INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

MODEL-BASED IDENTIFICATION OF ORIENTAL DOCUMENTS

RITA A. YACOUB SAID

A THESIS
IN
THE DEPARTMENT
OF
COMPUTER SCIENCE

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF COMPUTER SCIENCE AT
CONCORDIA UNIVERSITY
MONTREAL, QUEBEC, CANADA

SEPTEMBER 1999

Abstract

Model-Based Identification of Oriental Documents

Rita A. Yacoub Said

Computers with the capability of identifying languages printed in documents can support many potential applications including document classification for character recognition, translation, and language understanding. Language identification is normally done manually. However, the high volume and variety of languages encountered make manual identification impractical and an automatic language approach becomes necessary. Therefore, language identification is a key step in the automatic processing of document images.

This thesis is concerned with a model-based classification of Oriental documents into Chinese, Japanese, and Korean. A model-based approach locates an object, of which the computer has a model, in an image. In this work, the objects to be located are some of the most frequently appearing characters in each of the three Oriental languages, and the images to be searched for the objects are the Oriental documents fed to the system.

A major part of the work is to locate instances of the character models in an Oriental document, which is done by using the Hausdorff distance, a similarity measure defined between two sets of points. One of the point sets represents a model of some Oriental character to look for, and the other represents each character in the document image to be identified. Since Oriental documents are complex in structure, a portion of the text is extracted from the input document for further processing.

The block of text extracted from the document to be classified is subject to some preprocessing, namely skew correction and segmentation. Our method of classification then tries to match each character in the extracted block of text with the several most frequently used character models, starting with Korean, followed by Japanese, and ending with Chinese. This search order is crucial because of the major overlap of

many Chinese characters in the Japanese and Korean languages. To prevent Korean and Japanese documents from being misclassified as Chinese, the Chinese models are applied at the end.

The wide variety of text fonts adds to the complexity of the identification problem because our method is based on shape resemblance between the models and the document's characters. Our system is trained on 2 Chinese, 1 Japanese, and 2 Korean commonly used fonts. The system can be made to handle other fonts that differ markedly from those to which it is trained by adding the new fonts to the training set, thus increasing the reliability of the system.

A document is rejected by the system in case its text orientation could not be detected or no matches could be found in it, both in regular and in special fonts.

When tested on Hamanaka's database which contains 391 document images, the overall classification rate is 93.35% with a reliability of 98.92%. On Ding's database with 448 document images, the system achieved an overall classification rate of 98.21% with a 100.00% reliability. It is note worthy to mention that both databases were created at the Centre for Pattern Recognition and Machine Intelligence, CENPARMI. The proposed model-based classification approach proved to be effective for the identification of document images stored electronically.

Acknowledgements

I would like to express my sincere gratitude and appreciation to both of my supervisors Drs. C. Y. Suen and K. Liu for their invaluable guidance, assistance, and care during this work. This thesis would have been impossible without their encouragement, helpful discussions, and financial support from their research grants.

I am also very grateful to Nicholas Strathy whose code libraries were used in my project, Boulos Waked for sharing his deskewing algorithm, Christine Nadal for making the databases available, Stan Swiercz, Michael Assels, William Wong, and Mike Yu for their technical support.

Special thanks go to Masahiko Hamanaka and Jie Ding for constructing the valuable Oriental databases used in this project, Il-Seok Oh and Jinho Kim for sharing some information on the Korean language.

I wish to thank *all* the people at CENPARMI who helped me in one way or another since I first joined the centre.

Last but not least, I am indebted to my unique husband Fady, our great parents Antoine and Hiam, Nassif and Salma, our lovely sisters Tina and Sandra, our exceptional brothers Samy, Sameer, and Joseph (Big Joe) for their unbounded understanding and love.

Dedication

To Fady, my guiding light.

This Thesis
is also dedicated to

our *loving* parents.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The Challenge

Language is one of the most important means of human communication [SMR98]. The hundreds of languages that are in common use worldwide today are the products of long years of evolution. It is difficult to identify the language of a document if it is not printed in one we are familiar with. Hence teaching the computer to deduce and understand different languages is a big challenge. In case computers are capable of reading and identifying the languages printed in documents, several potential operations would be possible including document classification, language understanding, information retrieval, document sorting in support of character recognition, and translation. This topic is referred to as language *identification* or *differentiation* in the pattern recognition field.

The capability of recognizing multilingual documents by computers is a novel approach [Spi97]. In the past, language identification was done manually by experts due to the small amount of documents to be processed. However, the high volume and variety of documents stored electronically in image form makes manual identification impractical and developing an automated language identification system becomes a necessity. For instance, the early detection of the language present in a document has implications in the selection of the proper character recognition service [Spi97], the fact that will considerably facilitate further processing. Furthermore, a reliable

system would considerably help librarians and others who work with multilingual documents but do not know the language of the documents they deal with. Other automated systems such as information retrieval would also benefit [SR96]. Hence language identification has become a key step in the automatic processing of document images.

1.2 Research Objective

Enabling computers to automatically process information found in printed Oriental documents in an attempt to classify them into Chinese, Japanese, or Korean is the objective of this research. To meet this objective, the following major steps had to be carried out:

1. Find and rescale the most frequently used characters in each language to make templates for Chinese, Japanese, and Korean.
2. Manually extract a block of text from a new image, and then the computer takes over to segment it to character cells and rescale each character to the templates' size.
3. Compare each symbol of the new document to each language's templates.
4. Choose the language whose templates provide the best match.

1.3 Previous Work on Language Identification

Research on language identification is recent compared to optical character recognition. The two current major areas of study are:

1. Script or category classification: there are several families into which different languages are typeset, such as Roman, Arabic, Oriental, *etc.* Identifying the language family of a document is the concern of category classification. Related work can be found in [HKK95], [Spi97], [DLS97], [SBN98], *etc.*

2. Language identification: which consists of classifying a document within a certain language category. For instance, one could classify an Oriental document into Chinese, Japanese, or Korean specifically [Spi97], [SH98], [DLS97]. Work related to Roman language differentiation includes [Spi94], [SS94], *etc.*

The various techniques that have been developed for the identification of languages are mainly based on two approaches [DLS97]: searching for specific tokens in different languages ([HKK95]) and using statistical information ([DLS97], [Spi94], [SH98]).

Statistics-based approaches first extract some features like optical density ([Spi94], [LNB96]), upward and downward concavities ([LNB96]), horizontal projection profiles as well as the distribution of connected components and their sizes ([DLS97]), then the language of the document is identified based on the values of these features.

On the other hand, token (template) matching approaches determine a set of language specific tokens or models for each language and searches for such models in the document and then finds the best match.

Among the numerous families into which documents can be grouped, Oriental languages are the most spoken in the world, [SMR98]. Of these and of special interest to this work are the Chinese, Japanese, and the Korean languages. Therefore, a review of related work on Oriental language identification, based on both the statistical and the template matching approaches, will be presented in sections 1.3.1 and 1.3.2, respectively.

