029**] If a filter is present in the device, suitable materials for the filter depend on the level of organism clearance required i.e. the maximum size of organism that is considered as non-contaminating. The filter should be constructed to trap any organisms or particles over this size. A typical suitable filter pore size is $0.2\ \mathrm{H}\ 10^{-6}$ m i.e. 0.2 μm. Suitable materials with this pore size are polymeric membranes with suitable supporting materials. The materials are typically multi layered with each layer performing a different function such as pre-filtration, sterilising, and drainage.

[**0030**] Other possible filter media types could be metal based membranes, organic, inorganic or metallic fibre based, granular based, extruded porous structures or beads.

[**0031**] Materials used in the device of the invention can include additives to optimise characteristics for example for processing by moulding techniques, for sterilisation by exposure to high temperature and/or pressure. Additives can be included to optimise bactericidal and aesthetic (for example colour) characteristics.

BRIEF DESCRIPTION OF THE DRAWINGS

[**0032**] Embodiments of the invention will now be described, by way of example only, and with reference to the accompanying drawings, in which:

[**0033**] FIG. **1** is a cross section of an elevation view of a device according to the present invention;

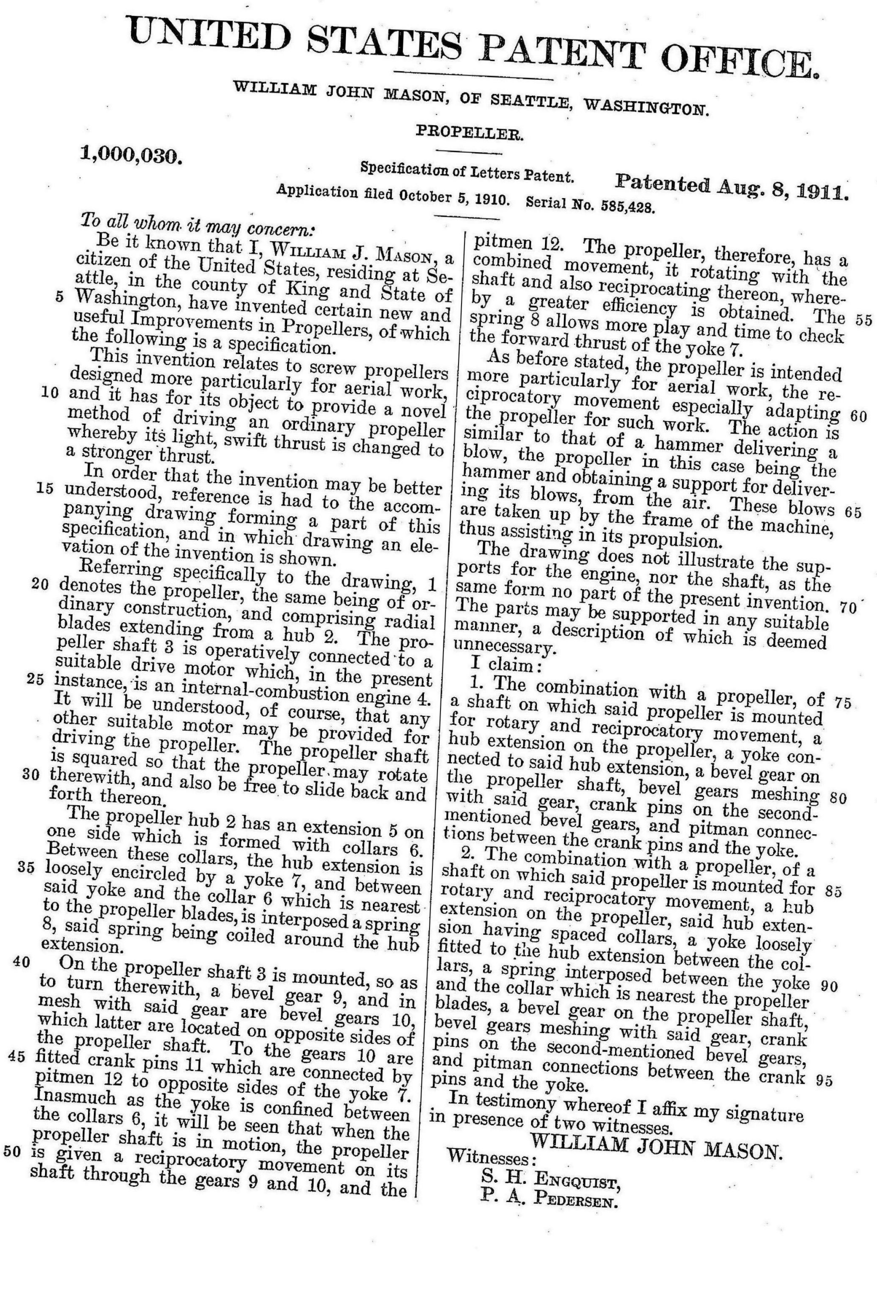
[**0034**] FIG. **2** is an isometric view of the outlet of the device shown in FIG. **1**; and

[**0035**] FIG. **3** is an isometric view of an outlet of another embodiment of a device according to the present invention.

DETAILED DESCRIPTION OF THE PRFERRED EMBODIMENT

[**0036**] Referring to the drawings, FIGS. **1** and **2** show a device (**10**) for treating a flowing liquid which comprises an

Figure 5.4: A patent image dated on 29 Nov. 2007 from the USPTO database.



UNITED STATES PATENT OFFICE.

WILLIAM JOHN MASON, OF SEATTLE, WASHINGTON.

PROPELLER.

1,000,030. Specification of Letters Patent. Patented Aug. 8, 1911.

Application filed October 5, 1910. Serial No. 585,428.

To all whom it may concern:

Be it known that I, William J. Mason, a citizen of the United States, residing at Seattle, in the county of King and State of Washington, have invented certain new and useful Improvements in Propellers, of which the following is a specification.

This invention relates to screw propellers designed more particularly for aerial work, and it has for its object to provide a novel method of driving an ordinary propeller whereby its light, swift thrust is changed to a stronger thrust.

In order that the invention may be better understood, reference is had to the accompanying drawing forming a part of this specification, and in which drawing an elevation of the invention is shown.

Referring specifically to the drawing, 1 denotes the propeller, the same being of ordinary construction, and comprising radial blades extending from a hub 2. The propeller shaft 3 is operatively connected to a suitable drive motor which, in the present instance, is an internal-combustion engine 4. It will be understood, of course, that any other suitable motor may be provided for driving the propeller. The propeller shaft is squared so that the propeller may rotate therewith, and also be free to slide back and forth thereon.

The propeller hub 2 has an extension 5 on one side which is formed with collars 6. Between these collars, the hub extension is loosely encircled by a yoke 7, and between said yoke and the collar 6 which is nearest to the propeller blades, is interposed a spring 8, said spring being coiled around the hub extension.

On the propeller shaft 3 is mounted, so as to turn therewith, a bevel gear 9, and in mesh with said gear are bevel gears 10, which latter are located on opposite sides of the propeller shaft. To the gears 10 are fitted crank pins 11 which are connected by pitmen 12 to opposite sides of the yoke 7. Inasmuch as the yoke is confined between the collars 6, it will be seen that when the propeller shaft is in motion, the propeller is given a reciprocatory movement on its shaft through the gears 9 and 10, and the pitmen 12. The propeller, therefore, has a combined movement, it rotating with the shaft and also reciprocating thereon, whereby a greater efficiency is obtained. The spring 8 allows more play and time to check the forward thrust of the yoke 7.

As before stated, the propeller is intended more particularly for aerial work, the reciprocatory movement especially adapting the propeller for such work. The action is similar to that of a hammer delivering a blow, the propeller in this case being the hammer and obtaining a support for delivering its blows, from the air. These blows are taken up by the frame of the machine, thus assisting in its propulsion.

The drawing does not illustrate the supports for the engine, nor the shaft, as the same form no part of the present invention. The parts may be supported in any suitable manner, a description of which is deemed unnecessary.

I claim:

1. The combination with a propeller, of a shaft on which said propeller is mounted for rotary and reciprocatory movement, a hub extension on the propeller, a yoke connected to said hub extension, a bevel gear on the propeller shaft, bevel gears meshing with said gear, crank pins on the second-mentioned bevel gears, and pitman connections between the crank pins and the yoke.

2. The combination with a propeller, of a shaft on which said propeller is mounted for rotary and reciprocatory movement, a hub extension on the propeller, said hub extension having spaced collars, a yoke loosely fitted to the hub extension between the collars, a spring interposed between the yoke and the collar which is nearest the propeller blades, a bevel gear on the propeller shaft, bevel gears meshing with said gear, crank pins on the second-mentioned bevel gears, and pitman connections between the crank pins and the yoke.

In testimony whereof I affix my signature in presence of two witnesses.

WILLIAM JOHN MASON.

Witnesses:
S. H. Engquist,
P. A. Pedersen.

Figure 5.5: A patent image dated on 8 Aug. 1911 from the USPTO database.

frequently in almost every pages of patents before 1970, and drawing pages of patents after 1970.

Retrieval of relevant information from degraded images poses a great challenge, because direct application of OCR on these images returns very poor results. In this chapter, a system which helps the user to view patent documents will be introduced. The overview of this system is shown in Figure 5.6. Solutions to address these problems mentioned above are provided:

- In order to facilitate faster keyword spotting in patent images, we propose a holistic word spotting method called Radial Projection Profile, which is fast, robust to touching and broken characters, as well as invariant to skew. In Section 5.2, I will introduce the method and a keyword spotting system based on it.
- In order to make textual content in the drawings searchable to users, we propose a method to locate text content out of patent drawings, which will be introduced in Section 5.3. The located content is then recognized by OCR, and linked to the corresponding text description.

5.2 A Holistic Word Spotting Method for Skewed Document Images

One strategy of handling skew is to remove it by skew detection methods. A review of this can be found in Section 2.3.1 of Chapter 2. However, skew detection itself is very time consuming. According to [CWL03], the average skew detection time per page is

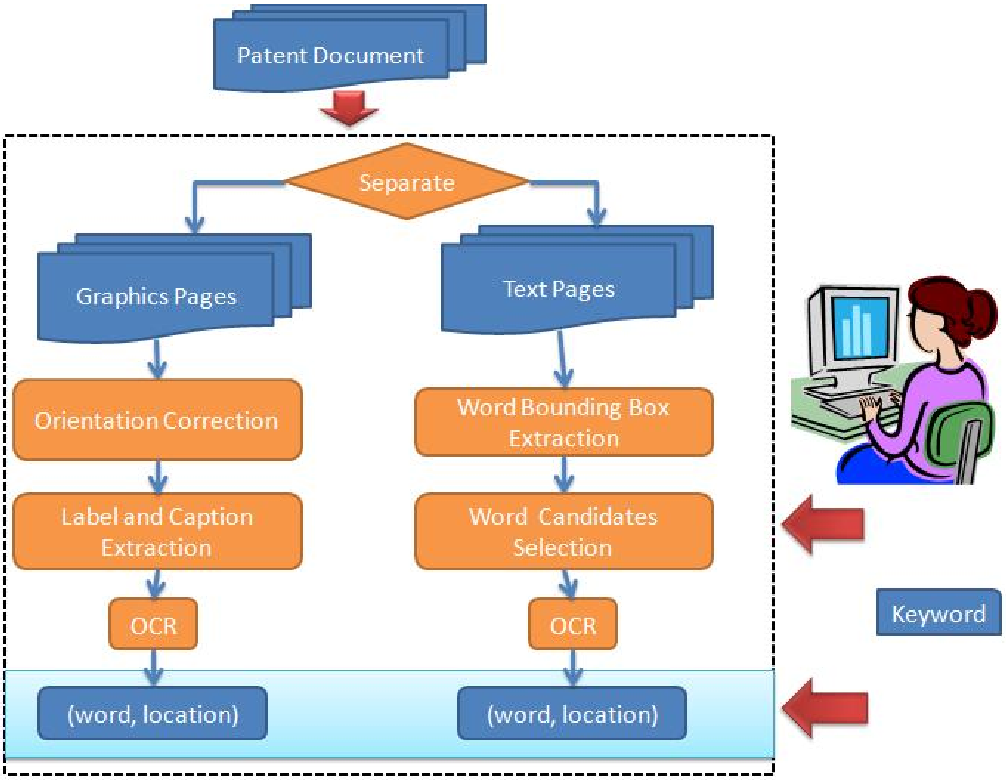


Figure 5.6: A system to help the user to browse a patent document.

at least one second per page, almost comparable to OCR. Therefore, it is better to avoid this step.

Both Word Shape Coding [LLT08, LT04, Spi94], and Holistic Word Spotting [RM03, MMS06, KJM07, HHS92] techniques are employed in the word spotting application. A review can be found in Section 2.1.3 of Chapter 2. I will introduce in this section a new holistic word spotting method, named Radial Projection Profile(RPP), to locate keyword in patent documents. There should be a balance between speed and robustness. Our holistic word spotting method has the following characteristics:

- Robust to skew and other noise.

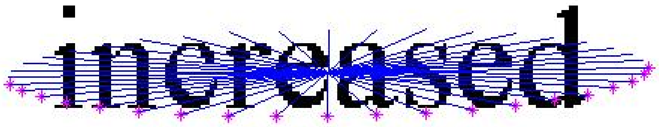


Figure 5.7: Radial projection profile.

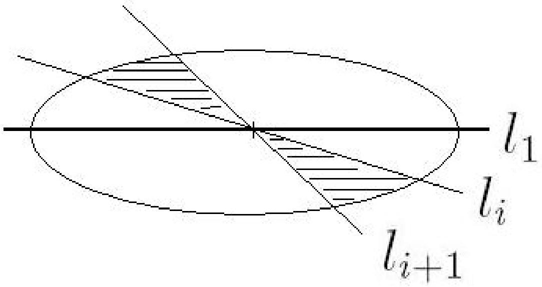


Figure 5.8: The way to sample points.

- Fast computation speed.

5.2.1 Radial Projection Profile

The goal of our method is to locate the word effectively in a degraded document image. Projection profile is a very important approach in pattern recognition area. However, when skew appears, this method fails. Therefore, we propose a radial projection method, which projects from a single point in contrast to a specific direction as usual projection methods do.

Assume there are a word W and k lines $\{l_i, i = 1 : k\}$ across the mass centroid, denoted by o, of W. An example of word "increase" and several lines are shown in Figure 5.7. l_1 is the line coincides with the major axis of the ellipse. k lines are sampled as follows. The ellipse which has the same normalized second central moment as the word region is found, and the sample points are found on the ellipse. l_1 is the line which coincides with the major axis of the ellipse. Suppose the total area of two sectors defined by l_i and l_{i+1} is $area(i, i+1)$, such as the shaded area labeled shadow in Figure 5.8. Lines are sampled in a way that all $area(i, i+1)$ have the same value. Then, the number of transitions on each line is collected by a histogram $hist$, and the ith bin of the histogram, namely $hist(i)$, records the number of transitions on line l_i.

We sample lines in this way instead of sampling it on an equal-distance manner as we employed in Chapter 4 (signature invariant to affine transformation), because transitions are denser near l_1. The benefit of this sampling method is to capture the variation of transition numbers as much as possible. Since the ellipse just evaluates the orientation of the word approximately, it is not accurate to align $hist1_1$ with $hist2_1$, because the major axis of the ellipse may change due to noise. To address this problem, the comparison of two histograms $hist1$ and $hist2$ is formulated as:

$$Score(hist1, hist2) = \underset{t}{\operatorname{argmax}} \sum_{j=(i+t)} (hist1(i) - hist2(j)) \qquad (5.2.1)$$

$$t = [1 : \alpha] \bigcup [k - \alpha : k] \qquad (5.2.2)$$

where α is a threshold to control the degree of variation. As a matter of fact, this is to search the smallest Manhattan distance between $hist1$ and $hist2$, by circularly shifting the values in $hist2$ by t elements. This idea is inspired by the fact that: when people trying to pick up keywords in a text, the word length is an important feature for us to detect the word. Table 5.1[1] is a breakdown of 3058 frequently-used English words by their length. It is easy to see that at least 80% words of a document will be eliminated if word length is employed as filtering criterion. Although word

[1] The data is from http://www.usingenglish.com/profiles/tdol/archives/000085.html.

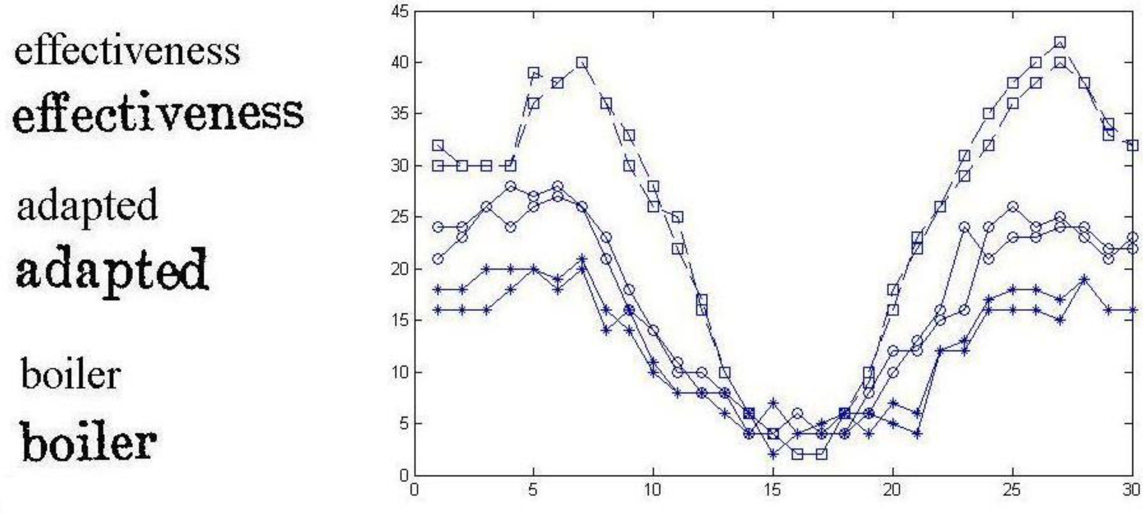


Figure 5.9: The radial projection profiles of three pairs of words.

length method is efficient and straightforward, it becomes difficult when there are broken and touching characters. The main purpose of our radial projection profile is to capture the length of a word in a way that it be hardly affected by character size, condition of touching and broken characters, or even font. Figure 5.9 shows the histograms of four pairs of words. The histograms of different words are labeled with the different makers, namely star (boiler), circle(adapted), and square (effectiveness). As shown in this figure, shorter words have flatter histograms, while longer words have histograms with higher peaks.

5.2.2 Experiment Results

Four other methods are employed for comparison in our experiment, namely, LU's, LV's, Ho's [HHS92], and Marinai's [MMS06]. Since Marinai's system is very complex, we didn't implement the system. The results shown in Tables 5.2 and 5.3 is summarized from the result of [MMS06] with the best recall and precision. We implement

Table 5.1: The breakdown of 3058 frequently-used English words by length.

1 letter words	93	3.0%	2 letter words	474	15.5%
3 letter words	612	20.0%	4 letter words	510	16.7%
5 letter words	397	13.0%	6 letter words	266	8.7%
7 letter words	236	7.7%	8 letter words	193	6.3%
9 letter words	115	3.8%	10 letter words	72	2.4%
11 letter words	37	1.2%	12 letter words	22	0.7%
13 letter words	22	0.7%	14 letter words	5	0.2%
15 letter words	3	0.1%			

LU's, LV's and Ho's methods with C++. The line finding, baseline fitting and word segmentation functions is provided by Tersseract.

Two testing datasets were prepared: Dataset I comprises of 20 pages in Times New Roman font from the synthetic image set of UWI, which have very good quality; Dataset II comprises of 20 pages in Times New Roman font from the real image set of UWI, which suffer from diverse degradation such as: dark or light printing, touching characters, broken characters, warping at the edge of page, and slight skew.

We randomly selected 100 words from a word list which have a length of 4 to 10 characters. They have 532 occurrences in both datasets in total. The parameters are set as $k = 60$, $\alpha = 5$ corresponding to a $\pm 5^\circ$ searching range.

From Tables 5.2 and 5.3, it is found that both ours and Ho's method are robust to image degradation, the recall and precision change slightly when the image quality become bad. However, Ho's method is more than 4 times slower than our method. LV's and LU's methods work fast and accurate when the document image is clean, but the performance drops quickly when noise appears. In particular, LV's method requires a deskew process in his original paper, but it is not implemented in our experiment. Marinai's system have the best performance in terms of recall and precision.

Table 5.2: Word spotting results (Set I).