Most of the language differentiation work to date has been performed on Roman languages. It is only lately that researchers have started to study the differentiation between Chinese, Japanese, and Korean. In comparison with Roman language differentiation, fewer methods have been proposed for Oriental scripts due to their complexities and their very large character sets.

1.3.1 Related Work on Statistics-Based Techniques

In earlier work [Spi94], Spitz proposed a method to differentiate between Chinese, Japanese, and Korean using optical density. In this method, a document image is first segmented into character cells and within each such cell, the optical density of a character cell is calculated. The optical density $D1_i$ of the i-th character cell is defined as the number of 'on' pixels over its area, represented as follows:

$$D1_i = \frac{B_i}{H_{L(i)} W_i} \tag{1}$$

where B_i is the number of black pixels in the i-th cell, W_i is the width of the i-th cell, and $H_{L(i)}$ is the height of the line to which the i-th cell belongs. Spitz showed that the histograms of the optical density corresponding to Chinese, Japanese, and Korean exhibit remarkably different distributions. The optical density function for Korean documents shows a distinct bimodal nature, with the low density mode smaller than the high density mode. The distribution of the Japanese documents is also characterized as bimodal, but in this instance, the lower density mode is greater than the high density one. In Chinese documents, there is only one significant mode. The three languages are classified by applying a linear discriminant analysis (LDA) to the histograms.

Spitz's optical density is dependent on the language to which the character belongs and is also sensitive to the font and the printing style [Din99], and is influenced by stroke width, as pointed out in [LNB96] and [SH98]. Density features depend on character segmentation, therefore the characters should be segmented almost successfully, because complexities in Oriental languages change considerably from character to character.

In [SH98], Suen *et al.* mentioned that while optical densities are influenced by stroke width, those extracted from thinned images are insensitive to stroke width. However, since thinning is usually time consuming, the authors proposed contour density of the original image, which is calculated as the fraction of its area that is

on contour, and can be used instead of the optical density of a thinned image. The contour density $D2_i$ of the i-th cell is defined as:

$$D2_i = \frac{C_i}{H_{L(i)} W_i} \tag{2}$$

where C_i is the number of contour pixels in the i-th cell.

The method that the authors proposed for the identification of Oriental languages is based on density (complexity) and curvature features, which they estimate to be effective for identification. Many curvature extraction methods for digital images have been proposed [WS93]. The curvature extraction method that the authors of [SH98] have used is the k-curvature defined by Rosenfeld and Kak [RK82], where the k-curvature is based on the derivative of tangent orientation. By using k-curvature, straight lines and curves can be detected. Curvatures has the merit that a character segmentation is not required, while it is needed to extract density. However, curvature is dependent on character sizes, therefore characters have to be normalized to the same height. Through their experiments, Suen *et al.* show that Japanese has distributions a little different from Chinese and Korean. However, it seems to be difficult to identify all three Oriental languages using only density features. In addition, using Rosenfeld's k-curvature, it has been shown that Korean can be identified by detecting straight lines and circles. However, curvature distributions for Chinese and Japanese are very similar to each other. The authors consider that Oriental languages can be identified using both the density and the curvature features. This method alleviates the problem with documents in multiple fonts. In the experiments, 6 Chinese, 6 Japanese, and 6 Korean sample document images were used, each document consisting of 200 characters.

In another work by Ding [Din99] and [DLS97], the author proposed three new features that are effective in discriminating documents in Chinese, Japanese, and Korean; these features are complex structure (C), Korean "circles" (K), and long vertical stroke (V). A character cell is said to have a complex structure if it has at least one

loop containing other components, or this loop contains more than one inner contour. Loops are extracted by applying the contour tracing algorithm [Str93], which determines the connected components and stores them in a representation which reflects their topological relationships. The relationships captured include the order of occurrence of the leftmost pixel of the uppermost part of each contour as the image is scanned row by row, and the nesting of contours within others. Korean circles and ellipses are contained in many Korean characters. Digitized circles and ellipses are not perfect in geometry, because they are very similar to squares, rectangles and other loops, which necessitated a search for methods to represent the difference between Korean circles, ellipses, and other loops based on more than their shape alone, such as size and location filtering. Korean vertical strokes is designed to target a group of frequently used Korean characters which contain isolated vertical strokes as tall as the Korean characters themselves. From the experimental results, the author concluded that complex structure is important for separating Japanese texts from Chinese. Furthermore, long vertical stroke is important for differentiating Korean from Chinese and Japanese, while Korean circle/ellipse is necessary for the separation of Korean from Chinese and Japanese texts.

Each processed document image should contain at least 200 characters, otherwise the document is considered not to contain enough information to be identified. Ding experimentally set the following to classify documents of the three languages:

1. If $K \geq 9$ and $V \geq 13$, then it is identified as Korean; otherwise
2. If $C \geq 23$, then it is classified as Chinese; if $C \leq 21$, it is Japanese; otherwise, reject.

Applying this rule to 114 Chinese documents, the recognition rate of the system was 94.69% , the error rate was 4.43%, and the rejection rate was 0.88% with one non-processed document. For 49 Japanese documents, the recognition rate was 95.92%, the error rate was 0.00% with a rejection rate of 4.08%. For 106 Korean documents, the recognition rate was 93.33%, the error rate was 6.67%, and the rejection rate was 0.00% with one non-processed document. Some Chinese documents were misclassified

as Japanese due to the existence of fewer complex structures in the font in which the Chinese documents are printed. The Korean samples misclassified as Japanese are due to problems with the Korean circle detection.

In an effort to improve classification results, Ding also used the K-means clustering algorithm. For 114 Chinese documents, the recognition rate was 94.69%, the error rate was 4.43%, and the rejection rate was 0.88%, with one non-processed document, same results as before. For 49 Japanese, the recognition increased to 97.96%, the error rate was 0.00%, and the rejection rate decreased to 2.04%. For 106 Korean, Ding achieved a recognition rate of 97.14%, an error rate of 1.91%, and a rejection rate of 0.95%, with one non-processed document. We clearly notice an improvement in Japanese and Korean rates, with Korean benefiting the most.

1.3.2 Related Work on Template Matching Techniques

The literature on the automatic Asian script identification using template matching approaches is very scarce. Most of the work done to date tackles the Oriental script identification process statistically. The token matching techniques have been mostly applied to European scripts like French, English, German, *etc* such as [NS93], [SS94], and [NBS97] rather than Asian scripts like Chinese, Japanese, and Korean. The template-based method of [HKK95] described in the current sub-section is concerned with many languages including Asian ones. However, the authors presented the overall results of their system and thus its performance purely on Oriental scripts cannot be estimated.

In [HKK95], Hochberg *et al.* described a system that automatically identifies the script of a document stored electronically in image format. The essence of the approach is for the script identifier to discover a set of representative symbols or *templates* in each script by means of cluster analysis, and then look for instances of them in new documents. The cluster analysis identifies textual symbols, clusters them, and calculates each cluster's centroid. This serves as a representative symbol for the cluster. Clusters with one or two members were eliminated to focus on the

templates that were most likely to be useful. To identify the script used in a new language, the authors compare a subset of its symbols to the templates for each script, and choose the script whose templates provide the best match. The author's classification algorithm accepted the number of symbols to examine and a reliability threshold. The system was tested on 65 test images and 68 challenge images. The test set contained images drawn from the same sources as the training set. The challenge set consisted of images not used in the training set including novel fonts and languages. Among other scripts, there were 10 images of each Asian script in the training set, 5 images of each in the testing set, and 7 Chinese, 3 Japanese, and 6 Korean in the challenge set. With the number of symbols over 75 and the reliability threshold over 50%, all 65 test images were correctly classified, and out of the 68 challenge images, two or three images were misclassified. The documents misclassified by the system are those printed in fonts that differ considerably from those in the training set. One should be able to solve this problem by augmenting the training set.

1.4 Organization of the Thesis

This chapter introduced the challenge of identifying scripts by computers and the reason an automated language identifier is necessary. It also outlined the potential uses of an automated language identification system. In addition, it presented some previous work on language identification of Asian languages. Statistics-based as well as template-matching techniques were explored.

Chapter 2, entitled "*Identification of Chinese, Japanese, and Korean*", will first shed light on the characteristics of the three Oriental languages and then will explain the preprocessing tasks to be undertaken before the system starts the identification process. Such tasks include deskewing, text orientation detection and segmentation. Most importantly, it will present this work's method of identification which deals with documents printed in various fonts.

Chapter 3, entitled "*Experimental Results*", will present an analysis of the experimental results in order to demonstrate the performance of the system tested on 2 independent datasets, the first created by Hamanaka and the second by Ding.

Chapter 4, entitled "*Conclusion and Future Directions*", will state the major contribution of this thesis and future research directions.

Chapter 2

Identification of Chinese, Japanese, and Korean

2.1 Aspects of Oriental Languages

The identification of Chinese, Japanese, and Korean is a challenging task due to their very large character sets and the complexity of their structure. Although Chinese, Japanese, and Korean have individual characteristics that distinguish them from each other, they all have a common structure of pictogram elements. They give people the impression of a picture. They are inter-related either through a common set of characters, or through usage of loaned words or meanings. Both Japanese and Korean have adapted many of the Chinese characters or expressions. For instance, some Chinese characters constitute a large proportion of Japanese texts, and some Chinese words form the vocabulary of an equally large proportion of Korean expressions [SMR98]. This large overlap of Chinese characters adds to their complexity. Prior to introducing the identification of these three Oriental languages, let's explore some of their unique aspects.

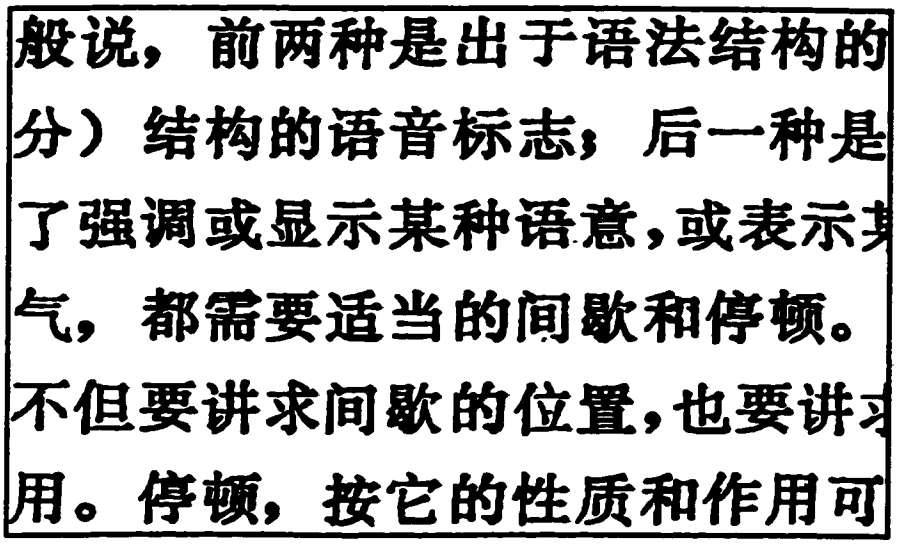

Figure 1: A sample Chinese (simplified) document image printed horizontally.

2.1.1 The Chinese Language

Chinese has a long history of more than 4,500 years. It is being used in many places like China, Hong Kong, Taiwan, Singapore, and Malaysia. Chinese characters are traditionally written from top to bottom in a vertical line that shifts from right to left, but nowadays most texts are written from left to right on a horizontal line. They are the only pure ideograms in the present world [Nak80] and have a very unique writing system due to the construction of its characters. Each character is formed by a certain number of strokes, usually less than 12 in number [Din99], and some characters are so complicated that they have more than 30 strokes [Wan88]. Each Chinese character is confined in a rectangle or square area known as the "square word". The other property of the Chinese language is its large vocabulary. It is believed that there are about 55,000 Chinese characters, while only about 3,000 to 4,000 are used daily. Nowadays, simplified Chinese characters are used in China, while the traditional form is still used in Hong Kong and Taiwan [SH98]. Figure 1 show an example of a horizontal Chinese document while Figure 2 illustrates a vertical one.

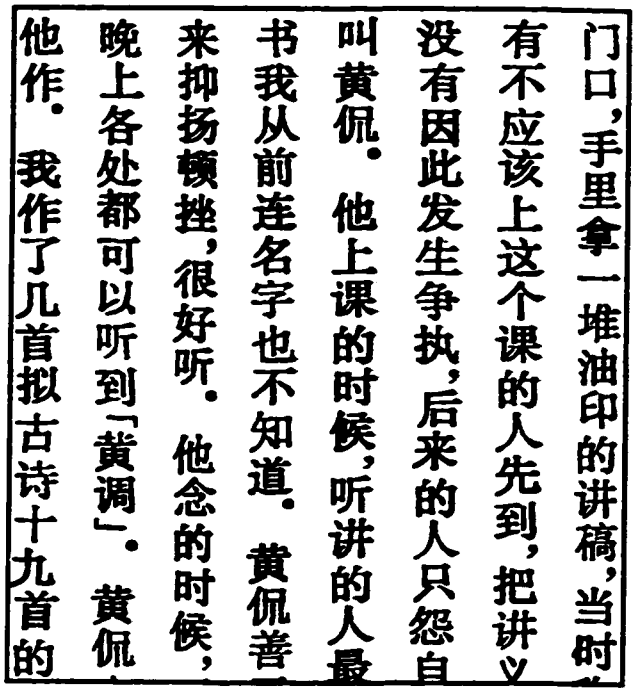

Figure 2: A sample Chinese (simplified) document image printed vertically.