	Recall	Precision	Processing Time (100 pages)	Retrieval Time (100 pages)
Ours	100%	57.86%	48.76	0.03
Ho's	96.67%	84.23%	210.21	0.05
LU's	92.48%	90.53%	42.01	0.15
LV's	97.33%	84.64%	46.63	0.41
Marinai's	100%	92%	Training needed Indexing needed	9.6

Table 5.3: Word spotting results (Set II).

	Recall	Precision	Processing Time (100 pages)	Retrieval Time (100 pages)
Ours	97.43%	52.54%	50.06	0.03
Ho	93.54%	76.43%	213.87	0.05
LU	65.48%	72.53%	45.67	0.15
LV	67.33%	60.64%	48.79	0.41
Marinai	100%	87%	Training needed Indexing needed	9.6

However, this method needs extra training and indexing; also the comparison is very slow. When document image is degraded, the processing of all methods slow down, mainly due to the line finding and baseline fitting processing.

From Tables 5.2 and 5.3, it is also found that the recall of the proposed method is higher than 95% even when the document images suffer from severe degradation. However, the precision is not satisfactory, because the radial projection method only capture the length of the word, and many words will share the same length.

In our experiment, several features are tested, including maximum length of white strokes, total number of black pixels, maximum length of black strokes, and the

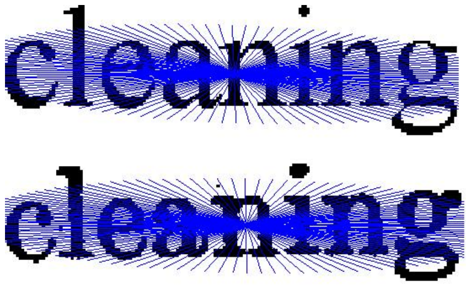
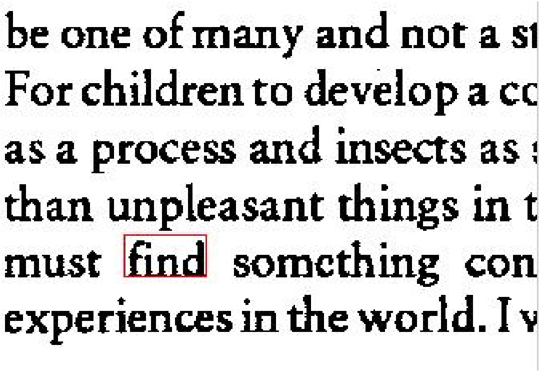


Figure 5.10: When the centroid moves, our method still works.



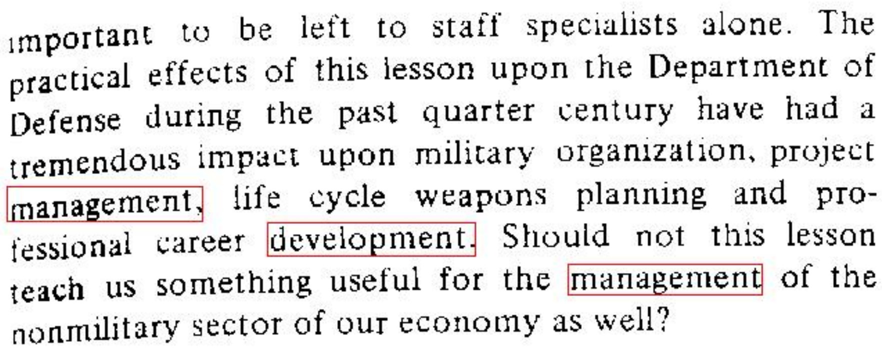
be one of many and not a st
For children to develop a cc
as a process and insects as
than unpleasant things in t
must find something con
experiences in the world. I v

Figure 5.11: An example where OCR fails and our method still detects.

number of transitions. The number of transitions is the most reliable feature, which is robust to font, size, dark and light print, as well as the location of centroid. The radial projection profile remains almost the same when the centroid is moving in a small range as shown in Figure 5.10.

Figure 5.14 shows that this method is tolerant to skew, touching and broken characters, and slight variation of fonts. Figure 5.11 shows some characters which cannot be correctly recognized by OCR but still can be retrieved by our method. Figure 5.12 shows an example of retrieving a word “management” on a warping surface, which frequently happens when a thick-bound book is scanned.

The number of transitions is employed as the feature of the projection in our



important to be left to staff specialists alone. The practical effects of this lesson upon the Department of Defense during the past quarter century have had a tremendous impact upon military organization, project management, life cycle weapons planning and professional career development. Should not this lesson teach us something useful for the management of the nonmilitary sector of our economy as well?

Figure 5.12: Spotting words on a warping surface.

method. In fact, several other possible features are also tried, including maximum length of white strokes, total number of black pixels, maximum length of black strokes, and the number of transitions. We found that the number of transitions is the most reliable feature, which is robust to many variations such as character size and bold style. In particular, the centroid of different word instances of the same identity often vary within a certain range due to noise. However, the radial projection profiles remain almost invariant under such variation. An example is shown in Figure 5.10. Figure 5.10(a) shows a query with the centroid and sample lines, and Figure 5.10(b) is a word retrieved by our method. The location of centroid are slightly different but our method is still able to find the word correctly.

5.2.3 Fast Keyword Spotting in Imaged Patent Documents

In this section, I will detail a keyword spotting system based on RPP. As shown in Figure 5.13, when a query comes in, word blobs are extracted by the XY-cut algorithm. Word blobs in the document which have similar ratio to the query are selected as the initial candidates. The ratio of a word is defined as the eccentricity

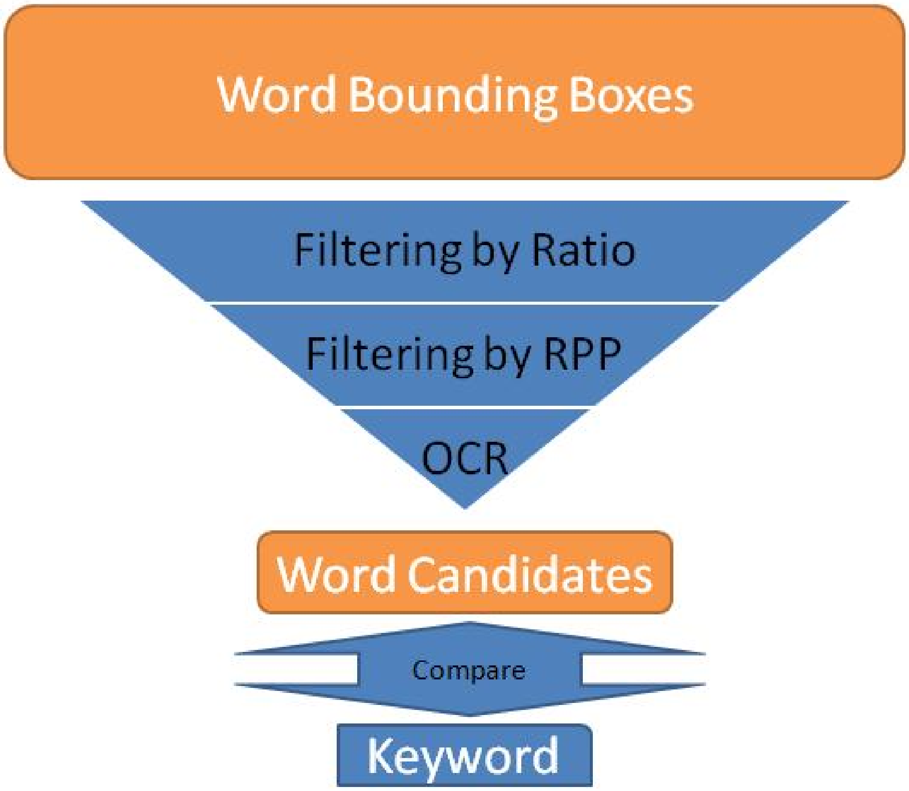
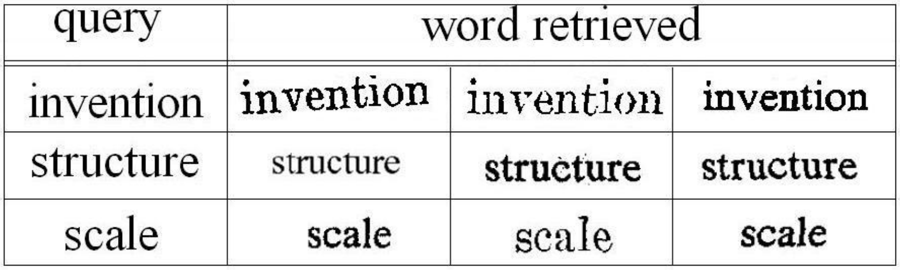


Figure 5.13: The workflow of the real time word spotting system.

of the ellipse that has the same second-moments as the word region. If the ratio of a word blob is within 0.5 to 1.5 times of the ratio of the query, it is selected as an initial candidate. The range is set wide, because we want to avoid false negative in this step. Then, the radial projection profiles of these candidates are compared with the that of the query, and candidates within a certain distance ($dist = 0.2$) from the query are remained otherwise filtered out. In our experiment, parameters of RPP are set as $k = 90$, $\alpha = 90$. It means that the maximum skew angle can be detected is $\pm 180°$, the minimum skew angle can be detected is $1°$. $1°$ is accepted for most OCR software. In the final step, an OCR is applied to the remaining candidates. The skew angle t detected by our method is used as an input of OCR program. The query is



query	word retrieved		
invention	invention	invention	invention
structure	structure	structure	structure
scale	scale	scale	scale

Figure 5.14: Some words retrieved by our method.

Table 5.4: Word spotting results in three 50-pages patent documents.

	Recall	Precision	Time spent in the first retrieval (s)	Time spent in other retrievals (s)
Tesseract	89.54%	84.15%	697.31	0.04
Our system	88.45%	83.53%	13.05	3.01

only compared with the output of OCR. The OCR used in our system is Tesseract V2.03(with deskewing function from OCRopus, but no other preprocessing).

Three patent documents with at least 50 text pages (drawing pages are removed) are selected as testing data. We only used the first 50 pages of these documents, because this makes it easier to compare the results. 100 keywords are searched in each document respectively.

The average precision, recall, and speed for our system and the Tesseract system alone are shown in Table 5.4. The speed measure is divided into two parts, namely the first retrieval and others. During the first retrieval, processing steps like layout analysis, word boundary extraction, and recognition are conducted. It also includes the time for comparing query against the stored data. The time for other retrievals only contains the time of comparison. Figure 5.14 shows some word examples retrieved by our method in the patent dataset.

The experiment shows that our system was more than 50 times faster than the original OCR program, and yet has comparable accuracy. However, it takes much longer than OCR system to process queries other than the first one, because we still need OCR to in the “other retrieval” process. However, with the help of better OCR program, we may be able to make this time shorter. The precision and recall of our system is not very impressive. Because the recognition accuracy of Tesseract employed in our system is not optimal for patent collections. In a nutshell, one drawback of our system is that the speed and accuracy is bounded by the OCR program employed. Although it took 7.56 seconds to process a document using radial projection profile alone, compared to 697.31 seconds to use OCR alone, we have to take relatively longer time to recognize these word candidates identified by the radial projection profile method.

5.3 Textual Information Extraction from Graphics

Text content in a drawing plays an essential role in helping users to understand the drawing. In this section, a system to extract and recognize captions and labels in drawings is introduced. To make the extracted captions and labels searchable, they are recognized and linked with the corresponding descriptions by HTML functions in our patent viewing system. Users thereafter are able to efficiently jump to the relevant description by clicking the captions or labels, or vice versa.

5.3.1 System Description

The workflow of the system is illustrated in Figure 5.15. Firstly, drawing pages (the drawing section) are separated from text pages (the other sections). The next step is to rectify rotated pages. A rotated page, such as Figure 5.2(b), is a page whose caption and labels are vertically positioned (that of typical pages are horizontally posed). Thirdly, bounding boxes of the captions and labels are located in figures. Then, the content in the target bounding boxes are recognized by an OCR software. Subsequently, a post processing step is employed to filter out errors and words that are out of our interests. Finally, captions and labels are linked to the description by HTML functions. Users are able to swiftly search a caption or label in the description by clicking the caption or label in the figure. A browsing interface of the system is shown in Figure 5.23.

5.3.2 Drawing/Text Page Separation

The first task is to separate graphic pages from text pages. The heuristic is that, in a text page, black pixels spread all over the page uniformly, while in a drawing page, the distribution of black pixels is very uneven. A page is divided into N $k \times k$ blocks. The black pixel density of block i, is obtained by:

$$s(i) = \frac{1}{k \times k} \sum_{k \times k} g(t) \tag{5.3.1}$$

where t is the pixel index of block i; $g(t) = 1$ if t is a black pixel, otherwise $g(t) = 0$. The black pixel density of the whole page is defined as:

$$mean = \frac{1}{N} \sum_{N} s(i) \tag{5.3.2}$$

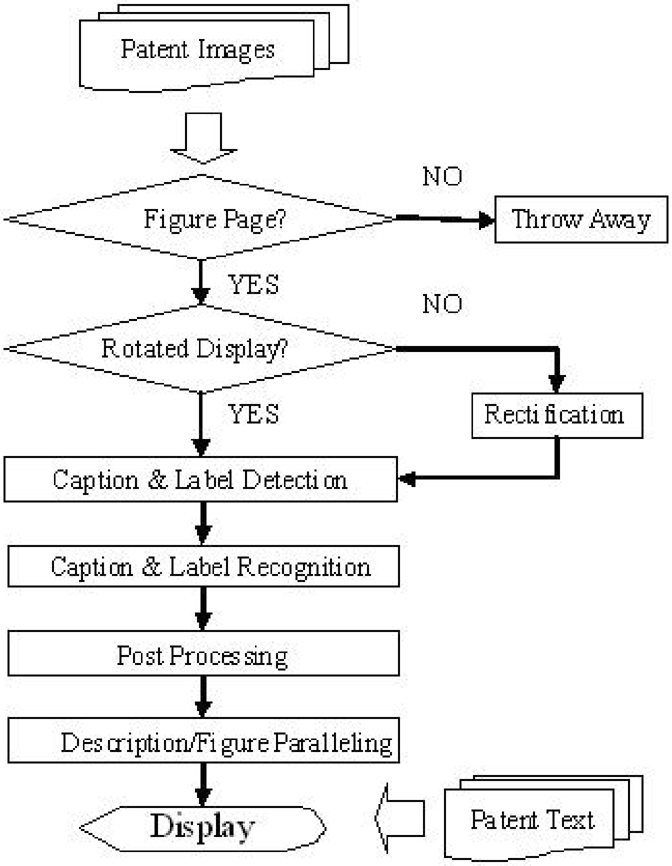


Figure 5.15: The workflow of the drawing image processing system.

The black pixel standard deviation of the whole page is defined as:

$$dev = \sqrt{\frac{1}{N}\sum_{N}(s(i) - mean)^2} \tag{5.3.3}$$

Because the layout of patent documents are quite homogeneous in the database, *mean* and *dev* are very consistent for text pages (except the last page). Therefore, a simple threshold is employed to distinguish drawing pages from text pages. In the experiments, if $mean \in [0.04, 0.06]$ and $dev \in [0.085, 0.1]$, the page is a text page, otherwise, drawing page.

5.3.3 Landscape Page Rectification

In a landscape drawing page, the orientation of the head is perpendicular to that of the content text. The landscape page detection is based on the text components (characters) obtained by a preliminary classification method. After all connected components are extracted, each component is then classified as either a text component or a drawing component according to the width, height, width/height ratio (denoted by “ratio” in Table 5.5), area, and black pixel density. The decision rule is shown in Table 5.5. If a parameter of a component is within the suggested range, it is classified as a text component, otherwise a drawing component.

This rectification method is similar to Document-Spectrum analysis [O'G93]. The nearest neighbor character of a character is found, and the central line defined by the centroids of both characters is computed. Angles of central lines are collected. The dominant angle decides whether the page is rotated or not.

5.3.4 Caption/Label Detection

Many works have been reported to address the drawing/text separation problem. The drawing/text separation approaches can be divided into run-length analysis [LTW95]

Table 5.5: Preliminary component classification criteria.

	Width	Height	Ratio	Area	BPD
Upper bound	100	100	10	1500	0.9
Lower bound	5	5	1	900	0.2

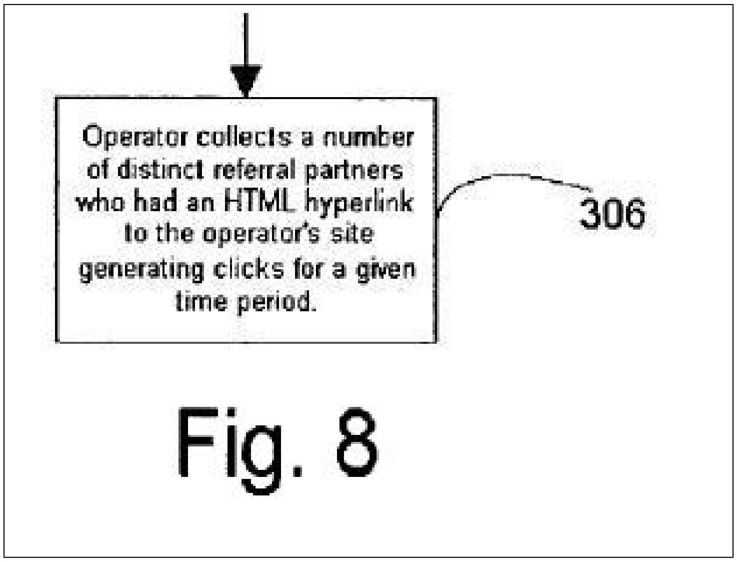

Operator collects a number of distinct referral partners who had an HTML hyperlink to the operator's site generating clicks for a given time period.
306
Fig. 8

Figure 5.16: A figure of a flow chart, where the caption, labels and explanations are of different character sizes.

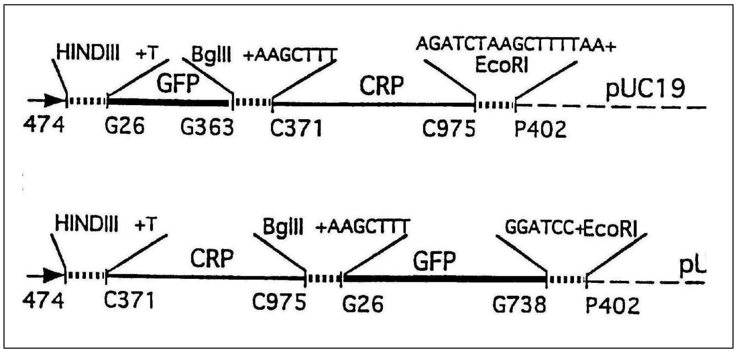


Figure 5.17: DNA sequences in a figure.

and connected component analysis [GTLT95, TTP+02, FK88, HA96]. The former has been widely used in processing text-rich document, while the latter is more frequently employed in processing drawing-rich document. As for component-based approaches, Gao [GTLT95] made use of a histogram of component size to detect the possible size of text components; Fletcher [FK88] employed a histogram of component area to find an appropriate area threshold in order to identify text components; Tombre [TTP+02] proposed a similar but further improved method to detect text components which are not horizontally or vertically posed. He [HA96] proposed a clustering-based approach making use of the radii of components[2]. All these approaches mentioned above assumed that the text components are of uniform or similar size, and text components are the majority of components. However, it is not true in patent figures. After manually checking with hundreds of patent images, we found that:

- Text in a patent image may include heads, labels, captions, and others (explanations in flow charts, DNA sequence, legends, and etc). We are only interested in labels and captions.
- Text in different patents are of sizes. Generally, the height of labels consistently ranges from 30 to 50 pixels, while the height of captions ranges from 50 to 400 pixels.
- Text in different patents are printed of diverse fonts, while some are handwritten, especially in patents before 1970.
- Different types of text may be of quite different size and fonts in the same document. The labels may not be the majority of the text. An example is

[2] The radius of a component is defined as the maximum Euclidean distance between the mass centroid of the component and a pixel on the out contour of the component.

shown in Figure 5.16.

- Many patent images, especially old ones, suffer from noise such as skew, salt and pepper noise, and touching/broken characters. In particular, touching/broken characters are quite common, making the size of components an unstable indicator of their identity. An example of severe touching characters is shown in Figure 5.17, where several 'T' are touching each others.

Because of these reasons, previous methods may not work on the patent archive, and hence we propose a new unsupervised-clustering-based method to detect text components (characters) in complex drawing in a general case. One merit of unsupervised clustering is that it avoids the training process, which is impractical for the current task due to the huge number of patents to be processed. The detection method includes two steps: detecting characters and grouping characters into words. This character detection method is based on three observations:

- Height: Labels and captions are of the same heights and fonts in the same document.

- Pattern: comprising of ten digits and tens of English characters, the pattern of each character repeats several times in a patent.

- Neighborhood: the nearest neighbor of a character is often a character.