2.1.2 The Japanese Language

Japanese imported Chinese characters from China over a thousand years ago. Around 4,000 Chinese characters, called *Kanji*, are used daily in Japan. Contemporary Japanese is written in Kanji and two sets of Japanese Kanas, namely *Hiragana* and *Katakana* which are phonetic symbols invented by the Japanese to represent particles and other grammatical devices which have no exact equivalents in Chinese [SMR98]. According to [SH98], Kana characters are pure syllabaries, obtained by simplifying some Kanji characters; their average number of stroke is 2 or 3. Hiragana is a cursive script which is used together with Kanji, while Katakana is an angular script which is used for loaned words and emphasis. Each set contains about 80 characters. 50% of usual Japanese texts is occupied by Hiragana and about 30 to 40% is by Kanji. Like Chinese, the orientation of Japanese text is either horizontal as shown in Figure 3 or vertical as depicted in Figure 4.

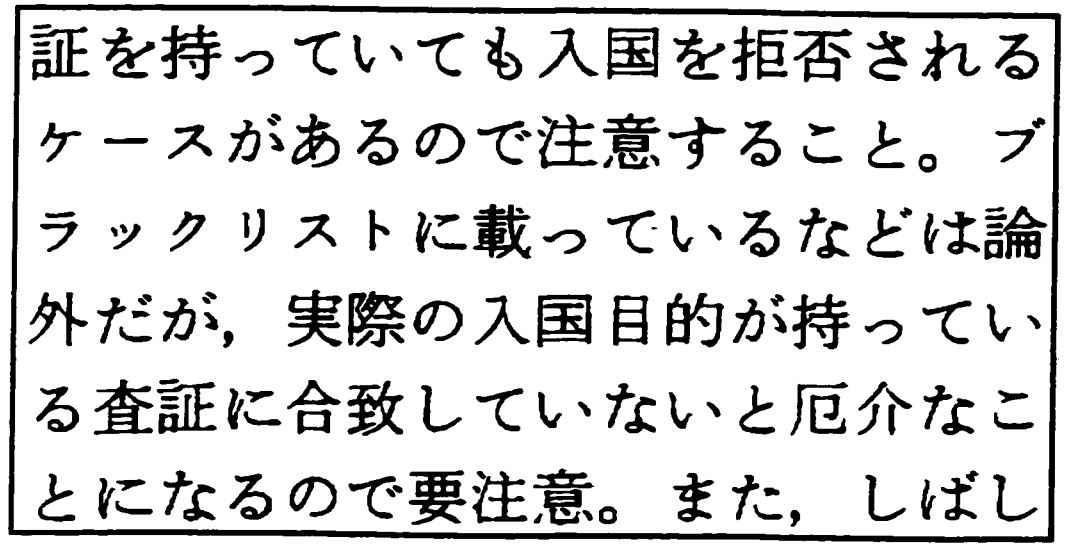

Figure 3: A sample Japanese document image printed horizontally.

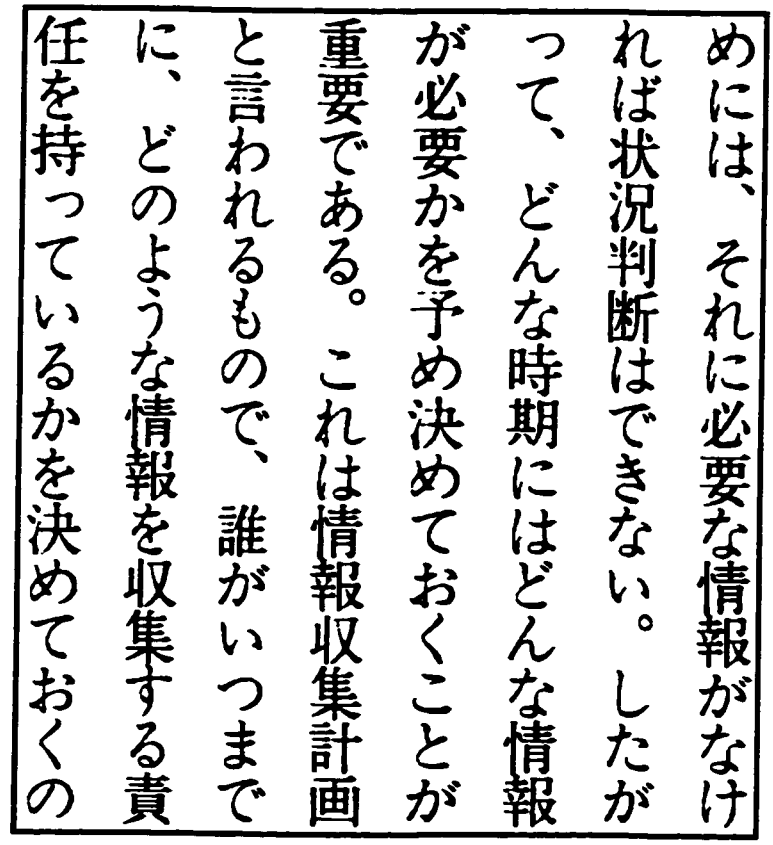

Figure 4: A sample Japanese document image printed vertically.

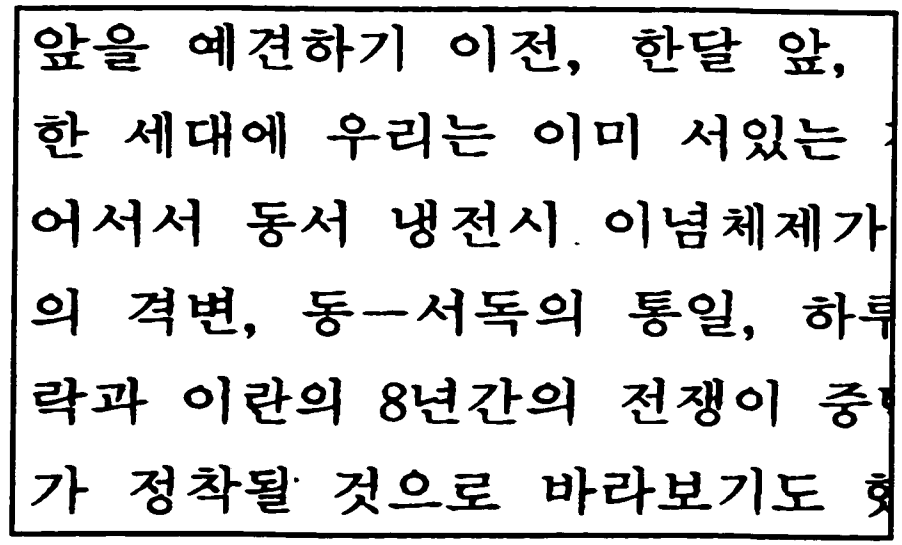

Figure 5: A sample Korean document image printed horizontally.

2.1.3 The Korean Language

Up to the 15th century, there was no distinct Korean alphabet, and Koreans used Chinese ideographs to express their language in writing, sometimes to represent the ideograph's original meaning and sometimes to simply express sounds. However, learning complex Chinese characters was a difficulty for the common people. Therefore, King Sejong the Great commissioned a team of scholars to devise a new simple method of writing down spoken Korean. As a result, *Hangul* was invented in 1443 [SMR98]. Hangul is a syllabic system characterized by its two dimensional composition of three graphemes called the first sound (consonant), the middle sound (vowel), and the optional last sound (consonant). The number of graphemes which belong to the first sound is 19, that to the middle sound is 21, and that to the last is 27. The number of possible Korean characters that are generated this way is very large, a total of 11,172 characters but only 2,350 of them are enough for daily use [LKB96]. There are many character shapes that look alike, which makes Korean identification very difficult. Hangul is used throughout Korea though Chinese characters are still used to represent original Chinese loaned words. Like Chinese and Japanese, Korean text can be printed horizontally or vertically as illustrated in Figures 5 and 6, respectively.

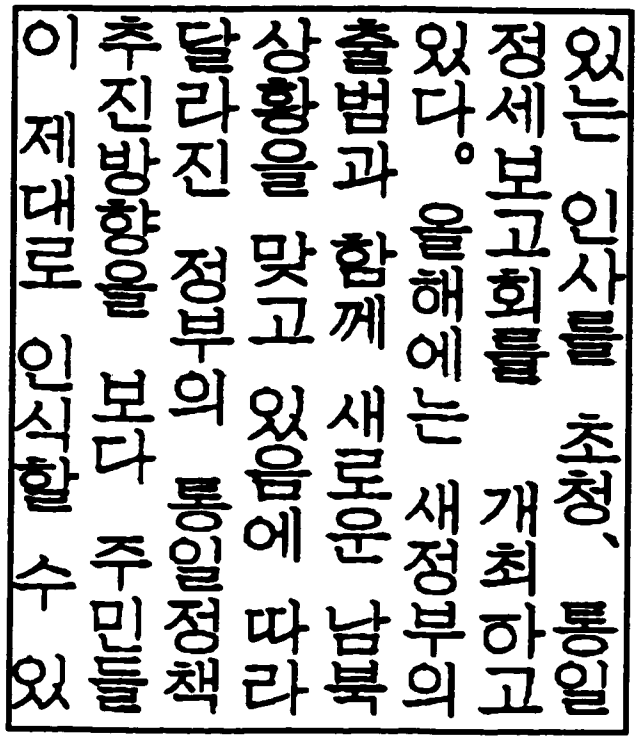

Figure 6: A sample Korean document image printed vertically.

2.2 Preprocessing

2.2.1 Skew Correction

Due to scanning for example, the lines of a document image are not always straight. Since our segmentation method relies on horizontal and vertical projection profiles, the document image is deskewed prior to segmentation in case a skew angle was detected. The skew correction algorithm used by this work is presented in [WBS98] where Waked *et al.* used the Hough Transform [DH75], which is commonly used for detecting lines in an image. The idea is to transform points in an image space xy to a new space domain $\rho\theta$, using the transform equation $\rho_i = xcos\theta_i + ysin\theta_i$. The authors then search for peaks which correspond to lines in the image. The peak is formed when the transformed points lie along a given line in the image. The angle of each line can be found from the coordinates ρ and θ of the peak. Because the transform equation is time-consuming, not all the pixels are transformed. Only two points from

each box of connected components are transformed: the bottom left and right corners.

Another important consideration is to construct the Hough domain by quantifying the Hough space along ρ and θ. θ is chosen to be between $-90°$ and $+90°$, and the resolution angle is set to 0.5°. Along the ρ axis, ρ_i ranges between $-\rho_{max}$ and $+\rho_{max}$, where $\rho_{max} = (h^2 + w^2)/R$, h and w are the height and the width of the image, and R is the resolution, set to 1/5 of the average box height [FK88]. After the Hough space is defined, the transformation equation is used where each time a sinusoidal curve intersects another at (ρ_i,θ_i), the value is incremented at this location. Then, the maximal peaks are located within a window of 8x8 cells, and the maximum is detected within a neighborhood surrounding each maximal peak. The skew angle is then searched for in the region where angles occur most frequently, namely $-25°$ and $+25°$. Therefore, the skew angle is considered as the most frequently occurring local maximum in this range. Figures 7 and 9 both show a skewed Korean document with a skew angle of $-1.5°$ and 0.5°, respectively. Figures 8 and 10 show their respective deskewed image.

2.2.2 Detection of Text Orientation

The deskewing algorithm results in an image with zero skew angle. Next, we have to determine the text orientation of the deskewed image, i.e., whether the image is printed horizontally or vertically. Text orientation detection is a prerequisite for our segmentation algorithm, as will be explained in sub-section 2.2.3.

The process of determining the text orientation of an image is based on the following observations:

- The horizontal projection profile of a horizontal image displays clear white-runs between the histograms which reflect *inter-line* gaps as shown in Figure 12, which displays the horizontal projection of the horizontal image in Figure 11. However, the horizontal projection of a vertical image displays very few gaps, if any at all. Figure 14 shows the horizontal projection of the vertical image of

우리는 이미 여러차례 있었던 중동 전쟁을 통하여 지난 일을 찾으시는 하나님의 인간 역사의 운행을 생각해 볼 수 있다. 애굽의 바로, 앗수르, 바벨론, 메데파사, 세계 중앙에 있는 땅 중동에서 이스라엘의 회복과 함께 지금까지 있었던 수 차례의 이스라엘과의 전쟁, 우리는 이 애굽의 옛 영화를 꿈꾸던 에집트의 낫세르로 인하여 일어났던 중동 전쟁, 옛 앗수르의 영화를 바라보던 시리아, 메데-파사의 옛꿈을 실현해 보고자 했던 이란의 호메니옹 등, 이제는 옛 바벨론 제국의 느부갓네살의 꿈을 바라보던 이락의 사담 훗세인등 이 역사의 복고(復古)를 과연 역사의 흐름의 우연이라고 할 것인가?

결코 그럴 수가 없는 것이다. 이 모든 것은 하나님의 운행하시는 역사의 흐름대로 우리 주께서 알파와 오메가가 되시고 처음이요, 나중이신 우리 주님, 인간 역사를 시작하신 우리 주께서 이제는 인간역사를 마치시는 종말의 역사를 행하시는 우리 주님의 다시오시기 위한 인간역사의 흐름의 발자국 소리로 우리는 귀를 기우려야하겠다.