Each component c_i is given a score $s(c_i)$, which indicates the likelihood of being a character, based on the three pieces of information. The detection method is as follows:

1: Noise Filtering. Components with an area smaller than 50 pixels or a solidity[3] greater than 0.8 are filtered out. This step will remove spots and short lines in the image, which occur very frequently. The remaining components are denoted by γ. Character '1' may be removed too, but they will be recovered in later steps.

2: Find the initial candidate set using pattern information. Components in γ are resized to 30×30 pixels. A hierarchical clustering algorithm [DH73], with Hamming distance as the distance measure, is applied to γ. If two components have less than T_1 pixels disagreement, they are classified into the same cluster. If a cluster has more than t members (components), its members are initial character candidates, otherwise they are removed from γ. t is set as the number of drawing pages in a document, in order to guarantee that the characters of captions can be found in this step. Because in an extreme case, one drawing page contains only one figure. The results of this step are shown in Figure 5.18(a).

3: Clustering initial candidates by their heights. Similar to step 2, components in γ are clustered by their heights into $K2$. If the height difference between two components is smaller than T_2, they are classified into the same cluster. The centroid of each cluster $k2_j$, denoted by $k2_j.centroid$, is recorded.

4: Expand γ using height information. Check all components in the image, if the height difference between a component and a centroid $k2_j.centroid$ is smaller than T_2, the components is classified into cluster $k2_j$. The score of the components is assigned as the number of initial candidates in this cluster. The results of step 3 and 4 are illustrated in Figure 5.18(b).

[3] Solidity is a scalar specifying the proportion of the pixels in the convex hull that are also in the region.

5: Filter out undesirable candidates using neighborhood information. There is a link between component $c_i \in \gamma$ and component $c_j \in \gamma$, if c_j is one of the n nearest neighbors of c_i over all components. Based on a one-degree propagation algorithm, scores propagate among components through links, as follows:

$$p(c_i, c_j) = score(c_i) + score(c_j) * \omega(c_i, c_j) \tag{5.3.4}$$

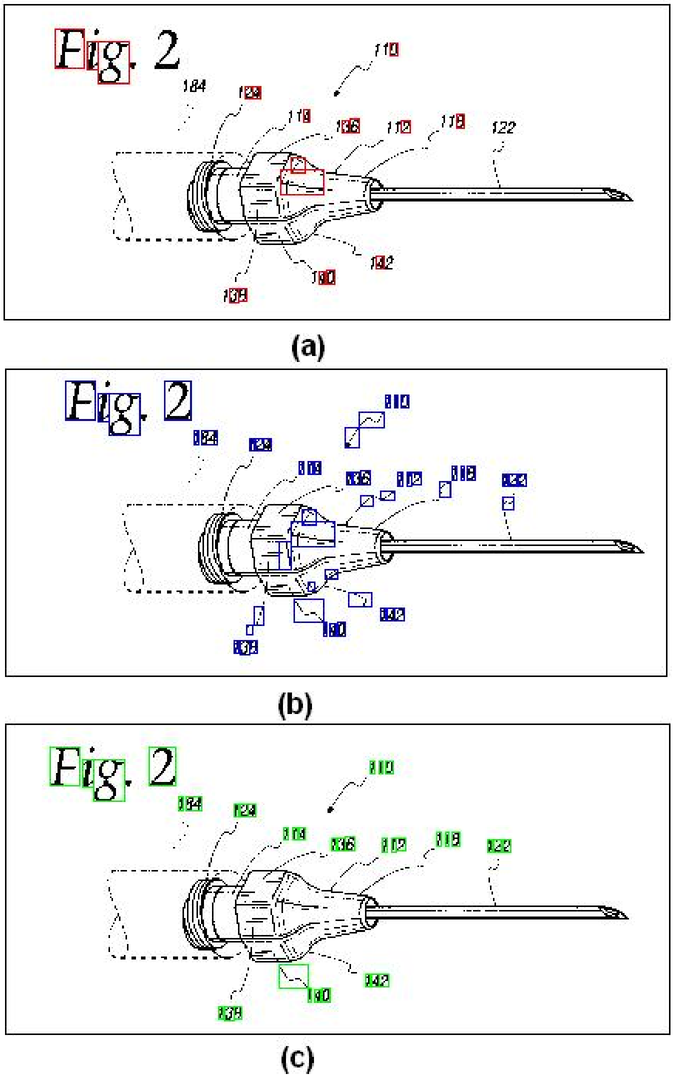
$$\omega(c_i, c_j) = \frac{min(c_i.bottom - c_j.top, j.bottom - i.top, 0)}{i.height + j.height} \tag{5.3.5}$$

where $min(i.bottom - j.top, j.bottom - i.top, 0)$ is the overlapping part of both components in the vertical direction.

The purpose of step 5 is twofold. Firstly, characters of captions repeat less frequently than those of labels. However, captions are of longer strings, thus the propagation will give characters of captions higher scores. Secondly, noise, surrounded by an unpredictable complex graphical context, is not likely to receive such extra credit. The results of this step are shown in Figure 5.18(c).

6: Components with scores greater than T_3 are considered as characters.

Three thresholds, namely T_1, T_2, and T_3, can be adjusted to achieve better performance in this algorithm. T_1 defines the maximum radius of a cluster in the pattern-based clustering. A larger T_1 value allows components of more different shape to be categorized into the same cluster. T_2 defines the maximum radius of a cluster in the height-based clustering. A larger T_2 value allows components in a cluster to have a wider variance in term of heights. Text of handwritten or text suffering from severe noise, which leads to large difference in appearance among characters of the same identity, should have larger T_1 and T_2. T_3 is a threshold to adjust the compromise between precision and recall. A larger T_3 value brings higher precision, otherwise



(a)

(b)

(c)

Figure 5.18: Caption/label detection results in a figure.

higher recall. $T3$ is set as the number of graphic pages for each patent, in order to guarantee that the captions can always be found. The pseudo-code of the algorithm is shown as follows.

```
**for all** $\{c_i | c_i \in C,\ c_i.solidity > 0.8,\ c_i.area < 50\}$
  $\gamma \leftarrow c_i$
**end**

$K = cluster(\gamma,\ pattern,\ T_1)$
**for all** $k_i \in K$
  **if** $k_i.size < t$
    remove members of $k_i$ from $\gamma$
    remove $k_i$ from $K$
  **end**
**end**

$K2 = cluster(\gamma,\ height,\ T_2)$
**for all** $c_i \in C$
  **for all** $k2_j \in K2$
    **if** $c_i.height \in [k2_j.centroid - t2,\ k2_j.centroid + T_2]$
      $k2_j \leftarrow c_i$, $\gamma \leftarrow c_i$
      $score(c_i) \leftarrow$ the original $k2_j.size$
    **end**
  **end**
**end**

**for all** $c_i \in \gamma$
  **for all** $c_j$ of $n$ nearest neighbors of $c_i$
    **if** $c_j \in \gamma$
      $score(c_i) \leftarrow score(c_i) + p(c_i, c_j)$
    **end**
  **end**
**end**

**for all** $c_i \in \gamma$
  **if** $score(c_i) < T3$
    remove $c_i$ from $\gamma$
  **end**
**end**
```

Since we are only interested in captions and labels, which are always in the same text line, the grouping task is quite simple. Text components are grouped into words or phrases by the grouping function:

$$f(c_i, c_j) = \sqrt{\frac{k c_i c_j}{c_i + c_j}} \tag{5.3.6}$$

where c_i and c_j are the areas of two components; the coefficient k is a constant value, which can be adjusted by the batch of samples in use ($k = 20$ in our experiment). c_i and c_j are considered as in the same group (a word), if they satisfy both conditions: $Euclidian\_distance(c_i, c_j) < f(c_i, c_j)$ and $\omega(c_i, c_j) > 0.5$.

5.3.5 Post processing

Bounding box detection step may generate false boxes which have no text information. Also, a figure may contain text description other than labels or captions. In addition, OCR process introduces recognition errors too. Therefore many undesirable contents appear in the OCR output. This step is used to pick up valid captions and labels from the OCR output. A valid label is a string comprising of n consecutive digits ($n < 3$). A valid caption may have two different patterns: "Fig *" and "Figure *" regardless of uppercase or lowercase. For example, both "FIG.1" and "figure.1_(a)" are valid captions. In this step, OCR output which do not comply with these three pattern are filtered out.

5.3.6 Experimental Results and Discussion

The system was tested on two patent sets. These patents were deliberately chosen from patent archives of different years. Set I comprises of 60 pages of patents (24

Table 5.6: Experimental results on Set I.

	Detection Accuracy	Recognition Accuracy
Formula	$\frac{correctly\ detected\ items}{ground-truth\ items}$	$\frac{correctly\ recognised\ items}{ground-truth\ items}$
Labels	96.12%	96.12%
Captions	95.21%	93.10%

Table 5.7: Experimental results on Set II.

	Detection Accuracy	Recognition Accuracy
Formula	$\frac{correctly\ detected\ items}{ground-truth\ items}$	$\frac{correctly\ recognised\ items}{ground-truth\ items}$
Labels	92.12%	85.12%
Captions	83.08%	62.30%

documents) after 1975. These patent images are usually well printed. Set II comprises of 40 pages (27 documents) from 1867 to 1975. Comparing with Set I, images of Set II suffer from image quality degradations such as salt and pepper noise, skew, and shadow. Besides, most text of these patents are handwritten, and thus characters of the same identity may have very different appearances.

In the experiment, in the drawing/text page separation step and the rotated page rectification step, we achieved 100% accuracy. The results of the caption/label detection on Sets I and II are shown in Table 5.6 with $T_1 = 20$, and $T_2 = 4$, and Table 5.7 with $T_1 = 40$, and $T_2 = 8$ respectively. The guideline of tuning parameter is to make the recall as high as possible, while to keep the precision to a reasonable level, because recognition results of drawing components almost are invalid strings.

Table 5.6 shows that the detection method located 96.12% of labels and 95.21% of captions in testing data set I. The main reason which caused errors in label detection was that a few labels were touching the drawing, such as label 69 in Figure 5.19.

Those labels cannot be detected and are then missed in the final output. Table 5.6 also shows that all detected labels are correctly recognized by OCR, but several detected captions were wrongly recognized. These captions, such as those shown in Figure 5.20, are handwritten, which cannot be recognized by OCR. It should be noted that, although the detection method may falsely classify some drawing components as text components, the recognition results of these components were invalid strings, which would be filtered out in the post-processing step.

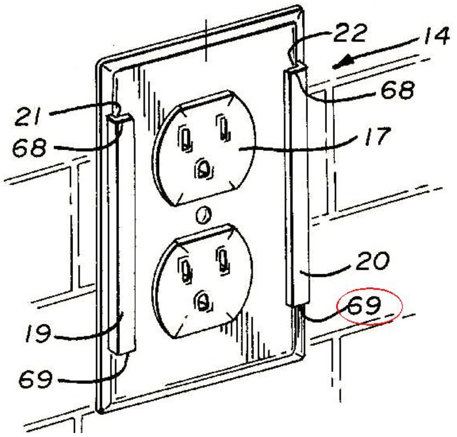


Figure 5.19: Because label 69 is connected to the drawing, it is classified as graphic component.

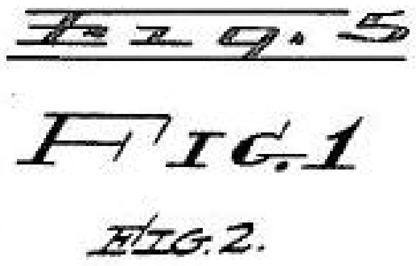


Figure 5.20: Handwritten captions in Set I.

Table 5.7 shows that the detection method located 92.12% of labels and 83.08% of captions in testing data set II. However, about 7% of detected labels and 20% of

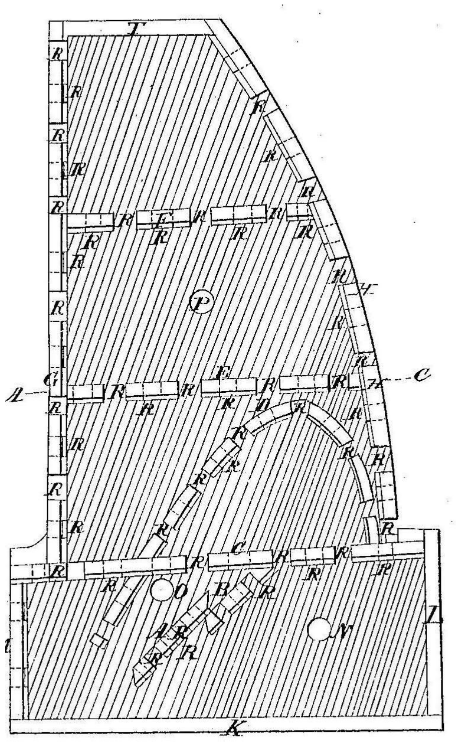


Figure 5.21: Labels appear on top of the drawing, making it difficult to detect them.

detected captions could not be recognized. The reasons for detection errors of Set II are more diverse. In early patents, labels may appear on top of the drawing such as shown in Figure 5.21, making the detection extremely difficult. Furthermore, handwritten labels and captions of varying appearances cause the pattern-based clustering step fail badly, as the template matching measure employed in this step is sensitive to handwritten variation. The component bounding box may change largely due to the stretching of one stroke. For example, in Figure 5.22, captions in the same document cannot be grouped into the same cluster due to handwritten variation. In this case, setting a higher T_1 will help. The reason for recognition errors is similar to that of Set I.

Besides labels and captions, there is some other text content in figures. In this

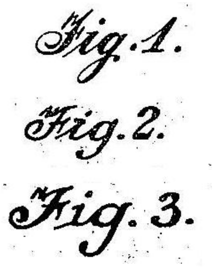


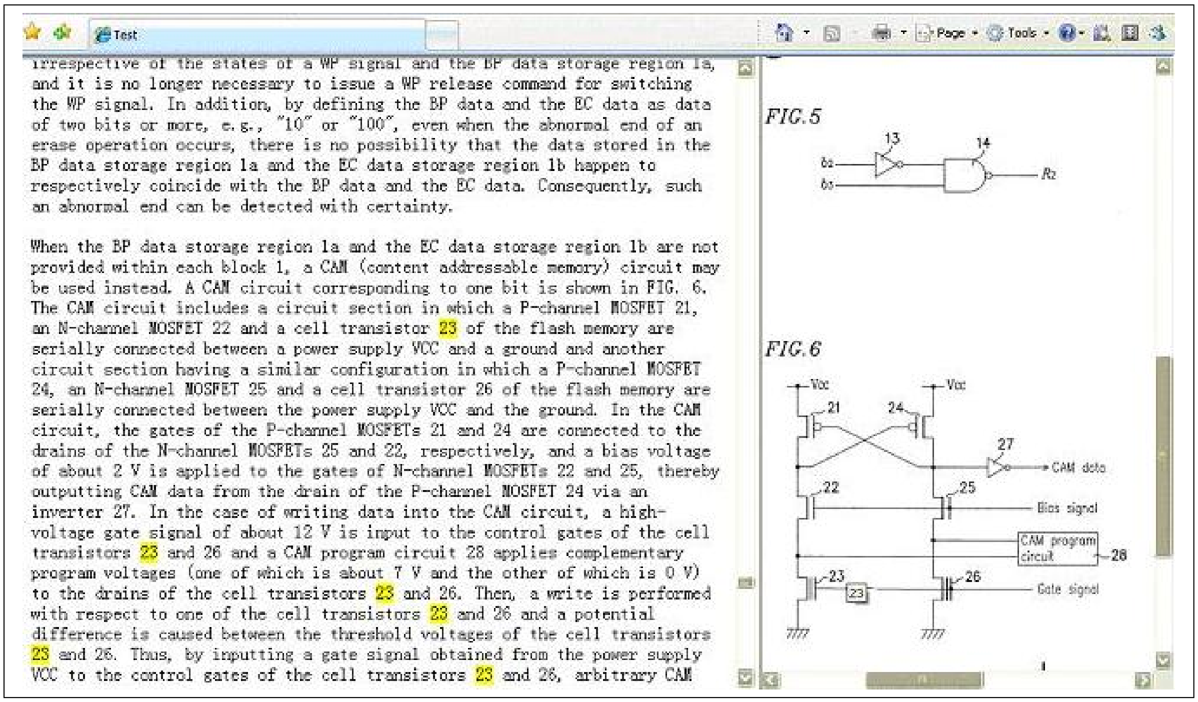
Fig.1.
Fig.2.
Fig.3.

Figure 5.22: Handwritten captions in Set II have very different appearances, and cause the pattern-based clustering to fail.

experiment, we did not calculate the detection and recognition accuracy for them. However, after manually checking the output, we knew that the recall of detection was more than 90%, even for those DNA sequences shown in Figure 5.17 which have severe amount of consecutive characters touching.

5.3.7 User Interface Demo

In order to show the extraction result, the system provides a spotting function for captions and labels across the description, if a user clicks the corresponding areas in the figure. A snapshot of the preliminary user interface of our system is shown in Figure 5.23. The left part of the interface is a window displaying the text version of the patent, and the right part is a window display the drawing pages of the patent. Drawing pages are shown in order. When a label is clicked, the corresponding label occurrences in the text window are located and highlighted. Figure 5.23 shows when label 23 was clicked.



irrespective of the states of a WP signal and the BP data storage region 1a, and it is no longer necessary to issue a WP release command for switching the WP signal. In addition, by defining the BP data and the EC data as data of two bits or more, e.g., "10" or "100", even when the abnormal end of an erase operation occurs, there is no possibility that the data stored in the BP data storage region 1a and the EC data storage region 1b happen to respectively coincide with the BP data and the EC data. Consequently, such an abnormal end can be detected with certainty.

When the BP data storage region 1a and the EC data storage region 1b are not provided within each block 1, a CAM (content addressable memory) circuit may be used instead. A CAM circuit corresponding to one bit is shown in FIG. 6. The CAM circuit includes a circuit section in which a P-channel MOSFET 21, an N-channel MOSFET 22 and a cell transistor 23 of the flash memory are serially connected between a power supply VCC and a ground and another circuit section having a similar configuration in which a P-channel MOSFET 24, an N-channel MOSFET 25 and a cell transistor 26 of the flash memory are serially connected between the power supply VCC and the ground. In the CAM circuit, the gates of the P-channel MOSFETs 21 and 24 are connected to the drains of the N-channel MOSFETs 25 and 22, respectively, and a bias voltage of about 2 V is applied to the gates of N-channel MOSFETs 22 and 25, thereby outputting CAM data from the drain of the P-channel MOSFET 24 via an inverter 27. In the case of writing data into the CAM circuit, a high-voltage gate signal of about 12 V is input to the control gates of the cell transistors 23 and 26 and a CAM program circuit 28 applies complementary program voltages (one of which is about 7 V and the other of which is 0 V) to the drains of the cell transistors 23 and 26. Then, a write is performed with respect to one of the cell transistors 23 and 26 and a potential difference is caused between the threshold voltages of the cell transistors 23 and 26. Thus, by inputting a gate signal obtained from the power supply VCC to the control gates of the cell transistors 23 and 26, arbitrary CAM

Figure 5.23: A snapshot of the system interface. The left part of the interface is a window displaying the text version of a patent, and the right part is a window display the drawing images of the patent.

5.4 Summary

In this chapter, I introduced a prototype of a patent viewing system. Particularly, we have implemented two core components of this system, namely, the word spotting module and the graphics viewing module. In the word spotting module, we proposed a fast keyword spotting method for patent document images based on Radial Projection Profile. This method avoids unnecessary OCR processing and thus expedites the spotting speed for lengthy patent documents. In the graphics viewing module, we proposed a method to extract captions and labels from diverse drawings of patents, and connected them with their corresponding occurrences in the patent text. This system is a better tool for users to efficiently browse patents than other software available.

Chapter 6

Character/Symbol Recognition in Real Scene Images

With the advancement of camera technology, recognition of characters and symbols in real scene images becomes an extremely important issue, as it is a foundation of many applications. Real scene character/symbol recognition is a broad research topic, aiming at recognizing characters or symbols in real scene images and overcoming all difficulties encountered. Existing rectification methods are not applicable to real-scene images, because they are proposed mainly for camera-based document images. Also, as introduced in Section 2.2 of Chapter 2, some sub-fields of robust recognition have been extensively studied, employing context information specific to this sub-field. However, there is no satisfactory resolution of the more general case. In this chapter, we will tackle this problem by proposing a recognition method resilient to perspective deformation, which is applicable to characters and symbols.

The essence of our method is a global descriptor called Cross Ratio Spectrum proposed by us. The attractive characteristics of the proposed recognition method includes:

- Perspective invariant. This is the only perspective invariant recognition method applicable to simple structure shapes.
- No binarization process is needed. This method works directly on edge images. Signs and symbols are hard to register and binarize properly from the real scene.
- The point level correspondence achieved by our method is helpful to restore the fronto-parallel view of a perspectively deformed image.

In the rest of this chapter, the notation Q refers to the query character/symbol. Similarly, T refers to a template character/symbol:

- Compute a cross ratio spectrum for each sample point on the convex hull of both T and Q.
- Estimate distances between each pair of sample points of T and Q.
- Find the point-level correspondence of Q and T, and estimate the similarity between Q and T.

Since I have introduced the perspective rectification methods in Section 2.3.2, I will only present methods which directly access the content of real scene images in Section 6.1. I will explain our methodology in Sections 6.2 and 6.3. The speed issue will be discussed in Sections 6.5 and 6.6. I will show the experiment results for synthetic character recognition, real scene character recognition, real scene compound symbol recognition in Sections 6.4, 6.7 and 6.8 respectively.

6.1 Related Work

Shape description is a core issue of planar symbol recognition. Many shape description techniques have been developed and reported in the past. A detailed review of shape description techniques can be found in [ZL04]. These shape representation approaches can be divided into two categories: contour based and region based. In region-based approaches, all points within a shape region are taken into account to obtain the shape representation. While in contour-based approaches, the exterior or interior boundary is exploited. It is claimed that, making use of all information of a shape, region-based approaches are more robust to minor boundary noise and deformation, to which contour-based approaches are sensitive to, often at expense of slower speed. However, due to the fact that humans can recognize object solely from its shape, the contour of a shape carries more semantics than the interior region. More importantly, almost all perspective-invariant techniques are contour based, except [SF04]. In this paper [SF04], a way to generate projective moment invariants with a form of infinite series is proposed, however, leaving the discriminating power and convergence problem open.

Followed by the classification in [ZL04], contour-based approaches may be further divided into two sub-categories: structural based, which divides the shape into a sequence of shape primitive, and global based, which represents the shape by a numeric feature vector. Many structure primitives (such as hole, intersection, and concavity) are invariant to perspective deformation. In [LT06a], structural invariants, namely, ascender and descender, vertical runs, and water reservoirs were employed to classify English characters into a reduced symbol set. Nevertheless, structural invariants alone are not discriminating enough to differentiate structure similar shape, such as o and 0. Therefore algebraic invariants drawing from structural primitives are more frequently used than structure primitives, and these methods fall into global based sub-category. A model-based recognition system, called LEWIS [RZFM95], made use of algebraic invariants computed from three primitive sets respectively: five lines, a pair of conics, and a conic and two lines extracted by polygon approximation. Orrite et al. [OH04] used bitangent points in a shape to estimate a transformation between the viewed object and the model object. A drawback of these approaches is that desirable structure primitives are not always present in the shape.

Some methods are structure-independent and thus are more generally applicable. The first one is MPEG-7 visual contour shape descriptors (CSS) [Bob01], which is a global descriptor based on curvature scale space. It represents a shape by features of their curvature scale space image, such as the number of peaks, the height of the highest peaks, and the positions of the remaining peaks. The second one is Scale Invariant Feature Transform (SIFT) [Low04]. SIFT descriptor is a local descriptor based on intensity information. It is a 128-dimensional vector, describing the neighborhood information of a key point. Key points are extracted from an object, as maxima or minima of the DoG images across scales. SIFT descriptor is distinctive, robust to occlusion, and does not require segmentation. A comparative evaluation of local descriptors in [MS05] showed that SIFT descriptor performs significantly better than many other local descriptors proposed in the literature. CSS and SIFT are widely employed in applications where perspective deformation is involved. However, both methods presuppose that the object which needs to be identified is a complex object with great variation in intensity. This assumption might not be applicable to symbols, because they have very simple structure. This point will be explained in the experiment part in detail. Another important descriptor is Shape Context [BMP02], which shows a good ability to handle moderate perspective deformation in [MS05]. More importantly, it shows discriminating ability to recognize symbols with non-rigid transformation in [BMP02]. In the shape context method, each sample point on the shape contour is represented by the distribution of the remaining points relative to it, and a point-to-point correspondence between the query and a template is solved by a bipartite graph matching. After that, a Thin Plate Spline model-based transformation is estimated for a better alignment between two shapes. The distance between two shapes is given by a weighted sum of shape context distance, image appearance distance and bending energy. Iterations are employed for better recognition result. The last method, proposed by Suk [SF04], is a way to generate projective moment invariants with a form of infinite series. However, discriminating power and convergence problems were left open, and was later challenged by [XL07].

In our experiment, CSS, SIFT, and Shape Context methods will be employed for comparison.

6.2 Cross ratio spectrum

Cross Ratio is a fundamental invariant for perspective transformation [MZ92]. The cross ratio of four collinear points (P_1, P_2, P_3, P_4) displaying in order, as shown in Figure 6.1, is defined as:

$$cross\_ratio(P_1, P_2, P_3, P_4) = \frac{P_1P_3}{P_2P_3} / \frac{P_1P_4}{P_2P_4} \tag{6.2.1}$$

where P_iP_j denotes the distance between P_i and P_j. $cross\_ratio(P_1, P_2, P_3, P_4)$

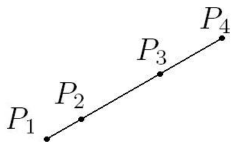


Figure 6.1: Four collinear points.

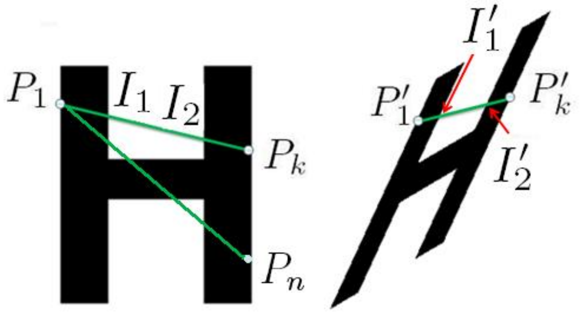


Figure 6.2: Character 'H' in the fronto-parallel view and a perspective view.

remains constant under any projective transformation.

6.2.1 Cross Ratio Spectrum

Figure 6.2 shows a character 'H' under a fronto-parallel view (H) and a perspective view (H'). Suppose pixels $P_1 \in H$ and $P_k \in H$ have mapping pixels $P'_1 \in H'$ and $P'_k \in H'$, respectively. Then I_1 and I_2 (intersections of the strokes and line P_1P_k) have mapping pixels I'_1 and I'_2 (intersections of the strokes and the line $P'_1P'_k$). Consequently, the following equation holds:

$$cross\_ratio(P_1, I_1, I_2, P_k) = cross\_ratio(P'_1, I'_1, I'_2, P'_k) \qquad (6.2.2)$$

For simplicity, we rewrite the cross ratio notation and leave out intersections as $CR(P_1, P_k) = cross\_ratio(P_1, I_1, I_2, P_k)$. When there are more than two intersections

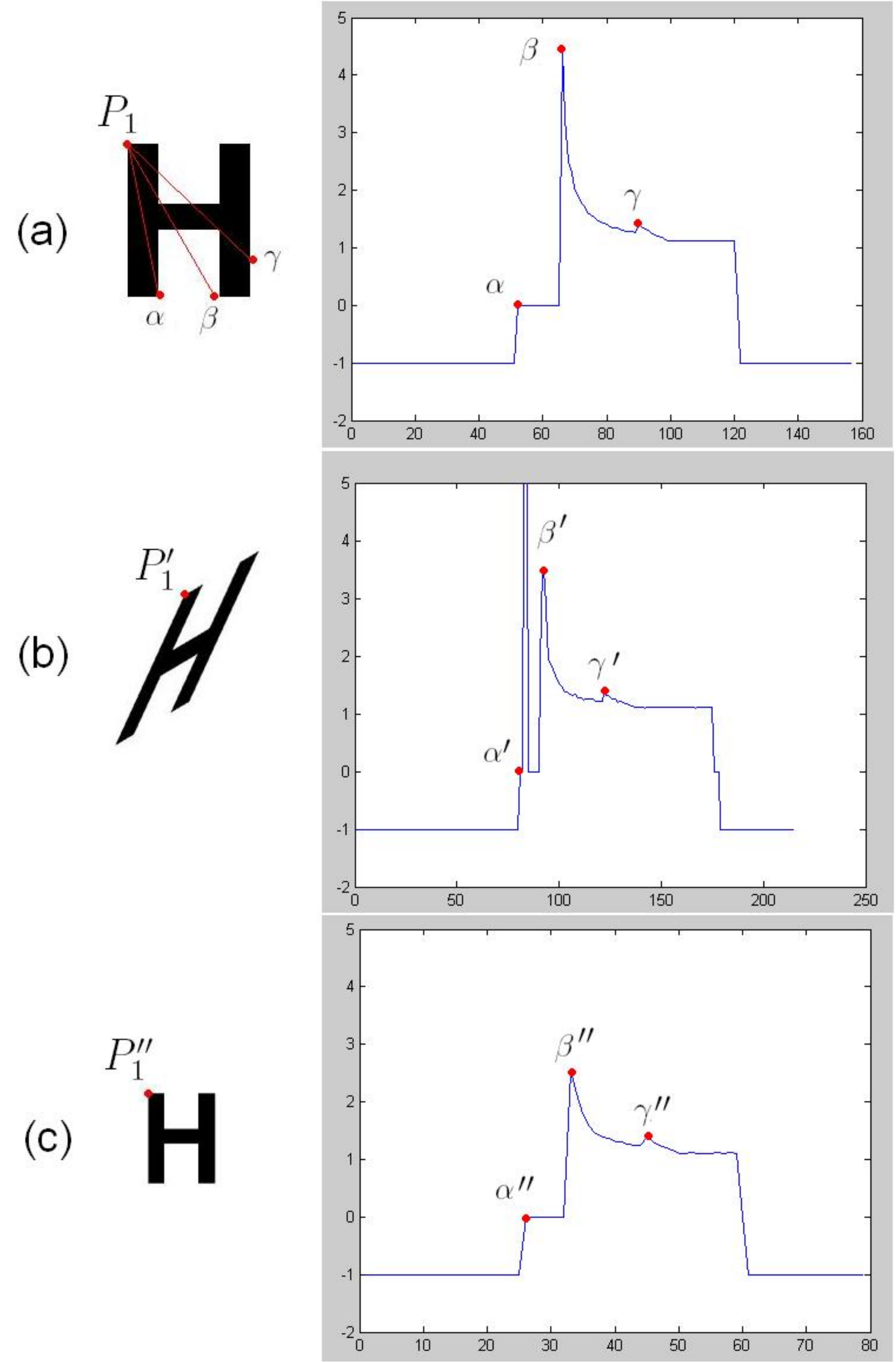


Figure 6.3: Cross Ratio Spectra of mapping points P_1, P'_1 and P''_1.

between two points, such as P_1 and P_n shown in Figure 6.2, only the first two intersections (near P_1) are used. The hypothesis is that, symbols have very simple structures, and the shapes comprising of the first two intersections already give enough information to differentiate them. Some information of the inner structure of a symbol is lost using this method. However, even if the shape comprising of the first two intersections is not distinctive enough, in theory we still can extend our method to employ other intersections in the same manner. Note that intersections at the convex hull itself are ignored in our implementation of the method, to prevent noise from being introduced by a jagged outer contour. If the number of intersections is 0 or 1 and thus no cross ratio value can be computed, the pseudo-cross ratio value (because we can not calculate a cross ratio from only two or three points) is assigned as -1 and 0 respectively. These two values are chosen because cross ratio values range from 1 to ∞. This assignment is to guarantee that both pseudo-cross ratios are distinct from a real one. A cross ratio spectrum is a sequence of cross ratios. The sequence exhibits a wavelet-like form when plotted shown in Figure , and thus we call it "cross ratio spectrum". Suppose the sample point sequence of the convex hull of H is $\{P_s, s = [1 : S]\}$, where P_2 is the anti-clock-wise neighbor pixels of P_1, and so forth. The Cross Ratio Spectrum (CRS) of a pixel P_i is defined as:

$$CRS(P_i) = \{CR(P_i, P_{i+1}), ..., CR(P_i, P_n), CR(P_i, P_1), ..., CR(P_i, P_{i-1})\} \tag{6.2.3}$$

Examples of cross ratio spectra are shown in Figure 6.2.1.

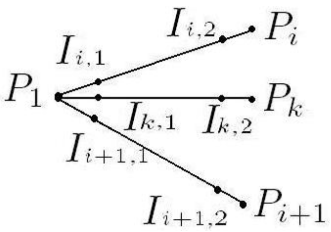


Figure 6.4: A new point P_k is added between P_i and P_{i+1}.

6.2.2 Modeling the Perspective Deformation in a Cross Ratio Spectrum

It is easy to know that the spectrum is intrinsically translation and rotation invariant, given the starting point. Because cross ratio values are collected along the convex contour with respect to the starting point. An example of character 'H' is shown in Figure 6.2.1(a), and its variances are shown in Figures 6.2.1(b)(perspective) and 6.2.1 (c) (scaling) respectively. The cross ratio spectra of three mapping pixels P_1, P_1' and P_1'' are also shown in Figures 6.2.1(a), (b), and (c) respectively, where the x-axis is the pixel index and the y-axis is the cross ratio value. The pixels corresponding to the peaks shown in the spectrum curves (labeled with Greek characters) are shown. Our observations are:

- Visually, three spectra are quite similar to each other, but compared to spectrum a, spectrum b has a certain fluctuation on the x-axis, while spectrum c is scaled along x-axis.
- The value of peaks β, β', β'' are quite different.
- There is a abnormal peak between α' and β'.

The following explains how the first observation happens. Suppose there are two neighboring pixels $P_i \in P$ and $P_{i+1} \in P$, where P is a fronto-parallel character, as shown in Figure 6.4. $I_{i,1}$, $I_{i,2}$, $I_{i+1,1}$, and $I_{i+1,2}$ are the intersections between pixel $P_1 \in P$ and the other two pixels. It is assumed that P_1, P_i and P_{i+1} are on the smooth part of the convex hull (not at the corner), where the stroke width does not change or changes slowly in the neighborhood. Hence $I_{i,1}$ and $I_{i+1,1}$, $I_{i,2}$ and $I_{i+1,2}$ are near to each other, thus $CR(P_1, P_i) \approx CR(P_1, P_{i+1})$. After perspective projection, the segment, to which P_i and P_{i+1} belong, is elongated, new pixels are added between them. Suppose only one pixel P_k is added between P_i and P_{i+1} at the beginning. Similarly, we find $CR(P_1, P_k) \approx CR(P_1, P_i)$. More pixels could be added in the same manner. In short, the cross ratios of those newly added pixels can be approximated by that of the original pixels. Because the number of pixels on the smooth part always dominates, pixels at corners are statistically unimportant. In a nutshell, under perspective transformation, some parts of a character expand, while some parts shrink, which leads to the increase or decrease of the number of pixels on certain parts of the convex hull. As a result, some segments of the spectrum curve are elongated, while others are shortened. Therefore, the perspective deformation in an image can be modeled as an uneven stretching deformation in our spectrum.

The following explains how the second and third observations happen. Following Equation 6.2.2, suppose $s_1 = P_1P_2$, $s_2 = P_2P_3$, $s_3 = P_3P_4$, thus

$$cross\_ratio(P_1, P_2, P_3, P_4) = \frac{\frac{s_1}{s_2} + 1}{\frac{s_1}{s_2+s_3} + 1} \tag{6.2.4}$$

which is an increasing function of variable s_2. s_2 becomes quite short when the line at P_1 passes through a corner, such as at location β, β', and β'', leading to peaks. β, β', and β'' are different due to quantization errors. The reason accounting for abnormal

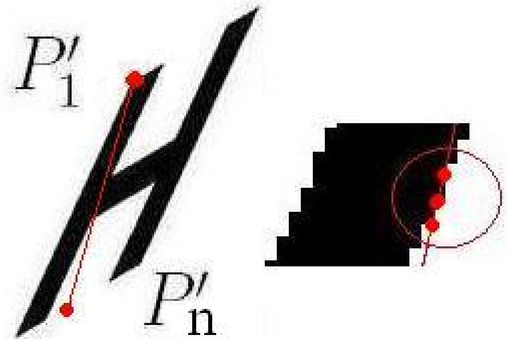


Figure 6.5: False intersections on a jagged inner contour.

impulses in Figure 6.2.1(b) is a jagged inner contour. As shown in Figure 6.5, there should be only one intersection between P'_1 and P'_n, but because of the jagged inner contour, false intersections are detected, leading to a very short s_2. A cost function (Equation 6.2.9) is chosen to minimize the impact of this noise.

6.2.3 Comparing Cross Ratio Spectra

In our method, spectrum b is modeled as an uneven stretching deformation of spectrum a. Hence, we use Dynamic Time Warping (DTW) to compare the similarity between spectrum a and b.

DTW is widely used in speech recognition to eliminate the time-axis fluctuation between the given word and a template [WG97], which is similar to the uneven stretching effect in a spectrum.

Suppose that two sequences $f(i), i = 1 : m$ and $g(j), j = 1 : n$ characterize two signal f and g. The best match between f and g is given by:

$$\arg\min \sum_{w} cost(f(i), g(j)) \tag{6.2.5}$$

where $cost(.,.)$ is a cost function, and $w(.)$ is a warping path between $(1, 1)$ and (m, n)

in a two-dimensional square lattice, given by:

$$w = (w_k, k = [1 : K]) \quad max(m, n) \leq K \leq m + n - 1 \tag{6.2.6}$$

$$w_k = (i, j) \tag{6.2.7}$$

The warping path is subjected to several constraints, which are intrinsically posed by the DTW:

- Boundary Condition: $w_1 = (1, 1)$ and w_K=(m,n).
- Continuity and Monotonicity: given $w_k = (i, j)$ and $w_{k-1} = (i', j')$, $0 \leq i'-i \leq 1$ and $0 \leq j' - j \leq 1$.

The notation of $CRS(Q_i)$ given in Equation 6.2.3 is rewritten as $CRS(Q_i) = \{q_u, u = 1 : M - 1\}$ for simplicity. Similarly, $CRS(T_j) = \{t_v, v = 1 : N - 1\}$. The comparison between Q_i and T_j is formulated as:

$$DTW(u, v) = min \begin{cases} DTW(u - 1, v - 1) + c(u, v) \\ DTW(u - 1, v) + c(u, v) \\ DTW(u, v - 1) + c(u, v) \end{cases} \tag{6.2.8}$$

$$c(u, v) = \frac{abs(log(CR(Q_i, Q_u)) - log(CR(T_j, T_v))}{log(CR(Q_i, Q_u)) + log(CR(T_j, T_v))} \tag{6.2.9}$$

It is observed that large cross ratios are unstable. Also, we found that most cross ratios of symbols are within the range of $(1, 2]$. The $log(CR(\cdot, \cdot))$ representation is used in the cost function $c(\cdot, \cdot)$, in order to reduce the weight of unstable cross ratios and to differentiate common cross ratios within the range of $(1, 2]$ better. If $CR(\cdot, \cdot)$ is -1 or 0, $log(CR(\cdot, \cdot))$ is assigned as -1 and -0.5 respectively. The cost function

(a) The DTW-distance-table

	Q_1	...	Q_M	Q_1	...	Q_M
T_N						
...						
T_1						

(b) A sub-table comprising of column $\{1, ..., M\}$

	Q_1	...	Q_M
T_N			
...			
T_1			

Table 6.1: Planar symbol recognition.

$c(\cdot, \cdot)$ is chosen to minimize the effect of noise, and to maximize the penalty when a pseudo-cross-ratio is misaligned with a real one. The distance between points Q_i and T_j is given by the last item:

$$DTW\_dist(Q_i, T_j) = DTW(M-1, N-1) \tag{6.2.10}$$

6.3 Planar Symbol Recognition

Having introduced the concept of cross ratio spectrum and the comparison method, we will explain in this section how to compare the similarity of two symbols using cross ratio spectra. For Q, the spectrum sequence is defined as $SS(Q) = \{CRS(Q_1), CRS(Q_2), ..., CRS(Q_M)\}$. Similarly, $SS(T) = \{CRS(T_1), CRS(T_2), ..., CRS(T_N)\}$.