그 뿐인가 마지막 출현할 다니엘서 7장에 언급한 4번째 짐승 옛 로마 제국의 부활을 꿈꾸고자하는 구라파의 92년예정의 통합을 눈 앞에둔 우리의 세대는 전도서 3 : 15말씀과 같이 이미 옛날

Figure 7: A Korean document image with a skew angle of $-1.5°$.

우리는 이미 여러차례 있었던 중동 전쟁을 통하여 지난 일을 찾으시는 하나님의 인간 역사의 운행을 생각해 볼 수 있다. 애굽의 바로, 앗수르, 바벨론, 메데파사, 세계 중앙에 있는 땅 중동에서 이스라엘의 회복과 함께 지금까지 있었던 수 차례의 이스라엘과의 전쟁, 우리는 이 애굽의 옛 영화를 꿈꾸던 에집트의 낫세르로 인하여 일어났던 중동 전쟁, 옛 앗수르의 영화를 바라보던 시리아, 메데-파사의 옛꿈을 실현해 보고자 했던 이란의 호메니옹 등, 이제는 옛 바벨론 제국의 느부갓네살의 꿈을 바라보던 이락의 사담 훗세인등 이 역사의 복고(復古)를 과연 역사의 흐름의 우연이라고 할 것인가?

결코 그럴 수가 없는 것이다. 이 모든 것은 하나님의 운행하시는 역사의 흐름대로 우리 주께서 알파와 오메가가 되시고 처음이요, 나중이신 우리 주님, 인간 역사를 시작하신 우리 주께서 이제는 인간역사를 마치시는 종말의 역사를 행하시는 우리 주님의 다시오시기 위한 인간역사의 흐름의 발자국 소리로 우리는 귀를 기우려야하겠다.

그 뿐인가 마지막 출현할 다니엘서 7장에 언급한 4번째 짐승 옛 로마 제국의 부활을 꿈꾸고자하는 구라파의 92년예정의 통합을 눈 앞에둔 우리의 세대는 전도서 3 : 15말씀과 같이 이미 옛날

Figure 8: The resulting Korean document of Figure 7 after deskewing.

앞을 예견하기 이전, 한달 앞, 내일 일을 대비하기 힘든 불투명한 세대에 우리는 이미 서있는 것을 발견할 것이다. 90년대에 들어서서 동서 냉전시 이념체제가 붕괴되는 것 같더니 동·구라파의 격변, 동—서독의 통일, 하루 하루 숨가쁜 역사의 진전등 이락과 이란의 8년간의 전쟁이 중단됨으로 중동에서 어느정도 평화가 정착될 것으로 바라보기도 했었다. 돌연 이락의 쿠웨이트 침공으로 지금은 미국, 영국을 비롯한 다국적군과 이락과, 쿠웨이트 및 사우디 아라비아 반도에서 치열한 전쟁의 불꽃이 이미 온 인류 문명을 파괴할지도 모르는 치열한 첨단 무기까지 동원한 전쟁에서 수많은 인명의 살상 가공스러운 위력의 과학 병기 폭음과 비명 속에 살아져간 인간 영혼의 파괴를 이 처참한 전쟁을 통하여 우리는 보아왔다.

우리는 숨가쁜 가운데 스커트 마시일의 발사로 이스라엘의 얼룩진 피의 희생, 세계 제4위의 군사대국인 이락이 전쟁발발 1개월 반만에 전국토의 초토화, 그러한 참혹한 상황 속에서도 우려하던 화학전 내지 생화학 무기 또는 핵무기의 사용을 자제한 것, 이제는 중동의 평화가 깃들지 않겠는가하는 인간의 지각을 이용한 많은 추측도 나오고 있지만 중동의 불씨는 결코 사라진 것은 아니다. 다음 단계에서 더 크나큰 하나님의 행하실 일이 남아 있는 것을 간과해서는 않되겠다.

Figure 9: A Korean document image with a skew angle of 0.5°.

앞을 예견하기 이전, 한달 앞, 내일 일을 대비하기 힘든 불투명한 세대에 우리는 이미 서있는 것을 발견할 것이다. 90년대에 들어서서 동서 냉전시 이념체제가 붕괴되는 것 같더니 동·구라파의 격변, 동—서독의 통일, 하루 하루 숨가쁜 역사의 진전등 이락과 이란의 8년간의 전쟁이 중단됨으로 중동에서 어느정도 평화가 정착될 것으로 바라보기도 했었다. 돌연 이락의 쿠웨이트 침공으로 지금은 미국, 영국을 비롯한 다국적군과 이락과, 쿠웨이트 및 사우디 아라비아 반도에서 치열한 전쟁의 불꽃이 이미 온 인류 문명을 파괴할지도 모르는 치열한 첨단 무기까지 동원한 전쟁에서 수많은 인명의 살상 가공스러운 위력의 과학 병기 폭음과 비명 속에 살아져간 인간 영혼의 파괴를 이 처참한 전쟁을 통하여 우리는 보아왔다.

우리는 숨가쁜 가운데 스커트 마시일의 발사로 이스라엘의 얼룩진 피의 희생, 세계 제4위의 군사대국인 이락이 전쟁발발 1개월 반만에 전국토의 초토화, 그러한 참혹한 상황 속에서도 우려하던 화학전 내지 생화학 무기 또는 핵무기의 사용을 자제한 것, 이제는 중동의 평화가 깃들지 않겠는가하는 인간의 지각을 이용한 많은 추측도 나오고 있지만 중동의 불씨는 결코 사라진 것은 아니다. 다음 단계에서 더 크나큰 하나님의 행하실 일이 남아 있는 것을 간과해서는 않되겠다.

Figure 10: The resulting Korean document of Figure 9 after deskewing.

Figure 13.

- On the other hand, the vertical projection profile of a vertical image displays clear white-runs between the histograms which reflect *inter-column* gaps as shown in Figure 16, which displays the vertical projection of the vertical image in Figure 13. However, the vertical projection of a horizontal image displays very few gaps, if any at all. Figure 15 shows the vertical projection of the horizontal image of Figure 11.

- As a result, there are more white pixels in the horizontal projection of a horizontal document than in its vertical one. This can be noticed in Figures 12 and 15. By the same token, there are more white pixels in the vertical projection of a vertical document than in its horizontal one, as shown in Figures 14 and 16. It is worthwhile to mention that the leading and ending white-runs of the vertical and the horizontal projections are ignored because the white pixels in them are not significant as illustrated in Figure 17. It rarely happens that the number of white pixels in the vertical projection of a vertical document is the same as the number of white pixels its horizontal one, or that the number of white pixels in the horizontal projection of a horizontal document is the same as the number of white pixels its vertical one. In such a case, the document is rejected because its text orientation cannot be detected.

Our text orientation detection involves the following steps:

- Given the deskewed image, it is vertically and horizontally projected.

- The vertical and the horizontal projections are scanned in turn searching for white pixels, excluding the ones in the leading and ending white-runs of the projections.