The distance between Q and T can be formulated as:

$$SS\_dist(Q, T) = \arg\min \sum_{w\_global} DTW\_dist(Q_i, T_j) \tag{6.3.1}$$

where w_global is the global warping path between (Q_1, T_1) and (Q_M, T_N), as well as the correspondence between Q and T. Our strategy to solve this equation is as follows. An arbitrary sample point of T is chosen as the starting point T_1. T_1 is then aligned with each Q_i as the boundary condition. In particular, DTW comparisons are conducted between each pair of Q_i and T_j, and a DTW distance table is constructed in the manner given by Table 6.1(a). Cells in the table denote the distances of corresponding pixel pairs. Each time, a DTW is applied to a sub-table comprising of column $\{\hbar, \hbar + 1, ..., \hbar + M - 1\}$ of the table, to align T_1 with $Q_\hbar$ and T_N with $Q_{\hbar+M-1}$ as the boundary condition. Table 6.1(b) illustrates a subtable when $\hbar = 1$. The comparison is formulated as follows:

$$DTW(i,j) = min \begin{cases} DTW(i-1, j-1) + c(i,j) \\ DTW(i-1, j) + c(i,j) \\ DTW(i, j-1) + c(i,j) \end{cases} \tag{6.3.2}$$

$$c(i,j) = DTW\_dist\_table(\hbar + i - 1, j) \tag{6.3.3}$$

where $i = 1 : M$ and $j = 1 : N$. A candidate distance between Q and T is given by $DTW(M, N)$. M DTW comparisons are conducted. Among M candidate distances, the smallest one gives the desired global distance.

6.3.1 Character/Symbol Recognition

The proposed recognition method falls into the category of prototype-based recognition, which represents a category by ideal prototypes (templates). In all experiments in this chapter, we took a nearest neighbor recognition strategy: a query is compared with all templates, and the template which has the smallest distance with the query gives the identity of the query. The recognition algorithm is shown in Table 6.2.

```
Algorithm:Recognizing(Q)
$best\_so\_far = inf$
for all templates $T^i$
    $true\_dist = ss\_DTW(Q, T^i)$
    if $true\_dist < best\_so\_far$
      $best\_so\_far = true\_dist$
      $index\_of\_best\_match = i$
    end
end
```

Table 6.2: Scan the prototype set.

6.4 Synthetic Image Testing

In this section, the ability of handling perspective deformation of the proposed method will be illustrated with a well defined synthetic image set. Scale Invariant Feature Transforms (SIFT) with Harris-Affine detector[1], Shape Context[2], MPEG7 contour shape space descriptor (CSS)[3] and a widely used commercial OCR called Scansoft OmniPage Pro 14.0 (OCR) are employed as comparative methods.

Shape context is a global descriptor, in which each sample point on the shape contour is represented by the distribution of the remaining points relative to it, and a point-to-point correspondence between the query and a template is solved by a bipartite graph matching. After that, a Thin Plate Spline model-based transformation is estimated for a better alignment between two shapes. The distance between two shapes is given by a sum of shape context distances. Iterations are employed for better recognition result. Our experiment follows the same process as introduced in

[1] http://www.robots.ox.ac.uk/~vgg/research/affine/index.html

[2] http://www.eecs.berkeley.edu/Research/Projects/CS/vision/shape/sc_digits.html

[3] http://mpeg7.doc.gold.ac.uk

[BMP02].

CSS is also a global descriptor. It represents a shape by features of its curvature scale space image, such as the number of peaks, the height of the highest peaks, and the positions of the remaining peaks. The identity of Q is given by the template which has the minimum CSS score with Q.

SIFT is an local affine-invariant descriptor which describes a local region around a key point. SIFT descriptor is robust to occlusion, and does not require segmentation. A foreseeable problem of applying SIFT descriptor to symbols is the lack of discriminating power. First, many symbols share similar structural primitives, like concavities of 'N', 'Z', 'M', 'V', which are difficult to distinguish using SIFT. Furthermore, the symmetry of a symbol itself causes ambiguity. False matches may happen among symmetric structural primitives of the same symbol. For example, key points at the left bottom of character 'A' may be matched to points at the right bottom of 'A'. In order to solve the structural ambiguity and maximize the recognition strength of SIFT descriptor, the recognition process is designed as follows. A Harris-Affine detector is used to detect Affine-invariant key points. For each key point of Q, its first 20 nearest neighbors are found in the training set. If the distance is less than a threshold (200 in the experiment), the neighbor is kept, otherwise is thrown away. The RANSAC fitting algorithm [FP03] is then used to further filter false matches. False matches (outliers) is removed by checking for agreement, between each match and the perspective transformation model (8 degrees of freedom) generated by RANSAC. The identity of Q is given by the template which has the maximum number of correct matches with Q.

Because the Ominipage OCR assumes that the character ready for recognizing

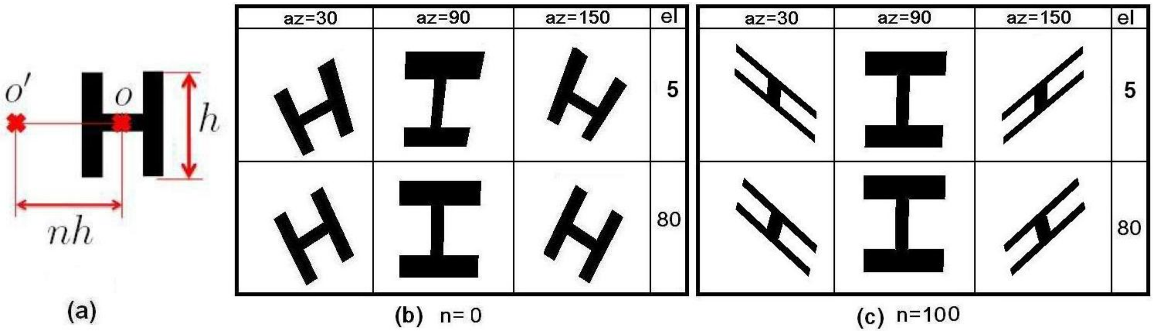


Figure 6.6: Samples of synthetic character images.

should be in a correct orientation, only a slight skew is acceptable. A query image is rotated the azimuth angle backward to tentatively remove the skew.

6.4.1 Experimental Setup

A template set is trained on synthetic fronto-parallel images of 62 characters, namely 26 uppercase English characters, 26 lowercase English characters, and 10 digits, of Arial font and bold style. 18 testing datasets are generated by Matlab using various perspective parameters. Characters ‘l’ and ‘I’ are considered the same in our experiment.

The perspective images are generated by setting the target point at a specific point o', and setting the perspective viewing angle as $25°$ (to model a general camera lens), while changing the azimuth (az) and elevation (el) angles gradually. o' is at the same horizontal line as the center of a character, denoted by o, with a distance of $n \times h$, where n is a positive integer and h is the height of the character. An illustration is shown in Figure 6.6(a). Generally, the larger the n is, the greater the deformation is. For each testing set, n and el are predefined, and az is set as $\{30°, 90°, 150°, 210°, 270°, 330°\}$ respectively. Therefore, each testing set comprises of

(a) Spectrum Sequence

el=	5°	10°	30°	50°	70°	90°
$n = 0$	96.77	97.31	97.04	100	100	100
$n = 50$	96.23	97.04	97.04	100	100	100
$n = 100$	96.23	97.04	97.04	97.84	97.84	100

(b) SIFT

el=	5°	10°	30°	50°	70°	90°
$n = 0$	39.24	45.16	55.91	59.67	67.74	81.18
$n = 50$	11.02	13.70	16.12	17.47	18.81	75.26
$n = 100$	11.55	10.21	11.02	9.94	8.60	65.05

(c) Shape Context

el=	5°	10°	30°	50°	70°	90°
$n = 0$	64.24	67.47	68.27	71.50	73.92	96.23
$n = 50$	15.86	16.93	20.43	18.01	54.03	66.39
$n = 100$	15.32	13.44	15.05	14.51	50.00	68.27

(d) OCR

el=	5°	10°	30°	50°	70°	90°
$n = 0$	92.47	92.47	97.84	97.84	100	100
$n = 50$	0	0	2.68	3.22	3.49	98.65
$n = 100$	0	0	0	0	2.41	98.11

Table 6.3: Recognition accuracy of synthetic images.

$6 \times 62 = 372$ characters. Examples of the character 'H' under different perspective parameters are shown in Figures 6.6(b) and (c). We found that perspective deformation varies greatly along with the elevation angle when the target point is at the center of the character. In contrast, when the target point is far away from the character, the variation tends to be small.

6.4.2 Experiment Results

Tables 6.2(a), 6.2(a), 6.2(c), and 6.2(d) show the recognition accuracy using our method Spectrum Sequence, SIFT, Shape Context, and OCR methods respectively, where accuracy is the number of correctly recognized characters over the total number of query characters. The accuracy in each cell is based on a testing set comprising of 372 characters generated with corresponding perspective parameters. The result of CSS is not tabulated, simply because it cannot distinguish simple symbols. CSS descriptor gave the best score to about 10 to 20 templates in each run. However it still showed a certain resistance to moderate perspective deformation.

It is easy to see that when characters are deformed by perspective projection, our method has a better recognition accuracy than other methods. Generally, when n increases or el decreases, the deformation becomes more severe. Table 6.2(a) shows that the performance of our method degraded only a little with increasing deformation. In particular, the character pair ‘O’ and ‘D’ are responsible for most errors due to the high similarity between both characters. It is worth noting that, the proposed method gave small distances to visually similar characters like ‘W’ and ‘w’ but made few mistakes in differentiating them. For the performance of SIFT descriptor shown in Table 6.2(b), when the perspective deformation is small, such as $n = 0$ or $el = 90°$, errors are mainly caused by the structural similarity of characters. However, when the deformation is more severe, the descriptor is less resistant to the deformation. An illustration can also be found in Figures 6.7(c) and (d), where SIFT fails to find correct corresponding points between two characters under severe perspective deformation. Table 6.2(c) shows that Shape Context is not that robust to severe perspective deformation too. One possible reason is that both SIFT and

Shape Context descriptors are statistics based, but the expansion/shrinking effect in perspective deformation obviously will affect the statistics. Table 6.36.2(d) shows the recognition result gotten by OCR. With necessary preprocessing, OCR achieved an accuracy as high as above 92%, when the deformation is moderate. In particular, ‘O’ is mis-recognized as ‘0’ for almost all testing datasets with perspective deformation. However, when the deformation became more severe, the performance of OCR drops rapidly. One reason for the rapid degradation is that rotating az degree backward may not turn the character into a right position. In this case, only a few characters can be recognized.

Figure 6.7 shows the pixel-level correspondence, between a query ($az = 30°, el = 5°, n = 100$) and a template, achieved by our method Spectrum Sequence, Shape Context and SIFT respectively. Both Shape Context and SIFT methods fail, due to the severe perspective deformation as well as structural similarity of characters. Thanks to the flexibility of DTW comparison, our method is tolerant to image defects to a certain extent. Figure 6.8 shows the correspondence using our method when the character image is impaired in different ways: the bottom part of the character is truncated in Figure 6.8(a); the interior contour of the character is breached in Figure 6.8(b); the exterior contour of the character is eroded in Figure 6.8(c). Although the alignment is not as accurate as that shown in Figure 6.7, Spectrum Sequence method is still robust to such impairments and is able to align two images accurately.

To compare a query with one template, each of which has 100 sample points, it took 2.62 seconds for our method implemented in Java, on a PC configured with Pentium 4 CPU 3GHz, 0.99GB of RAM. We also implemented the method in Matlab to compare it with other methods. It took 10.23 seconds for our method (Matlab,

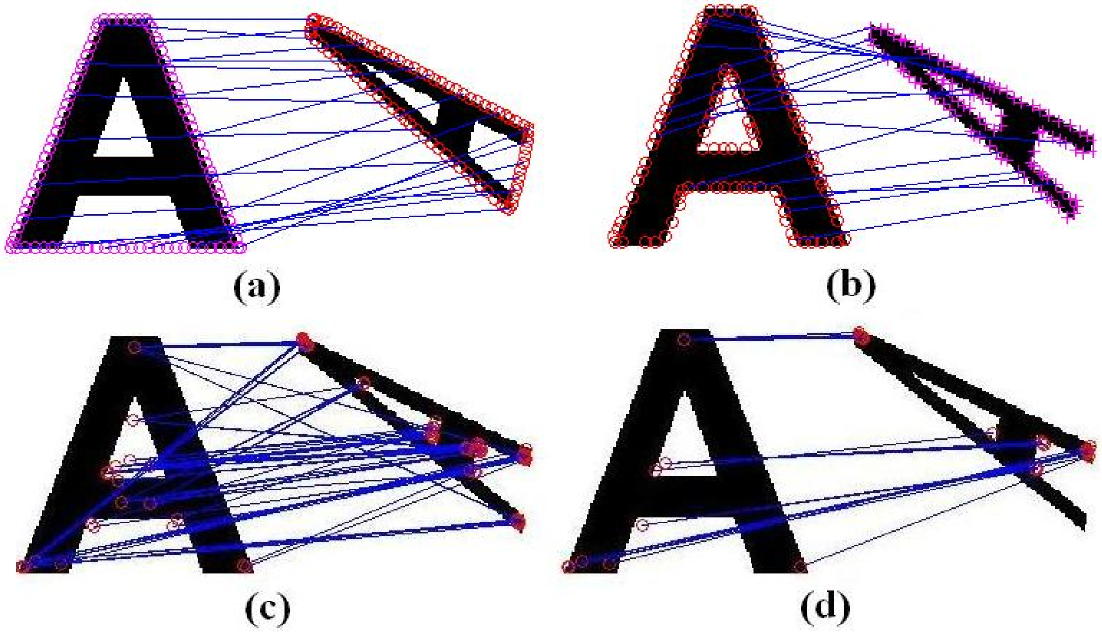


Figure 6.7: Pixel level correspondence of a template and a query generated by (a) our method, (b) Shape context, (c) SIFT, (d) SIFT with RANSAC.

100 sample points), 6.86 seconds for Shape Context(Matlab, 100 sample points), 6.61 seconds for Harris/SIFT/RANSAC (Linux/Matlab), 0.06 seconds for Contour Shape Space (C++) methods, and 0.26 seconds for OCR.

6.5 Speed Issue Discussion

The main drawback of the proposed method is the speed. A query Q has to be compared against a whole set of templates. Also, when Q is compared with a template T, comparisons between each pair of T_i and Q_j are needed. Constructing the DTW-distance-table needs $M \times N$ DTW comparisons. Moreover, searching the optimal warping path in the table takes M DTW comparisons. The time complexity of a single DTW comparison is $O(M \times N)$. Therefore, the time complexity for comparing

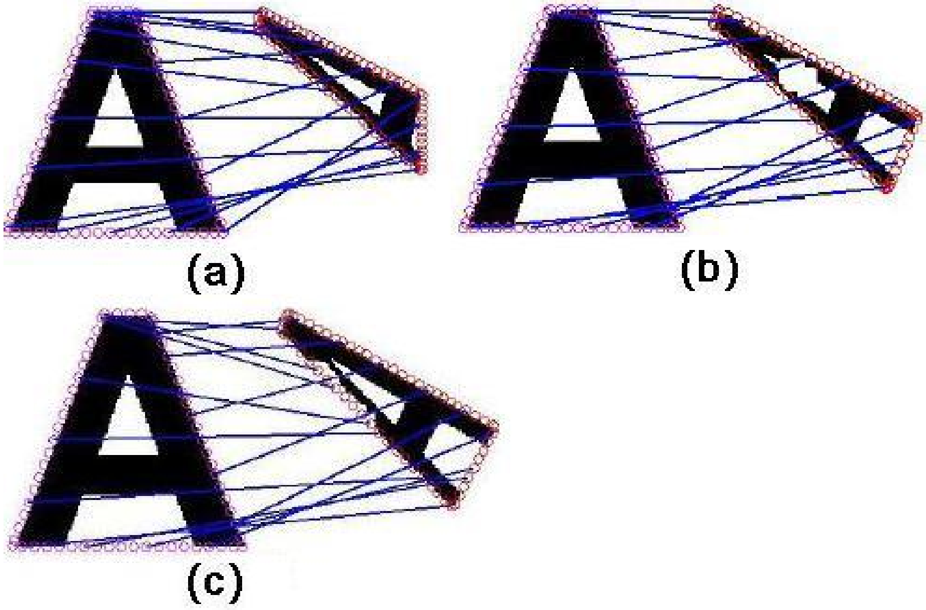


Figure 6.8: Pixel level correspondence of a template and impaired queries.

Q and T is $O(M^2 \times N^2)$.

An important observation is that many neighboring points have similar spectra. An example is shown in Figure 6.9. Two groups of neighboring points are labeled with $*$ or $\circ$ marker respectively, and spectra of points are labeled with the same markers. Spectra of the same group are similar to each other, by x-axis shifting. On the contrary, spectra of different groups are different. This phenomenon indicates two possible solutions to the speed problem, namely, reducing the number of sample points for each character and indexing templates by grouping neighboring points together.

6.5.1 Effect of the Number of Sample Points

It is easy to see that the essential factor which affects the speed is the number of points sampled on the convex contour of a character. In this subsection, this issue will be discussed. The recognition accuracy and speed of sampling $k = \{5, 10, 15, 20, 25, 30, 35, 40,$

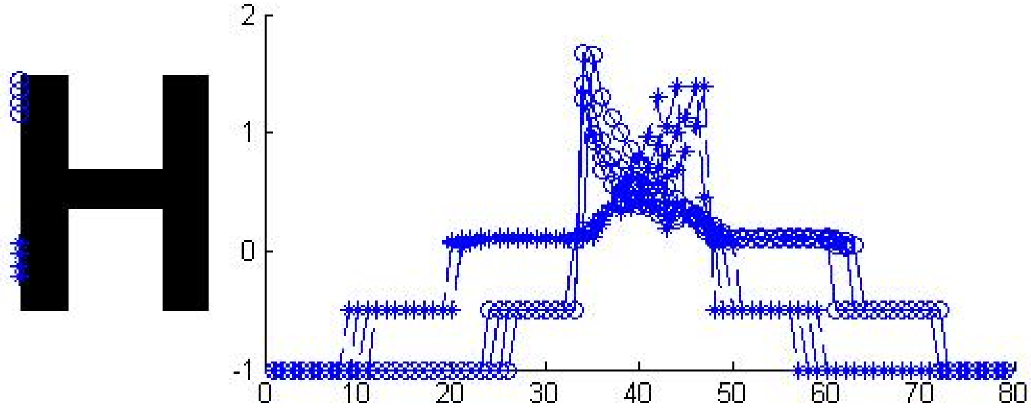


Figure 6.9: Neighboring points having similar spectra.

60, 80, 100} points on each character (both template and query) are shown in Figure 6.10. The speed for different k is shown as a ratio to the speed when $k = 100$(2.62 seconds). The result is based on one synthetic dataset (372 queries with parameters el = 5, n = 100). When k is greater than 60, the recognition accuracy remains very close to 96.23%.

We also run the same experiment for another 10 times, randomly changing the starting point of the sample points, and this has little impact on the final recognition accuracy when the number of points is larger than 40. When the number of sample points is less than 20, the recognition accuracy varies more than 10% as the starting point changes. A possible reason is that the sample points are too sparse to capture the shape of a character.

Furthermore, we scale characters in the testing dataset by 2, 4, and 8 times respectively, and use the scaled dataset as input. Because scaling will also affect the distribution of sample points. The results show that the size of query will not affect the recognition accuracy, too. In a nutshell, although the speed is bi-quadratic to the number of sample points N, N is nearly fixed in a character recognition application

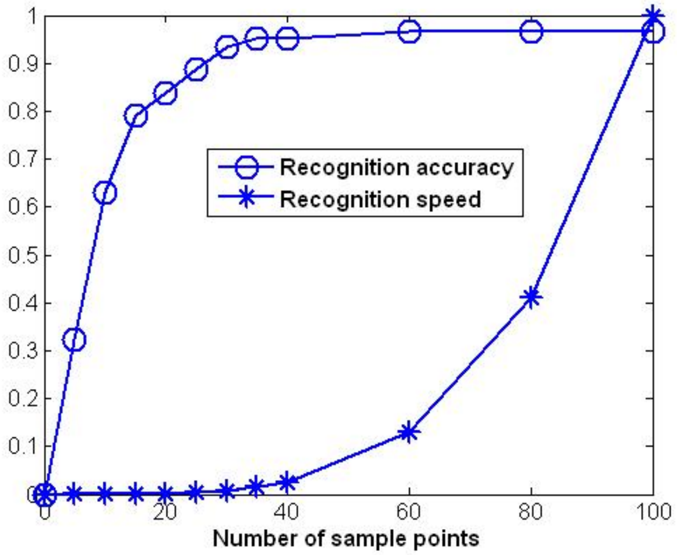


Figure 6.10: The recognition accuracy and speed with different number of sample points.

for all sizes of characters. Further, an appropriate N could be estimated by a training process, given that all templates are available.

6.5.2 Improving Accuracy by Iteration

In this experiment, errors happen within character pairs which are similar to each other, for example ‘d’ and ‘p’, ‘w’ and ‘W’, and ‘9’ and ‘6’. Although these errors could be addressed by considering context information, it is also possible to increase the accuracy using information generated by our Spectrum Sequence method. The way to increase the recognition accuracy by interaction with the point-level correspondence information generated by our Spectrum Sequence method is as follows: if the ratio of distances for Q to the nearest and the second-nearest templates is greater than 95%, an iterative comparison takes place. The perspective transformation matrix M with 8 degree freedom is estimated by Least Squares Fitting, taking all correspondences as input. A temporary template image is generated with M, and it is compared to Q again. With one additional iteration, we were able to improve the accuracy to 98.38% for the dataset ($el = 5$ and $n = 100$).

6.6 Indexing Templates

Indexing and searching time series has been extensively studied, and many speed optimization techniques have been proposed. A short review of these expedition techniques is made as follows, and the reason why they are not suitable for our method are explained too.

Constraints: limits the number of cells that evaluated in the cost matrix [Ita75, SC78]. It can be formulated as: given $w_k = (i, j)$ and $w_{k-1} = (i', j')$, $i - i' \leq G$ and $j - j' \leq G$. Global constraints will slightly speed up the DTW comparison, and more importantly, it will prevent over-fitting, where a small section of one spectrum maps onto a large section of another. In the experiment, G is set as $0.3M$, which is a loose constraint, because the perspective deformation may be quite severe.

Data abstraction: performs DTW on a reduced representation of time series [KP00]. A very important algorithm toward faster DTW algorithm based on iterative data abstraction, named FastDTW, was proposed in [SC07]. It is an accurate approximation of DTW, which has a linear time and space complexity. This algorithm uses a multilevel approach that recursively projects a solution from a coarser resolution and refines the projected solution. However, when the length of the time series in less than 1000, the running time is almost the same as a normal DTW algorithm [SC07]. In our experiment, a cross ratio spectrum has a length around 100.

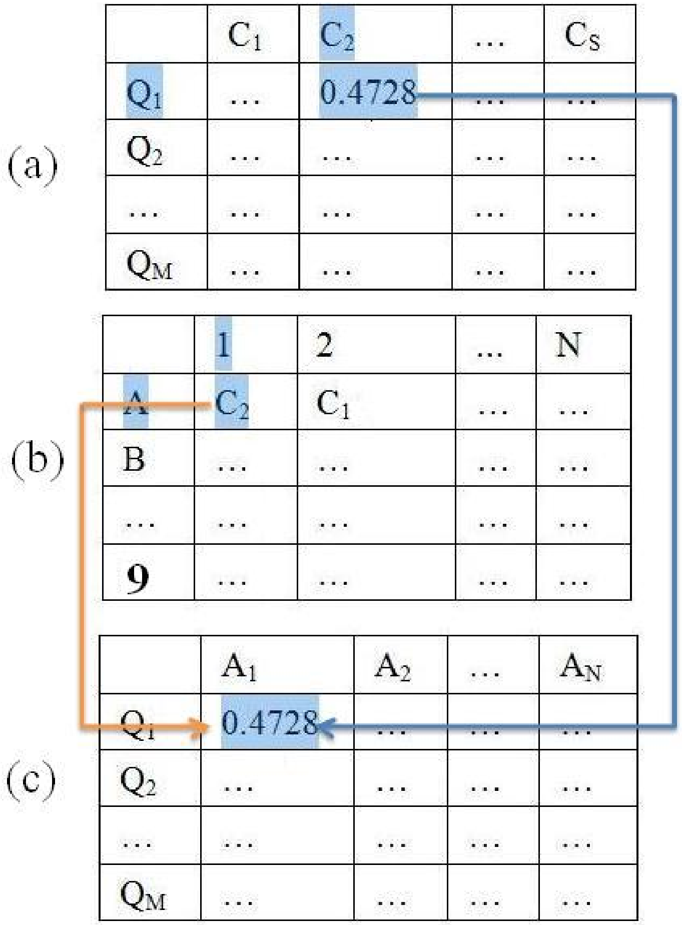
Lower Bound Indexing: reduces the number of candidate templates. Indexing time-series is aiming to reduce the number of times to conduct DTW. Many methods have been proposed to index time series in sound databases. An efficient indexing, retrieval and visualization framework for large scale of time series can be found in [FL95]. An survey about data indexing and retrieval in time series databases is presented in [KK04]. The dominant approach of indexing time series is based on a lower bound technique. Lower bound function estimation is an essential technique in Time Series Indexing, which is used to eliminate undesirable template candidates efficiently. Lower bound function $LB(\cdot, \cdot)$ provides an estimation of $LB(Q_i, T_j) < DTW\_dist(Q_i, T_j)$, that is, the possible minimum distance between Q_i and T_j. Two important lower bounding functions are proposed in [GP95] and [YJF98]. The first method extract a vector comprising of 4 elements, namely, the first, the last, minimum, and maximum elements of the sequence. The bounding function is given by the sum of distance between each corresponding elements of both sequences. The second method takes advantage of the observation that all the points in one sequence, that are larger (smaller) than the maximum (minimum) of the other sequence, must contribute at least the squared difference of their value and the maximum (minimum) value of the other sequence to the final DTW distance. However, both lower boundary functions provided by [GP95] and [YJF98] give too loose lower boundaries for our application, and gives almost no optimization on the speed when applied, because of the simple and similar structures of the symbols.

6.6.1 Optimized Recognition Method with Indexing

In view of the above, we propose a clustering based indexing method in the rest of this section, which indexes points of templates which have similar CRS.

K-means clustering algorithm was selected because it can specify the number of clusters we would like to have. After importing all the CRS information of template images, DTW is performed between each pair of CRS to determine their mutual distance. This is an expensive operation too, but since it is a one-time operation and can be treated as the training process, its complexity does not affect our actual recognition speed. The centroids of each cluster were chosen to be the one with minimum DTW distance to the rest of spectra in the cluster.

Suppose the number of clusters is S, and we denote the centroid of clusters as $\{C_s, s = 1 : S\}$; the number of templates is R, and we denote the templates as $\{T_r, r = 1 : R\}$; remember that the number of sample points for a template image is N and for a query image is M. During training, a cluster index table is built with dimension of $(N \times R)$ as shown in Figure 6.11 (b). For example, for template "A", its first sample point falls into cluster C_2, the second sample point falls into cluster C_1, etc. When a query comes in, the first step is to calculate the DTW distances between points of the query $\{Q_i, i = 1 : M\}$ and clusters $\{C_s, s = 1 : S\}$. The results are stored in a temporary table as shown in Figure 6.11(a). For instance, the DTW distance between Q_1 and C_2 is 0.4728. When a query is evaluated, a DTW distance table must be built, such as Figure 6.11(c). This time, the table can be filled directly by reading results from the cluster index table and the temporary table. For example, if we wish to get the DTW between Q_1 and A_1, the cluster index table will tell us that A1 belongs to cluster C_2, and then by using DTW result table, the DTW comparison



(a)

	C_1	C_2	…	C_S
Q_1	…	0.4728	…	…
Q_2	…	…	…	…
…	…	…	…	…
Q_M	…	…	…	…

(b)

	1	2	…	N
A	C_2	C_1	…	…
B	…	…	…	…
…	…	…	…	…
9	…	…	…	…

(c)

	A_1	A_2	…	A_N
Q_1	0.4728	…	…	…
Q_2	…	…	…	…
…	…	…	…	…
Q_M	…	…	…	…

Figure 6.11: Tables used in the optimized method: (a) Temporary table (b) Cluster index table (c) DTW distance table.

Table 6.4: Average recognition speed and accuracy per query.

Clusters	140	120	100	80	60
Time(s)	2.88	2.23	1.58	1.13	0.72
Accuracy(%)	93.81	92.20	91.67	76.61	55.10

result of Q_1 and C_2 can be retrieved directly and propagated into the corresponding cell of DTW distance table. Originally, if each query is compared with R templates, each time a DTW distance table is formed by performing $(N \times M)$ DTW comparisons, followed by DTW comparisons for M sub-tables. After clustering, $(M \times C)$ DTW comparisons are needed to construct the DTW result table, and for each template, M comparisons are to be made without any other overheads. Thus, the number of DTW operations needed is reduced drastically from $R \times M \times (N+1)$ to $(C+R) \times M$.

6.6.2 Experiment Results

Two experiments are designed and conducted in order to examine the proposed method. The first experiment aims to show how the number of clusters will affect the speed and accuracy. The second experiment is to show speed improvement by clustering; and the second one is designed to show how the degree of perspective deformation will affect the accuracy.

First, we would like to examine how the selection of number of clusters will affect the recognition speed and accuracy. In this experiment, the dataset with parameters$\{n = 100, el = 10\}$ is used, which has the most severe perspective distortion. Various numbers of clusters from 60 to 140 are selected as shown in Table 6.4. Each time, k-means clustering is performed with a specific number of clusters on the CRS of all template sample points. The centroids are saved and stored into a table so that during the actual recognition process, the system can directly read in the table and process it accordingly. Both methods are implemented in Java, and run on a PC configured with Pentium 4 CPU 3GHz, 0.99GB of RAM. 60 points are sampled on each character.

The original method takes 70.43 seconds to process a query (comparing against 62 templates). Table 6.4 shows the effect of number of clusters on both the recognition speed and accuracy. We can observe that by decreasing the number of clusters, the recognition time is further reduced. At the same time, the accuracy has been tradeoff to a certain extent as well. We also find that the execution time decreases almost linearly when the number of clusters decreases. However, the accuracy drops suddenly from 100 clusters to 80 clusters, and thus in the following experiment, the number of clusters is set as 100. This expedites the recognition process by up to 40 times. With 100 clusters, the total number of DTW comparisons needed is reduced to 4.2% of the original algorithm. Also, the index will largely reduce computational overhead of the original algorithm when implemented.

In the second experiment, all 12 testing datasets are used for both the original method and the optimized method. For clustering optimization, all the 62 templates are preprocessed by performing a k-means clustering algorithm on all the CRS of template sample points. Tables 6.5(a) and 6.5(b) show the time and accuracy comparison of the two methods. We can observe that with 100 clusters to represent all the CRS patterns, the accuracy drop with a reasonable limit of 6 percent but the recognition speed has been improved by 40 times. Errors often occur within characters with similar shape, like ‘b’ and ‘q’. Another important finding is that, even

Table 6.5: Average recognition accuracy per query for the original method and the optimized method.

(a) the original method

el=	10°	30°	50°	70°
$n = 0$	97.31	97.04	100	100
$n = 50$	97.04	97.04	100	100
$n = 100$	97.04	97.04	97.84	97.84

(b) the optimized method

el=	10°	30°	50°	70°
$n = 0$	93.27	93.81	94.35	100
$n = 50$	91.93	93.27	94.08	100
$n = 100$	91.39	93.27	91.93	93.81

though the recognition result might not be correct, it is most likely that the top 3 templates contain the answer.

6.6.3 Coarse to Fine Matching

Although indexing templates by grouping similar CRS will reduce the recognition speed largely, about 5% accuracy is lost consequently, mainly due to some similar characters. Therefore, we employ a two level coarse-to-fine matching scheme to further enhance the recognition accuracy, which still maintains the original quick speed. Firstly, n nearest templates of Q are identified by the optimized algorithm, and then these n templates are re-ranked by the original comparison algorithm.

6.7 Real-Scene Character Recognition

Robust character recognition is a broad research topic. It aims to recognize characters in real scene images and solve various difficulties encountered in recognition, including

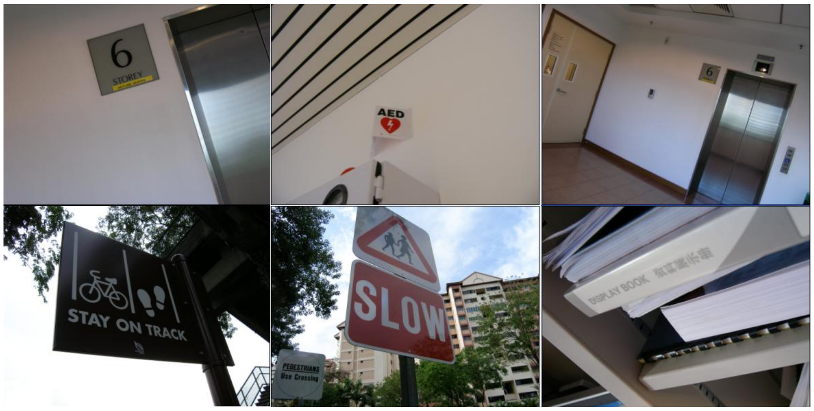


Figure 6.12: Examples of sign boards in real scene.

uneven illumination, occlusion, blur, highly decorated fonts, as well as perspective deformation. In this section, we will try to tackle this problem partially by handling severe perspective deformation, while keeping the other difficulties mentioned at a moderate level.

In this experiment, 100 sign boards are chosen. For each signboard, 4 photos are taken from different angle and distance, leading to 400 photos in total. Examples of these photos are shown in Figure 6.12. These photos are divided into training and testing datasets. Training dataset has 1 photo of each signboard, which are clear and nearly fronto parallel. Testing dataset has the other 3, which have more severe perspective deformation. Words are extracted by the method proposed by Chen [CY]. In order to avoid errors introduced by the extraction algorithm, non-character elements (here a character means either an English character or a digit) are manually eliminated in both training and testing datasets. Then the edge of a character is extracted by the Canny algorithm [CG86]. In order to remove undesirable edges caused by shadows or dusts on signboards, edges with a length shorter than $e = 0.02 \times \sqrt{s}$ are removed, where s is the area of the character bounding box.

We have gotten more than 1000 training characters in a few fonts from the training set. Many characters share the same font. To remove duplicate characters with the same font, a template set is trained from the training set as follows. The template set Γ is initialized with 62 synthetic characters described in Section IV. If a character of the training set could not be recognized correctly, the character is added to Γ. After running this for all characters, a template set, comprising of 182 characters, is ready for use. It is necessary to supplement the synthetic template set because of the wide variance of fonts used in real scenes. Also, the usage of a noise-free synthetic template as initial set has been proven to be helpful to improve the recognition accuracy. A clustering based indexing is applied on the template set. In recognition, after identifying the first $n = 5$ candidates, the full comparison is used on these candidates.

The testing set is further divided into three sub-sets. Set I has 923 characters, which can fit into a bounding box of 50 pixels; Set II has 1296 characters smaller than 100 pixels; Set III has the remaining 1026 characters. The result shows that the proposed method achieved an recognition accuracy as 70.53% in the testing Set I, 91.04% for Set II, and 92.69% for Set III. The recognition performance degrades when the character size gets smaller. Because when the character is small, it is very likely that the edge detected will be broken or connected to edges of background objects. For example, when a signboard is far away and there happens to be an object near it, the edge detector may not be able to generate a correct edge of the characters on the signboard, such as shown in Figure 6.13(b). However, when we used a threshold method to extract the word as shown in Figure 6.13(c), it is correctly recognized.

With this matching scheme (60 points, 150 indexing clusters, 5 nearest neighbors), it took 7.52 seconds to process a query over 182 templates on average.

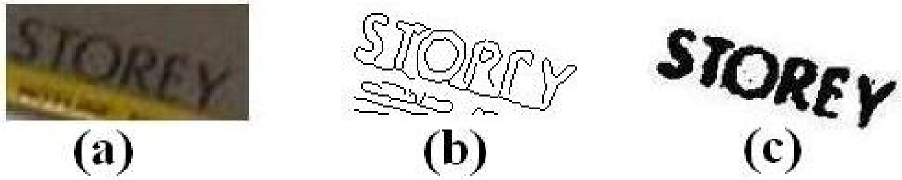


Figure 6.13: (a)Difficult testing photos in real scene.(b)The edge detection result of (a). (c) The binarization result of (a).

6.8 Real Scene Compound Symbol Recognition

Traffic signs, were selected in our experiments representatives of compound symbol comprising of several components. Traffic sign recognition is an extensively studied area, and has been implemented in Driver Support Systems, making use of a full range of features including color, shape, and texture. Traffic sign recognition generally emphasizes on fast recognition speed and robustness to motion blur. Perspective deformation normally is not a main problem addressed, because the camera on a vehicle is often far from the traffic sign. However, we chose traffic signs for our study here for ease of symbol detection and availability of a comprehensive template set. A subset of a standard traffic sign database[4](45 signs with red or blue frames) is employed as the template set. Both SIFT and Shape Context, with the same setting in Section IV, are employed for comparison, considering that road signs are more distinctive to each other than characters.

For a compound symbol with several components, a recognition strategy is to disassemble the symbol into components and then recognizing them separately. However, this strategy is not employed here, because many traffic sign segmentation algorithms

[4] http://en.wikipedia.org/wiki/Road_signs_in_Singapore, last modified on 26 Nov. 2008

	Our method	SIFT	Shape Context
SetI	94.79	72.26	87.89
SetII	92.18	72.39	81.25

Table 6.6: The recognition accuracy of traffic symbols.

such as [dlEMSA97] are already available. We do recommend using an extra segmentation step for multiple-component symbols in order to increase the distinctiveness of symbols. In particular, although the distance is not directly related to the number of components in a symbol, more components often lead to greater curvature in a symbol, and thus traffic signs often have greater average distance as well as greater distance range among each other than characters.

These 3 photos for each of 100 traffic boards are taken, and examples are shown in Figure 6.14. Many of them have elevation angles smaller than 20°, leading to severe perspective distortion of the traffic boards. In the experiment, we employ a simple yet effective color thresholding method proposed in [dlEMSA97] to detect signs with red and blue frames, with hardcoded color boundaries. Desirable traffic signs, 415 in total, are extracted, because some photos have more than one traffic signs. Then the edge is extracted by Canny algorithm, too. Edges with a length shorter than $e = 0.01 \times \sqrt{s}$[5] are removed, where s is the area of the bounding box. The parameters used in this experiment are 80 sampled points, 100 indexing clusters, and 3 nearest neighbors, based on a preliminary experiment on synthetic traffic symbols.

The testing dataset comprising of 300 photos is divided into two sets: Set I comprising of 256 signs whose sizes are within a 80 × 80 pixel bounding box, while Set II comprises of the remaining 192 signs. Table 6.6 shows the recognition results for our method, SIFT, and Shape Context respectively. The recognition performance of

[5] The setting is changed because usually traffic signs have larger size than real scene characters.

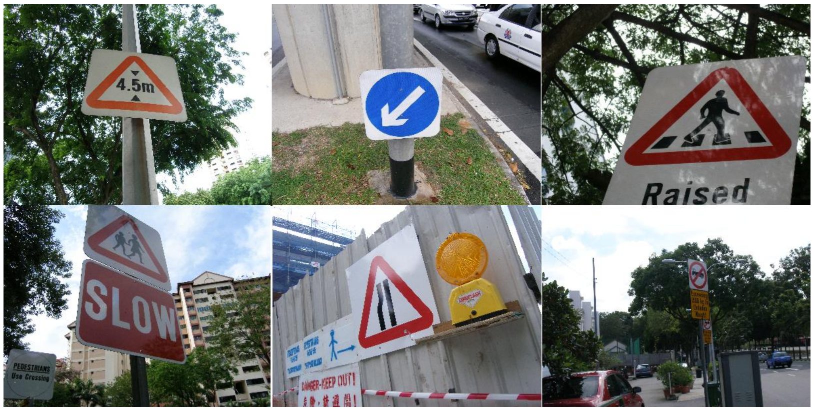


Figure 6.14: Samples of testing data.

our method is slightly better than character recognition. Two reasons may account for this. First, traffic signs are more distinctive from each other. Second, traffic signs used in the experiment have larger size and are more distinct from the white background, which is good for edge detection. Both SIFT and Shape Context methods give much better recognition results in this experiment than in the synthetic image experiment. One possible reason is the distinctiveness of traffic symbols. Another possible reason is that the perspective deformation appearing in most photos in this dataset is not as severe as those used in Section IV. In Section IV, characters with different perspective deformations are evenly distributed. However, due to physical constrains in real scene, traffic signs with severe perspective deformation are only a small fraction of the testing data. Figure 6.15 shows the results of rectifying two symbols by our method, SIFT, and Shape Context respectively, using the Least Squares method to evaluate a transformation model based on the correspondences achieved by these three methods.