- The text orientation is detected according to the following criteria:

  - The document is said to be horizontal in case more white pixels are found in its horizontal projection than in its vertical one.

科学が本質的な進歩を遂げるのは，どのようなときであろうか．ここではまず量子力学成立の歴史を見てみよう[1][2]．ヒルベルトやボルンの提案により，1922年の初夏，ニールス・ボーアによる彼の理論に関する連続講演会が，花々の輝くゲッチンゲンで開かれた．ゲッチンゲンにおける新しい物理学の出発の年である．誰いうとなくボーア祭と呼ぶようになった．ミュンヘンからゾンマーフェルトに連れられて，20歳の学生ハイゼンベルクもその祭典に参加していた．ハイゼンベルクは3日目のボーアの講演に異議を唱え，ボーア理論の最も核心に触れる質問を行った．ボーアは講演の後，ハイゼンベルクを散歩に誘いだし，この問題に関して長い議論を行った．以来，ボー

Figure 11: A horizontal Japanese document. Figure 12 shows its horizontal projection.

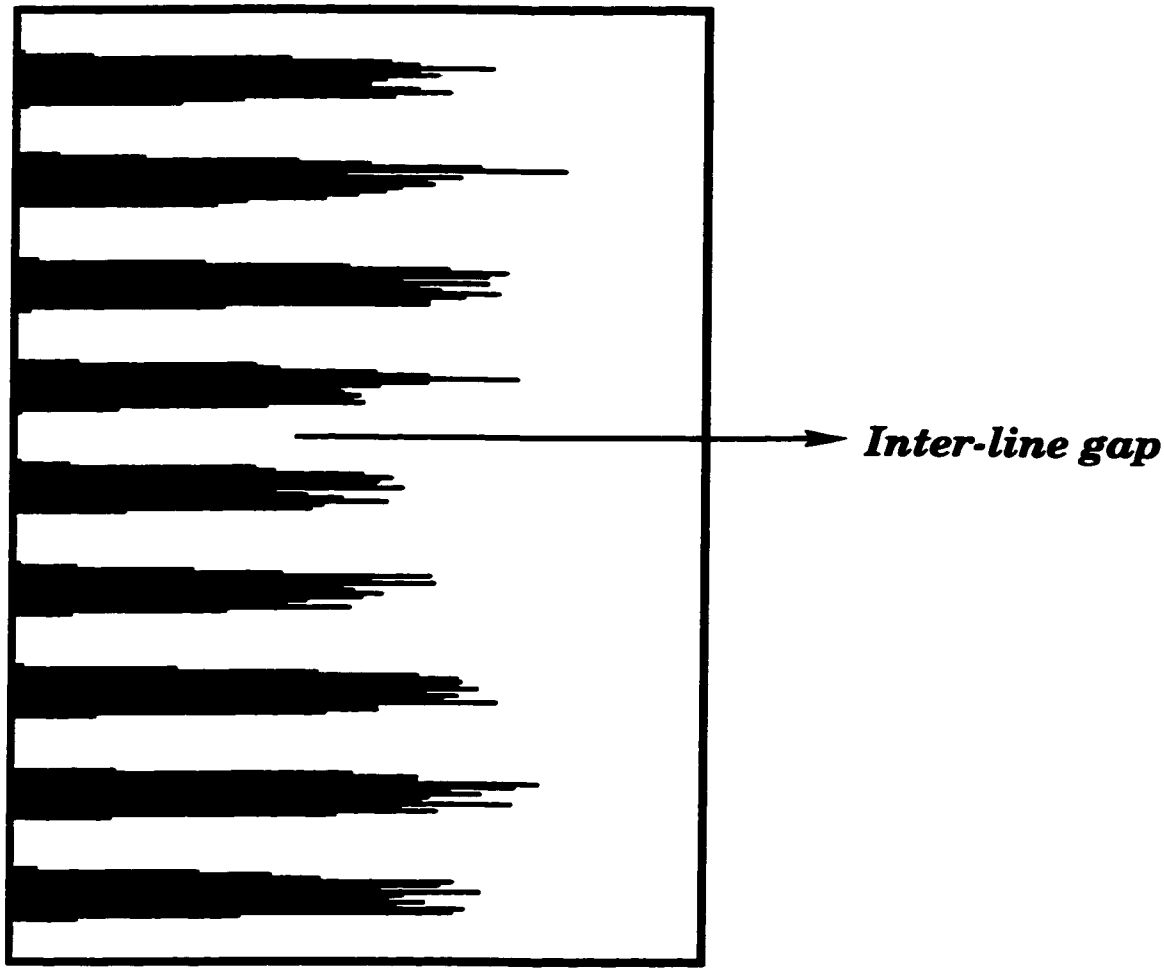

Figure 12: The horizontal projection of a horizontal Japanese document.

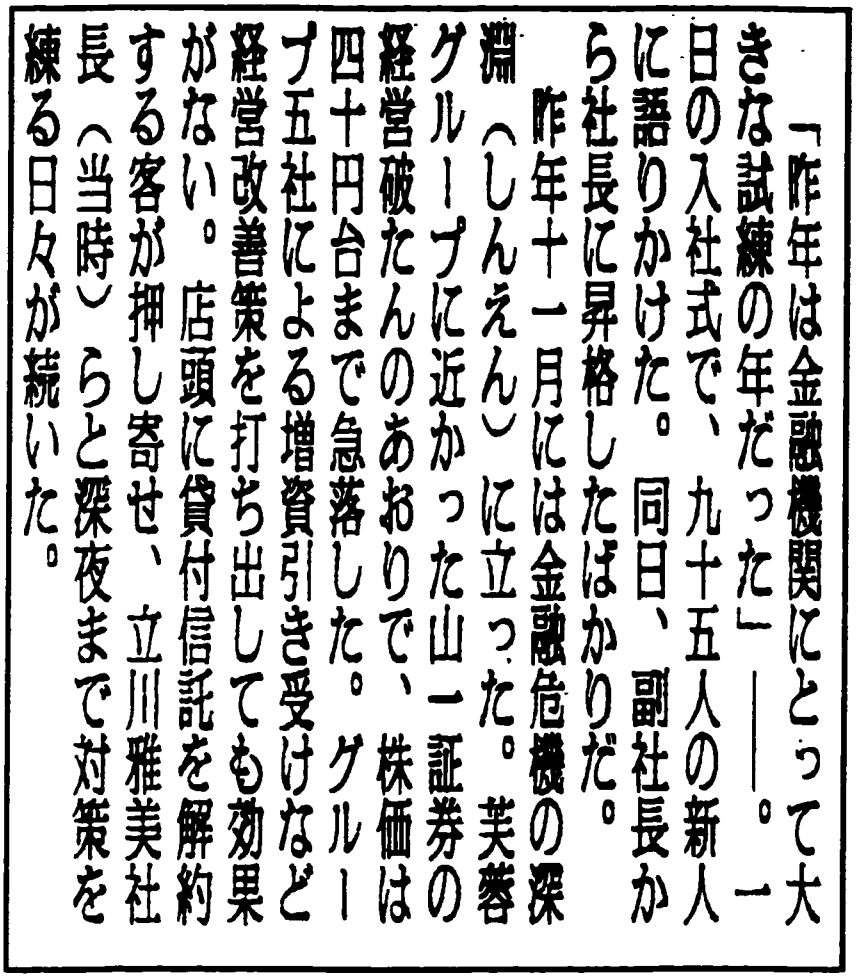

Figure 13: A vertical Japanese document. Figure 14 shows its horizontal projection.