The major noise is caused by edges of shadows and other objects on the traffic board. For example, Figure 6.16 shows the alignment of a deformed symbol. Short

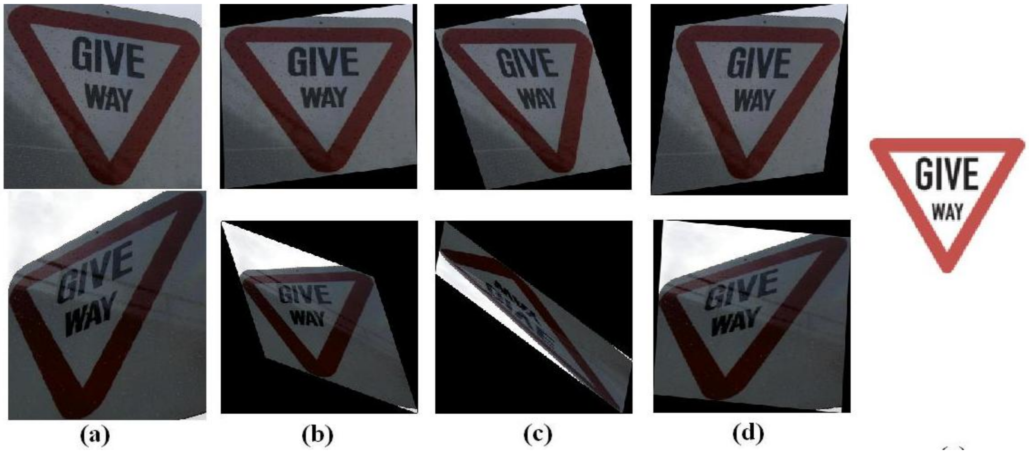


Figure 6.15: Rectify photos by the correspondence given by different methods, rectified images are scaled for better viewing purpose. (a) a real-scene symbol (b) by our method (c) by SIFT (d) by Shape Context (e) the template.

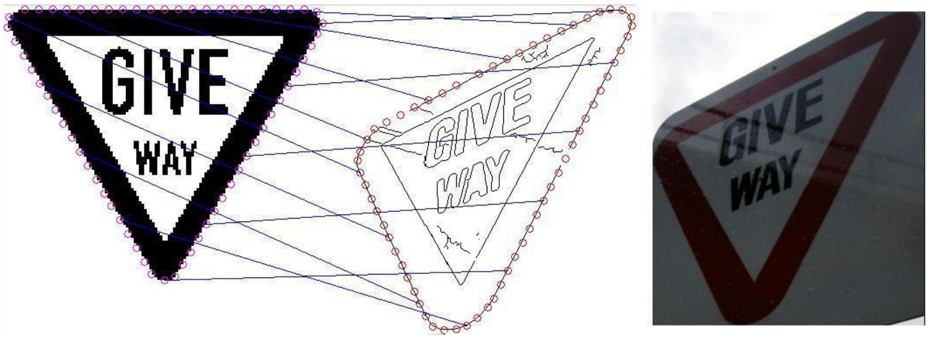


Figure 6.16: Pixel-level correspondence of a template and a deformed query.

edges produced by the shadow and rain drops remain after the preprocessing. In order to improve the accuracy, a better way to remove this noise should be included. Also, some part of the contour of the symbol is missing due to low contrast. However, our method is still able to give a correct alignment. The proposed method is also capable of giving a correct alignment between similar symbols. An example is shown in Figure 6.17, where the parts beneath the person are different between the two signs. In the experiment, it took 2.64 seconds to process each query over 45 templates on average.

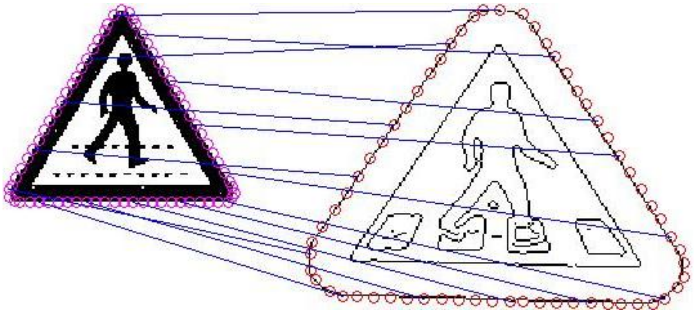


Figure 6.17: Pixel-level correspondence of two similar, but not identical, symbols.

6.9 Summary

In this chapter, I have presented a planar character/symbol recognition method based on a descriptor named Cross Ratio Spectrum proposed by us. In a cross ratio spectrum, the 3-Dimensional perspective deformation is mapped into 1-Dimensional stretching deformation, and thus can be solved by Dynamic Time Warping technique. This method also gives a point-level correspondence between a symbol and its perspective form, and thus helps to recover the transform matrix. We have also proposed a clustering based indexing method to expedite the recognition process. However, one issue that has not been addressed in this thesis is occlusion. It is a common problem for all global descriptors, we will address in the future, by exploring to apply the cross ratio spectrum on a local image patch instead of a whole symbol.

Chapter 7

Conclusion

7.1 Contributions

The main contribution of this thesis is to propose two methods to directly access the content of imaged text captured by cameras. In contrast to existing textual information extraction techniques, both methods totally avoid the perspective rectification step which takes extra time, causes errors, and fails in the absence of certain text layout.

In particular, we have proposed a character/symbol recognition technique applicable to real scene images with severe perspective deformation. This method is the first time to propose a truly perspective invariant solution to shape description. Existing solutions reduce the degree of difficulty by approximating with an affine deformation. It is an extremely difficult problem in this field. On one hand, traditional extraction methods fail to handle perspective deformation. On the other hand, the popular Affine-invariant descriptors have inadequate discriminating ability for characters and symbols with simple structures. Although effort has been made to remove perspective effect from camera-based document images, few solutions can be applied to real-scene images. Our character/symbol recognition method is capable of directly recognizing characters with severe perspective deformation. It also manifests robustness to image noise, tolerance to font variation as well as strong discriminating ability among similar characters. Another significant advantage of this method is its flexibility. By using universal features, it can be easily extended to other planar symbol set regardless of component cardinality. In addition, rectification techniques can be developed based on the correspondence output by this method. This work should facilitate those applications of camera-based image processing, which have to deal with severe perspective distortion, such as Sign Recognition, Mobile Phone Translator, and Speech Generator for the visually impaired.

For camera-based document images, we have proposed a word shape coding method, which is robust to perspective deformation. It is robust to different character styles and is also resilient to certain font variation. Based on this word shape coding method, a language identification technique for camera-based document images is developed. It is an essential step before other retrieval applications can be applied to camera-based document images in a multi-lingual environment. In addition, when there is no reliable OCR is available for camera-based document image, this method can be employed as the alternative of OCR for certain retrieval applications, such as duplicate detection. It is an application of finding similar or equivalent document images with camera-based document images as queries. It is as an important task for realizing digital desk environment.

Another practical contribution of this thesis is to propose methods and construct tools for the users of the U.S. patent database, providing them with a better patent viewing system. In particular, we have proposed a fast word spotting method for the word spotting module. It is invariant to Euclidean deformation as well as robust to noises like broken and touching characters. This module will solve the problem bothering many patent users that they have to wait for a long time before they can first search within a lengthy patent. Our method speeds up the keyword spotting by about 30 times than OCR alone, it will give patent users a faster and smoother keyword spotting experience. Since this method is language independent, it can be easily adapted to patents in different languages, especially suitable for databases like the European Patent database with multi-lingual patents. Another advantage of this method is that it able to estimate the skew angle at word level concurrently with spotting. Therefore, it may be used for rectifying the image if necessary. More importantly, it is a promising technique for skew detection for images with sparse text. In addition, for the U.S. patent database, we have also developed a graphics viewing system, to connect the captions and labels of figures in the drawing section to their relevant text description. With the help of this system, a patent user can conveniently jump from a figure to relevant text or vice versa by clicking corresponding captions and labels.

7.2 Limitations and Future Work

In this section, I will examine the limitations of the techniques proposed in this thesis, and make recommendations for further research.

The character/symbol recognition method proposed in Chapter 6 goes some distance in real-scene character/symbol recognition. However, this method is only capable to match a real scene symbol or character with its corresponding template or at least a visually similar template, thus presenting the limitation that this method may mis-match symbols with identical identities which have visually diffident appearances. This happens frequently in character recognition. The appearance of a character may vary according to different fonts and styles. Widely different fonts may fail the recognition. For example, we found that Calibri (non-serif) characters can be correctly recognized if templates are in Arial (non-serif), but may not be correctly recognized when templates are in Times New Roman (non-serif). Additionally, different styles (Bold and Italic) also put obstacles for recognition. However, this shortcoming can be overcome by employing more templates. By the clustering based indexing method presented in Section 6.6, the computational burden increased by adding more templates can be relieved. In a nutshell, for character recognition, this method is applicable to scenarios where the perspective deformation is the main recognition obstacle, the number of characters is relatively small, and the font of text are already known. For this method, two important issues that leave open by this thesis is the clustering (multi-component) and occlusion condition. In Chapter 6, we assumed that the correct segmentation of a multi-component character/symbol is known. Therefore, the performance of this recognition method is bounded by the segmentation step. However, because of the flexibility of our method, the segmentation step can be avoided by disassembling a multi-component symbol into components and then recognizing them separately. Another issue is that our method may not able to work when the character/symbol is partially occluded. This issue could be addressed by adapting the global descriptor Cross Ratio Spectrum to a local image patch. Then we will develop techniques for 3-Dimensional object recognition based on the optimized descriptor.

Our word shape coding method presented in Chapter 4 and the holistic word spotting method introduced in Chapter 5 are widely applicable to document images retrieval applications, when a fast speed is desired and the exact word information is not necessary, otherwise OCR is recommended. For these two methods, two issues should be further investigated. The first issue is to quantitatively assess the ambiguity. The ambiguity is an important performance indicator of a word shape coding method or a holistic word spotting method, which indicates the performance boundary of the method for a specific application. This issue has been left open. A quantitative study of the ambiguity of both methods will be done. The second issue is the parameter tuning. Two parameters, namely κ and n, are empirically decided to maximize the performance in this thesis. Adaptive methods to automatically tune both parameters will be included.

For the patent viewing system stated in Chapter 5, two essential parts of this system, namely the word spotting module and the graphics viewing module, have been implemented. Our immediate aim is to integrate these two modules into one system. Further work to enhance these two modules will also be done. In particular, a technique to recognize text content in drawings will be developed. In our current work, only labels and captions are extracted and recognized. However, other text content such as DNA sequences and flow charts, which are also essential for indexing and retrieving drawings, are thrown away. As introduced before, drawing pages of patents suffer from even more severe skew than text pages. Due to the sparse text available, existing de-skewing methods cannot be applied to these drawings. However, our holistic word spotting method is capable of predicating the skew angle at a word level, and thus it is a promising technique to address this problem.

Appendix A

Four Word Shape Coding Methods

In this appendix, four word shape coding schemes are introduced, in an ascending order to the number of symbols used in coding method.

A.1 TAN's method

TAN's method [THS$^+$03] is based on the vertical bar pattern, and 3 codes are employed in total. Vertical bars are extracted by pairing up local minimum and local maximum pixels located on the contour of the word. The mathematic definitions for the local maximum and minimum is as below:

Given an arbitrary curve $f(x)$, and two open intervals on the curve (a, c) and (c, b)

If $f'(x) < 0$ on (a, c), and $f'(x) > 0$ on (c, b) then f has a local minimum at $x = c$.

If $f'(x) > 0$ on (a, c), and $f'(x) < 0$ on (c, b) then f has a local maximum at $x = c$.

The local maximum and minimum pixels are detected by keeping track of the increasing and decreasing trends of the first and the last black pixels in each column. After that, vertical bars are determined by pairing a minimum pixel and a maximum pixel, when the x-distance between them is smaller than a threshold T. Further, vertical bars are classified into three categories depending on whether they have ascenders or descenders. Bars protruding into neither descender nor ascender zones are coded as ‘m’; bars protruding into only descender zones are coded as ‘q’; bars protruding into only ascender zones are coded as ‘b’. As shown in Figure A.1, the word “huge” is converted into vertical bar pattern “dmmmqqm”. Because glitches may appear along a horizontal stroke edge, some edge smoothing is required before coding. Thus there are three symbols used in this scheme.

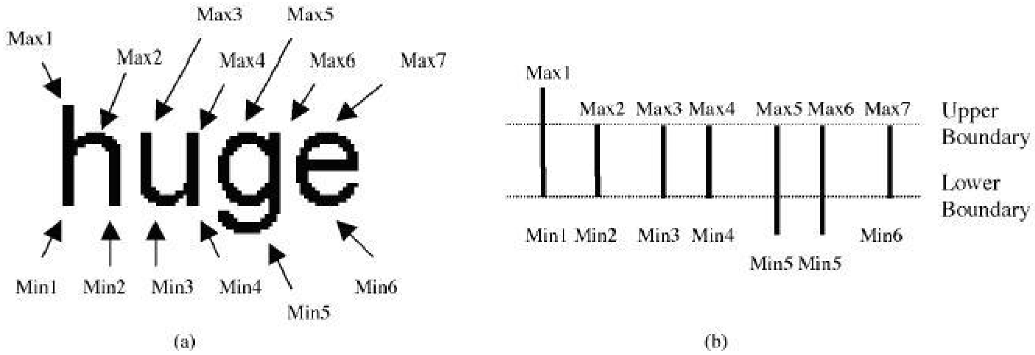


Figure A.1: Extracting the vertical bars from the word “huge” in TAN’s method (the figure is from [THS+03]).

A.2 LU’s method

LU’s method [LLT08] has 5 codes, namely, hole, ascender, descender, leftward water-reservoir (leftward WR), and rightward water-reservoir (rightward WR), encoded as

1, 2, 3, 4, and 5 respectively. For a character, if its code string are unique, the code string will be further encoded as shown in Figure A.2. For example the code string of ‘a’ is ‘43’, which will be further encoded as ‘a’. For document images suffering from various types of document degradation, there may exist a large number of character segmentation errors. In particular, most character segmentation errors result from the serif text font, which causes problems of touching between characters at the x-line or baseline positions of text. For example, adjacent characters “rt”, “rf”, and “rn” are frequently touching at the x-line and baseline position. Touching will produce an undesirable upward or downward water reservoir, but will not generate any leftward or rightward reservoirs. The order of feature codes depends on the location of the feature. Generally, a feature in the left is encoded before one in the right, and a feature in the top is encoded before one in the bottom. For example, the code string of the word “shape” is “sIanoe”. The set of code strings for alphabets in English is shown in Table A.1.

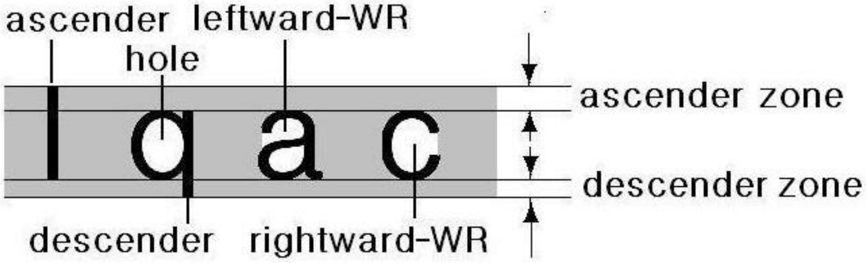


Figure A.2: Features employed in the word shape coding of LU’s (the figure is from [LLT08]).

Table A.1: Codes of 52 Roman Letters and digits by using LU's method.

Characters	Codes	Characters	Codes	Characters	Codes	Characters	Codes
a	a	b	lo	c	c	d	ol
e	e	f	f	g	g	hlIJLT17	l
i	i	j	j	ktK	lc	mnruvw	Nil
p	no	q	on	s	s	o	o
y	y	z	z	A	A	xX	ic
CG	C	DO04	O	E	E	B8	B
HMNUVWY	ll	P	P	Q	Q	F	F
S	S	Z	Z	2	2	R	R
5	5	6	6	9	9	3	3

A.3 SPITZ's method

SPITZ's method [Spi94] has 6 codes in total, and is character based, which means it has a one to one mapping from real English characters to the shape codes. In this method, character cells are firstly detected by connected component analysis. Each character cell is then classified by the presence of features like ascender/descender, the number of components and deep eastward concavity, which is shown in Table A.2. The feature extraction is based on x-line/baseline detection and concavity feature detection.

A.4 LV's method

LV's method [LT04] is stroke-based too, but has a much more complex code set which has 29 codes. Firstly, straight strokes and traversal strokes from the word image are extracted by maximum run-length analysis. Strokes lying in vertical or diagonal

Table A.2: Mapping of character image to shape codes by SPITZ's method.

Shape Code	Characters	Number of Components	Ascender	Descender	Deep Eastward Concavity
A	A-Zbdfhklt	1	YES	NO	NO
x	amnorsuvwxz	1	NO	NO	NO
e	ce	1	NO	NO	YES
g	gpqy	1	NO	YES	NO
i	i	2	YES	NO	NO
j	j	2	YES	NO	NO

directions are considered as straight strokes, and the residues are traversal strokes as shown in Figure A.3.

Each stroke is described by a two-tuple (σ, ϖ) , where σ is based on ascender/descender attribute as shown in Table A.3, and ϖ is based on the shape of the stroke. For the straight stroke, possible codes for ϖ are list below:

'l': vertical straight stroke line, such as that in the characters 'l', 'd'.

'w': left-down diagonal straight stroke line, such as that in the characters 'v', 'w'. 'x': one left-down diagonal straight stroke line crosses one right-down diagonal straight stroke line.

Table A.3: The value of coding for strokes in LV's method.

Shape Code	Ascender	X-line	Descender
x	NO	YES	NO
a	YES	NO	NO
A	YES	YES	NO
D	NO	YES	YES
Q	YES	YES	YES

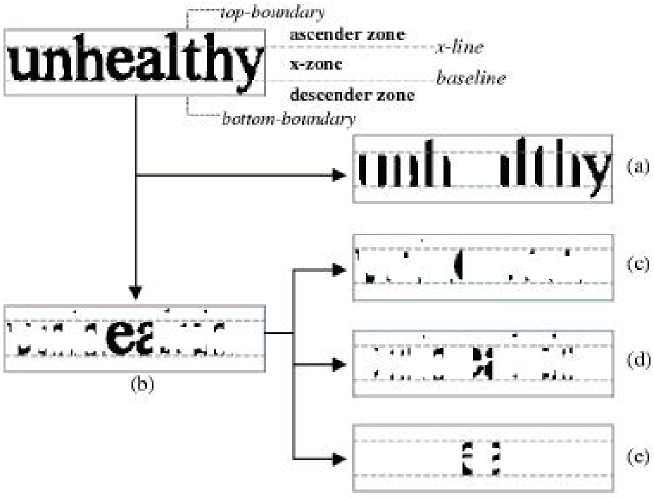


Figure A.3: Primitive string extraction (a) straight line stroke, (b) traversal strokes (c) traversal $TN = 2$, (d) traversal $TN = 4$, (e) traversal $TN = 6$ (the figure is from [LT04]).

'y': one left-down diagonal straight stroke line meets one right-down diagonal straight stroke line at its middle point.

'Y': one left-down diagonal stroke line, one right-down diagonal stroke line and one vertical stroke line cross in one point, like character 'Y'.

'k': one left-down diagonal stroke line, one right-down diagonal stroke line and one vertical stroke line meet in one point, like character 'k'.

For traversal strokes, possible codes for ϖ are based on T_N,namely, the number of transitions from a black pixel to a white pixel or vice versa:

If $T_N = 2$, two parameters are utilized to assign it a feature code. One is the ratio of its black pixel number to x-height, κ . The other is its relative position with respect to the x-line and the base line, $\xi = D_m/D_b$, where D_m is the distance from the topmost stroke pixel in the column to the x-line and D_b is the distance from the bottommost stroke pixel to the baseline.

‘n’: $\kappa < 0.2$ and $\xi < 0.3$

‘u’: $\kappa < 0.2$ and $\xi > 0.3$

‘c’: $\kappa > 0.5$ and $0.5 < \xi > 1.5$

If $T_N > 2$, the feature code is assigned as:

‘o’:$T_N = 4$

‘e’:$T_N = 6$

‘g’:$T_N = 8$

There are only 29 possible combinations of σ and ϖ in English, and hence each of the 29 possible two-tuples (σ, ϖ) is represented by a code. The code string for English characters are shown in Table A.4.

Table A.4: Primitive code strings of characters in LV's method.

Ch	Code String	Ch	Code String
a	(o,x)(e,x)(l,x)	A	(w,A)(v,A)
b	(lA)(o,x)(c,x)	B	(l,A)(e,A)(o,A)
c	(c,x)(o,x)	C	(c.A)(o.A)
d	(c,x)(o,x)(l,A)	D	(l,A)(o,A)(c,A)
e	(c,x)(e,x)(o,x)	E	(l,A)(e,A)
f	(n,x)(l,A)(u,a)	F	(l,A)(o,A)(u,a)
g	(g,D)(e,D)	G	(c,A)(o,A)(e,Q)(o,A)
h	(l,A)(n,x)(l,x)	H	(l,A)(n,x)(l,A)
i	(l,A)	I	(l,A)
j	(l,Q)	J	(u,x)(l,A)
k	(k,x)	K	(k,A)
l	(l,A)	L	(l,A)(u,x)
m	(l,x)(n,x)(l,x)(n,x)(l,x)	M	(l,A)(v,A)(w,A)(l,A)
n	(l,x)(n,x)(l,x)	N	(l,A)(v,A)(l,A)
o	(c,x)(o,x)(c,x)	0	(c,A)(o,A)(c,A)
p	(l,D)(o,x)(c,x)	P	(l,A)(o.A)(c,A)
y	(c,x)(o,x)(cD)	Q	(c,A)(o,A)(e,Q)(o,D)
r	(l,x)(n.x)	R	(l,A)(o,A)(e,A)(o,A)
s	(o,x)(e,x)(o,x)	S	(o.A)(e,A)(o,A)
t	(n,x)(l,A)(o,x)	T	(u,a)(l,A)(u,a)
u	(l, x)(u,x)(l,x)	U	(l,A)(u,x)(l,A)
v	(v,x)(w,x)	V	(v,A)(w,A)
w	(v, x)(w,x)(v,x)(w,x)	W	(v,A)(w,A)(v,A)(w,A)
x	(x,x)	X	(x,A)
y	(y,D)	Y	(Y,A)
z	(z,x)	Z	(z,A)

Bibliography

[AF00] A. Amin and S. Fischer. A document skew detection method using hough transform. *Pattern Analysis and Applications*, 3(3):243–253, 2000.

[BBS05] A. Busch, W.W. Boles, and S. Sridharan. Texture for script identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1720–1732, 2005.

[BMP02] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts full text. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509 – 522, 2002.

[Bob01] M. Bober. MPEG-7 visual shape descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):716 – 719, 2001.

[BSM95] C. Buckley, A. Singhal, and M. Mitra. New retrieval approaches using SMART. In *Proceedings of the 4th Text REtrieval Conference, NIST Special Publication*, pages 25–48, 1995.

[CB93] F.R. Chen and D.S. Bloomberg. Word spotting in scanned images using hidden markov models. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 1–4, 1993.

[CCCC04] S.L. Chang, L.S. Chen, Y.C. Chung, and S.W. Chen. Automatic license plate recognition. *IEEE Transactions on Intellegent Transport System*, 5(1):42–53, 2004.

[CFGS95] P. Comelli, P. Ferragina, M.N. Granieri, and F. Stabile. Optical recognition of motor vehicle license plates. *IEEE Transactions on Vehicular Technology*, 44(4):790–799, 1995.

[CG86] J. Canny and V. Govindraju. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–714, 1986.

[CHTB94] W.B. Croft, S.M. Harding, K. Taghva, and J. Borsack. An evaluation of information retrieval accuracy with simulated OCR output. In *the 3rd Symposium of Document Analysis and Information Retrieval*, pages 115–126, 1994.

[CM04] P. Clark and M. Mirmehdi. Recognizing text in real scenes. *International Journal Document Analysis and Recognition*, 4(4):243–257, 2004.

[CSB01] D. Chen, K. Shearer, and H. Bourlard. Text enhancement with asymmetric filter for video OCR. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 192–197, 2001.

[CSD+88] G. Ciardiello, G. Scafur, M.T. Degrandi, M.R. Spada, and M.P. Roccoteli. An experimental system for office document handling and text recognition. In *Proceedings of the 9th International Conference on Pattern Recognition*, pages 739–743, 1988.

[CWL03] Y. Cao, S. Wang, and H. Li. Skew detection and correction in document images based on straight-line fitting. *Pattern Recognition Letters*, 24(12):1871 – 1879, 2003.

[CY] X. Chen and A.L. Yuille. Detecting and reading text in natural scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[DH73] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* Wiley-Interscience, 1973.

[dlEMSA97] A. de la Escalera, L. Moreno, M. Salichs, and J. Armingol. Road traffic sign detection and classification. *IEEE Transactions on Industrial Electronics*, 44(6), 1997.

[DLL03] D. Doermann, J. Liang, and H. Li. Progress in camera-based document image analysis. In *Proceedings of the 7th International Conference Document Analysis and Recognition*, pages 606–617, 2003.

[Doe98] D. Doermann. The indexing and retrieval of document images : A survey. *Computer Vision and Image Understanding*, 70(3):287–298, 1998.

[DY95] D. Doermann and S. Yao. Generating synthetic data for text analysis systems. *In Symposium on Document Analysis and Information Retrieval*, pages 449–467, 1995.

[ESS+94] S. Estable, J. Schick, F. Stein, R. Janssen, R. Ott, W. Ritter, and Y.-J. Zheng. A real-time traffic sign recognition system. In *Proceedings of 1994 IEEE Intelligent Vehicles Symposium*, pages 213–218, 1994.

[FK88] L.A. Fletcher and R. Kasturi. A robust algorithm for text string separation from mixed text/graphics images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):910–918, 1988.

[FL95] C. Faloutsosand and K. Lin. Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *Proceedings of the 1995 ACM SIGMOD*, pages 163–174, 1995.

[FP03] D. A. Forsyth and J. Ponce. *Computer Vision, a modern approach.* Prentice Hall, 2003.

[GK07] C. Gope and N. Kehtarnavaz. Affine invariant comparison of point-sets using convex hulls and Hausdorff distances. *Pattern Recognition*, 40(1):309–320, 2007.

[GP95] K. Gollmer and C. Posten. Detection of distorted pattern using dynamic time warping algorithm and application for supervision of bioprocesses. In *Preprints of the IFAC Workshop on On-line Fault Detection and Supervision in the Chemical Process Industries*, 1995.

[GTLT95] J. Gao, L. Tang, W. Liu, and Z. Tang. Segmentation and recognition of dimension texts in engineering drawings. In *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 528–531, 1995.

[Gus97] D. Gusfield. *Algorithms on strings, trees, and sequences: computer science and computational biology.* Cambridge University Press, New York, NY, USA, 1997.

[HA96] S. He and N. Abe. A clustering-based approach to the separation of text strings from mixed text/graphics documents. In *Proceedings of the 13th International Conference on Pattern Recognition*, volume 3, pages 706 – 710, 1996.

[HCW97] S.M. Harding, W.B. Croft, and C. Weir. Probabilistic retrieval of ocr degraded text using n-grams. *The 1st European Conference Research and Advanced Technologies for Digital Libraries*, pages 345–359, 1997.

[HHS90] T.K. Ho, J.J. Hull, and S.N. Srihari. A word shape analysis approach to recognition of degraded word images. In *Proceedings of the 4th USPS Advanced Technology Conference*, volume 3, pages 217–231, 1990.

[HHS91] T.K. Ho, J.J. Hull, and S.N. Srihari. Word recognition with multi-level contextual knowledge. In *Proceedings of the 1st International Conference on Document Analysis and Recognition*, volume 1, pages 905–915, 1991.

[HHS92] T. K. Ho, J. J. Hull, and S. N. Srihari. A word shape analysis approach to lexicon based word recognition. *Pattern Recognition Letters*, 13(11):821–826, 1992.

[Hin90] S. C. Hindus. A document skew detection using runlength encoding and the hough transform. In *Proceedings of International Conference on Pattern Recognition*, pages 464–468, 1990.

[HKKT97] J. Hochberg, L. Kerns, P. Kelly, and T. Thomas. Automatic script identification from images using cluster-based templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):176–181, 1997.

[Hul86] J.J. Hull. Hypothesis generation in a computational model for visual word recognition. *IEEE Expert*, 1(3):63–70, 1986.

[ILA95] D.J. Ittner, D.D. Lewis, and D.D. Ahn. Text categorization of low quality images. In *Proceedings of SDAIR-95, the 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 301–315, 1995.

[Ita75] F. Itakura. Minimum prediction residual principle applied to speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 23(1):52–72, 1975.

[JT01] D. Jelinek and C.J. Taylor. Reconstruction of linearly parameterized models from single images with a camera of unknown focal length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(7):767–773, 2001.

[KJM07] A. Kumar, C.V. Jawahar, and R. Manmatha. Efficient search in document image collections. In *Proceedings of the 8th Asian Conference on Computer Vision*, pages 586–595, 2007.

[KK04] E. Keogh and S. Kasetty. On the need for time series data mining benchmarks: a survey and empirical demonstration. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 102 – 111, 2004.

[KP00] E. Keogh and M. Pazzani. Scaling up dynamic time warping for datamining applications. In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 285–289, 2000.

[LCK05] S. Lu, B. M. Chen, and C. C. Ko. Perspective rectification of document images using fuzzy set and morphological operations. *Image and Vision Computing*, 23(5):541–553, 2005.

[LDL05] J. Liang, D. Doermann, and H. Li. Camera-based analysis of text and documents: A survey. *International Journal on Document Analysis and Recognition*, 7(2):83–104, 2005.

[Lin03] X. Lin. Impact of imperfect OCR on part-of-speech tagging. In *Proceedings of the 7th International Conference on Document Analysis and Recognition*, volume 1, pages 284 – 288, 2003.

[LK95] C.M. Lee and A. Kankanhalli. Automatic extraction of characters in complex scene images. *International Journal of Pattern Recognition and Artificial Intelligence*, 9(1):67–82, 1995.

[LL95] M. Lalondeand and Y. Li. Road signs recognition - survey of the state of the art. *Technique Report, CRIM-IIT*, 1995.

[LLT08] S. Lu, L. Li, and C.L. Tan. Document image retrieval through word shape coding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 130(11):1913–1918, 2008.

[Low04] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal Computer Vision*, 2(60):91–110, 2004.

[LPS+03] S.M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young. ICDAR 2003 robust reading competitions. In *Proceedings of the 7th International Conference on Document Analysis and Recognition*, volume 2, pages 682–687, 2003.

[LT04] Y. Lu and C.L. Tan. Information retrieval in document image databases. *IEEE Transactions on Knowledge and Data Engineering*, 16(11):1398–1410, 2004.

[LT06a] S. Lu and C. L. Tan. Camera text recognition based on perspective invariants. In *Proceedings of the 18th International Conference on Pattern Recognition*, volume 2, pages 1042–1045, 2006.

[LT06b] S. Lu and C.L. Tan. Script and language identification in degraded and distorted document images. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 769–774, 2006.

[LT08] S. Lu and C.L. Tan. Script and language identification in noisy and degraded document images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(1):14–24, 2008.

[LTW95] D.X. Le, G.R. Thoma, and H. Wechsler. Classification of binary document images into textual or nontextual data blocks using neural network models. *Machine Vision and Applications*, 8(5):289–304, 1995.

[Luc05] S.M. Lucas. ICDAR 2005 text locating competition results. In *Proceedings of the 8th International Conference on Document Analysis and Recognition*, volume 1, pages 80–84, 2005.

[MBLH05] G.K. Myers, R.C. Bolles, Q.T. Luong, and J.A. Herson. Rectification and recognition of text in 3-D scenes. *International Journal Document Analysis and Recognition*, 7(2-3):147–158, 2005.

[MMS06] S. Marinai, E. Marino, and G. Soda. Font adaptive word indexing of modern printed documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1187 – 1199, 2006.

[MS05] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[MZ92] J.L. Mundy and A.P. Zisserman. *Geometric invariance in computer vision.* MIT Press, 1992.

[Nak94] T. Nakayama. Modeling content identification from document images. In *Proceedings of The 4th Conference on Applied Natural Language*, pages 22–27, 1994.

[NBSK97] N. Nobile, S. Bergler, C.Y. Suen, and S. Khoury. Language identification of on-line documents using word shapes. *Proceedings of the 4th International Conference on Document Analysis and Recognition*, pages 258–262, 1997.

[NNR00] G. Nagy, T.A. Nartker, and S.V. Rice. Optical character recognition: an illustrated guide to the frontier. In *Proceedings of SPIE: Document Recognition and Retrieval VII*, volume 3967, pages 58–69, 2000.

[O'G93] L. O'Gorman. The document spectrum for page layout analysis. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 15(11):1162–1173, 1993.

[OH04] C. Orrite and J.E. Herrero. Shape matching of partially occluded curves invariant under projective transformation. *Computer Vision and Image Understanding*, 93(1):34–64, 2004.

[OTA97] M. Ohta, A. Takasu, and J. Adachi. Retrieval methods for English text with misrecognized OCR characters. In *Proceedings of the 4th International Conference on Document Analysis and Recognition*, pages 950–956, 1997.

[Pil01] M. Pilu. Extraction of illusory linear clues in perspectively skewed documents. In *Proceedings of IEEE on Computer Vision and Pattern Recognition*, volume 1, pages 363–368, 2001.

[Pos86] W. Postl. Detection of linear oblique structure and skew scan in digitized documents. In *Proceedings of International Conference on Pattern Recognition*, pages 687–689, 1986.

[PP02] M. Pilu and S. Pollard. A light-weight text image processing method for handheld embedded cameras. In *Proceedings of British Machine Vision Conference*, pages 547–556, 2002.

[RM03] T.M. Rath and R. Manmatha. Word image matching using dynamic time warping. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 2, pages 521–527, 2003.

[RML04] T.M. Rath, R. Manmatha, and V. Lavrenko. A search engine for historical manuscript images. In *Proceedings of ACM SIGIR Conference Research and Development in Information Retrieval*, pages 369–376, 2004.

[RZFM95] C.A. Rothwell, A. Zisserman, D.A. Forsyth, and J.L. Mundy. Planar object recognition using projective shape representation. *International Journal on Computer Vision*, 16(1):57–99, 1995.

[SC78] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):59–165, 1978.

[SC07] S. Salvador and P. Chan. Toward accurate dynamic time wrapping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.

[SF04] T. Suk and J. Flusser. Projective moment invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10), 2004.

[SG89] S. N. Srihari and V. Govindraju. Analysis of textual images using hough transform. *Machine vision and applications*, 2(3):141–153, 1989.

[SIR99] T. Steiherz, N. Intrator, , and E. Rivlin. Skew detection via principal component analysis. In *Proceedings of the 5th International Conference on Document Analysis and Recognition*, pages 153–156, 1999.

[Spi94] A.L. Spitz. Using character shape codes for word spotting in document images. In *Proceedings of the 3rd International Workshop on Syntactic and Structural Pattern Recognition*, 1994.

[Spi97] A.L. Spitz. Determination of the script and language content of document images. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, 19(3):235–245, 1997.

[SS97] A.F. Smeaton and A.L. Spitz. Using character shape coding for information retrieval. In *Proceeding of the 4th International Conference Document Analysis and Recognition*, pages 974–978, 1997.

[Tak97] A. Takasu. An approximate string match for garbled text with various accuracy. In *Proceedings of the 4th International Conference on Document Analysis and Recognition*, volume 2, pages 957–961, 1997.

[TBC94] K. Taghva, J. Borsack, and A. Condit. Results of applying probabilistic IR to OCR text. In *Proceedings of the 7th ACM SIGIR Internationa Conference on Retrieval*, pages 202–211, 1994.

[TBC96] K. Taghva, J. Borsack, and A. Condit. Evaluation of model-based retrieval effectiveness with OCR text. *ACM Transactions on Information Systems*, 14(1):64–93, 1996.

[THS+03] C.L. Tan, W. Huang, S.Y. Sung, Z. Yu, and Y. Xu. Text retrieval from document images based on word shape analysis. *Applied Intelligence*, 18(3):257–270, 2003.

[TLH99] C.L. Tan, P.Y. Leong, and S. He. Language identification in multilingual documents. *International Symposium on Intelligent Multimedia and Distance Education*, pages 59–64, 1999.

[TNB+01a] K. Taghva, T. Nartker, J. Borsack, S. Lumos, A. Condit, and R. Young. Evaluating text categorization in the presence of ocr errors. In *Internatial Symposium on Electronic Imaging Science and Technology*, volume 4307, pages 68–74, 2001.

[TNB01b] K. Taghva, T. A. Nartker, and J. Borsack. Recognize, categorize, and retrieve. In *Proceedings of the Symposium on Document Image Understanding Technology*, pages 227–232, 2001.

[TTP+02] K. Tombre, S. Tabbone, L. Pssier, B. Lamiroy, and P. Dosch. Text/graphics separation revisited. In *Proceedings of the 5th International Workshop on Document Analysis Systems*, pages 200 – 211, 2002.

[Vin05] A. Vinciarelli. Noisy text categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1882–1895, 2005.

[WG97] K. Wang and T. Gasser. Alignment of curves by dynamic time warping. *The Annals of Statistics*, 25(3):1251–1276, 1997.

[WOKT98] Y. Watanabe, Y. Okada, Y. B. Kim, and T. Takeda. Translation camera. In *Proceedings of the 14th International Conference on Pattern Recognition*, pages 613–617, 1998.

[WZH00] W.J. Williams, E. Zalubas, and A.O. Hero. Word spotting in bitmapped fax documents. *Information Retrieval*, 2:207–226, May 2000.

[XL07] D. Xu and H. Li. 3-D projective moment invariants. *The Journal of Information and Computational Science*, 4(1), 2007.

[Yan93] H. Yan. Skew correction of document images using inerline cross-correlation. *Computer Vision Graphics Image Processing*, 55(6):538–543, 1993.

[YGZ$^+$01] J. Yang, J. Gao, Y. Zhang, X. Chen, and A. Waibel. An automatic sign recognition and translation system. In *Proceedings of Workshop on Perceptive User Interfaces*, pages 1–8, 2001.

[YJ96] B. Yu and A. K. Jain. A robust and fast skew detection algorithm for generic documents. *Pattern Recognition*, 29(10):599–1630, 1996.

[YJF98] B.K. Yi, H.V. Jagadishand, and C. Faloutsos. Efficient retrieval of similar time sequences under time warping. In *Proceedings of 14th International Conference on Data Engineering*, pages 201–208, 1998.

[YMMN05] T. Yamaguchi, M. Maruyama, H. Miyao, and Y. Nakano. Digit recognition in a natural scene with skew and slant normalization. *International Journal of Document Analysis and Recognition*, 7(2-3):168–177, 2005.

[YNF90] H. Fujisawa J. Higashino Y. Nakano, Y. Shima and M. Fujinawa. An algorithm for skew normalization of document images. In *Proceedings of the 10th International Conference on Pattern Recognition*, pages 8–13, 1990.

[YT00] Z. Yu and C.L. Tan. Image-based document vectors for text retrieval. In *Proceedings of the 5th International Conference on Pattern Recognition*, volume 4, pages 393–396, 2000.

[ZL04] D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37(1):1–19, 2004.

[ZTF04] Z. Zhang, C.L. Tan, and L. Fan. Restoration of curved document images through 3D shape modeling. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pages 10–15, 2004.

[ZYT07] L. Zhang, A.M. Yip, and C.L. Tan. A restoration framework for correcting photometric and geometeric distortions in camera-based document images. In *Proceedings of International Conference on Computer Vision*, pages 1–8, 2007.

Publications

- Linlin Li and Chew Lim Tan, Recognizing planar symbols with severe perspective deformation, IEEE Transactions on Pattern Analysis and Machine Intelligence, to appear.

- Peng Zhou, Linlin Li and Chew Lim Tan, Character recognition under severe perspective distortion, 10th International Conference on Document Analysis and Recognition, ICDAR 2009

- Shuyong Bai, Linlin Li and Chew Lim Tan, Keyword spotting in document images through word shape coding, 10th International Conference on Document Analysis and Recognition, ICDAR 2009

- Linlin Li and Chew Lim Tan, Character Recognition under Severe Perspective Distortion, 19th International Conference on Pattern Recognition, ICRP 2008.

- Linlin Li and Chew Lim Tan, Script Identification of Camera-based Images, 19th International Conference on Pattern Recognition, ICPR 2008.

- Linlin Li and Chew Lim Tan, A graphics image processing system, 8th IAPR International Workshop on Document Analysis Systems, On page(s): 455-462, DAS 2008.

- Linlin Li, Shijian Lu, and Chew Lim Tan A Figure Image Processing System. Graphics Recognition Lecture Notes in Computer Science, Volume: 5046, On page(s): 191-201, 2008.

- Shijian Lu, Linlin Li and Chew Lim Tan, Document image retrieval through word shape coding, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume: 30, Issue: 11, On page(s):1913-1918, 2008.

- Linlin Li and Chew Lim Tan, A word shape coding method for camera-based document images, 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, On page(s): 771-772, 2008.

- Linlin Li, Shijian Lu and Chew Lim Tan, A Fast Keyword-Spotting Technique, International Conference on Document Analysis and Recognition, Volume: 1, On page(s):68-72, ICDAR 2007.

- Shijian Lu, Linlin Li and Chew Lim Tan, Identification of Latin-base Languages through Character Stroke Categorization, 9th International Conference on Document Analysis and Recognition, Volume: 1, On page(s): 352-356, ICDAR 2007.

- Linlin Li and Chew Lim Tan, Improving OCR text categorization accuracy with electronic abstracts, 2nd International Conference on Document Image Analysis for Libraries, DIAL 2006.