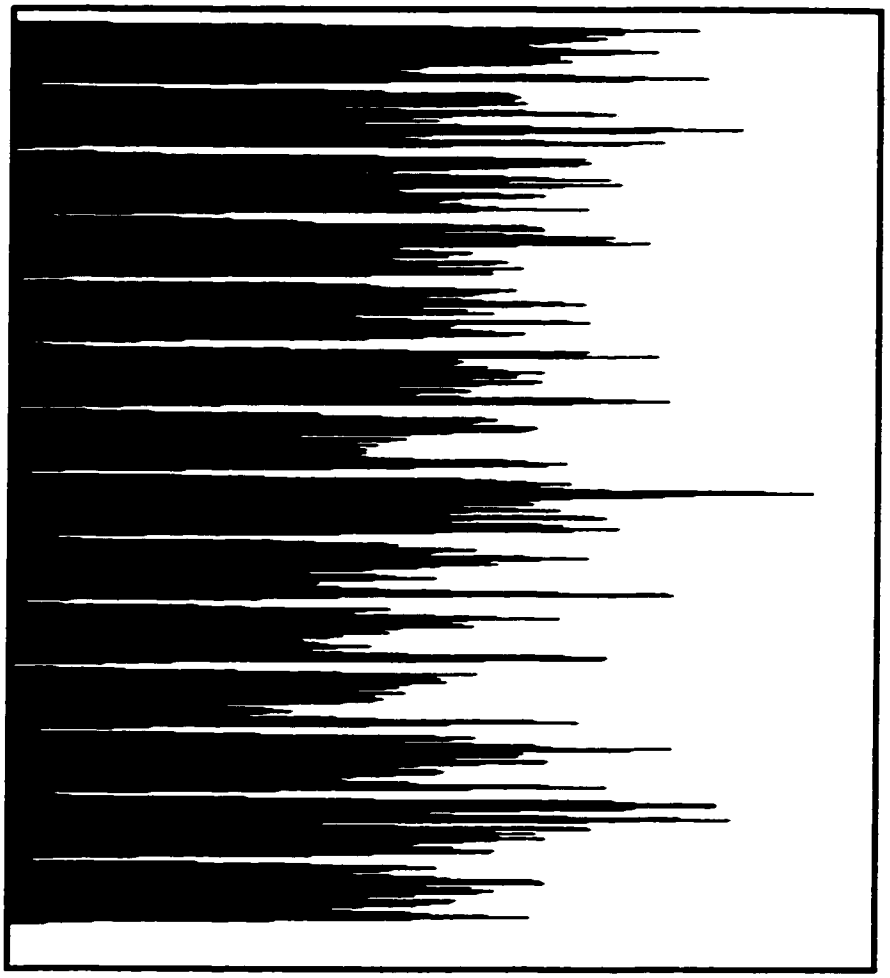

Figure 14: The horizontal projection of a vertical Japanese document.

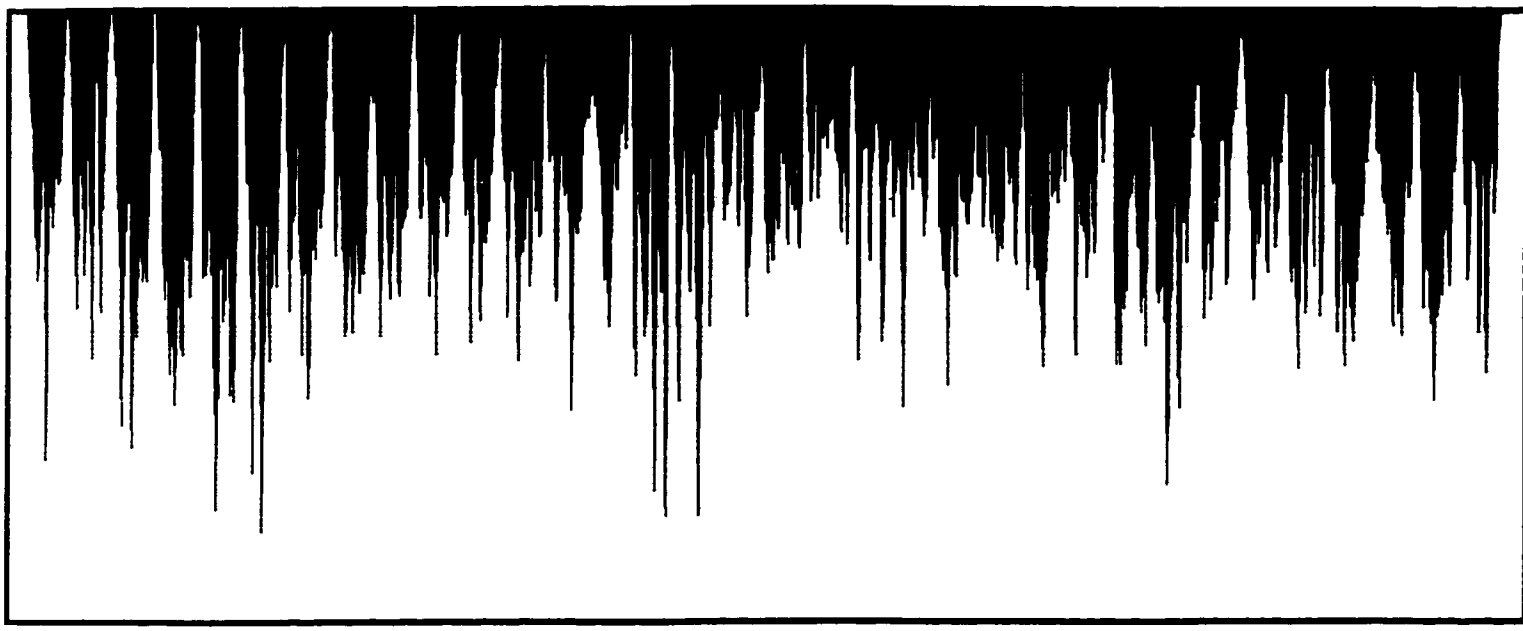

Figure 15: The vertical projection of the horizontal Japanese document of Figure 11.

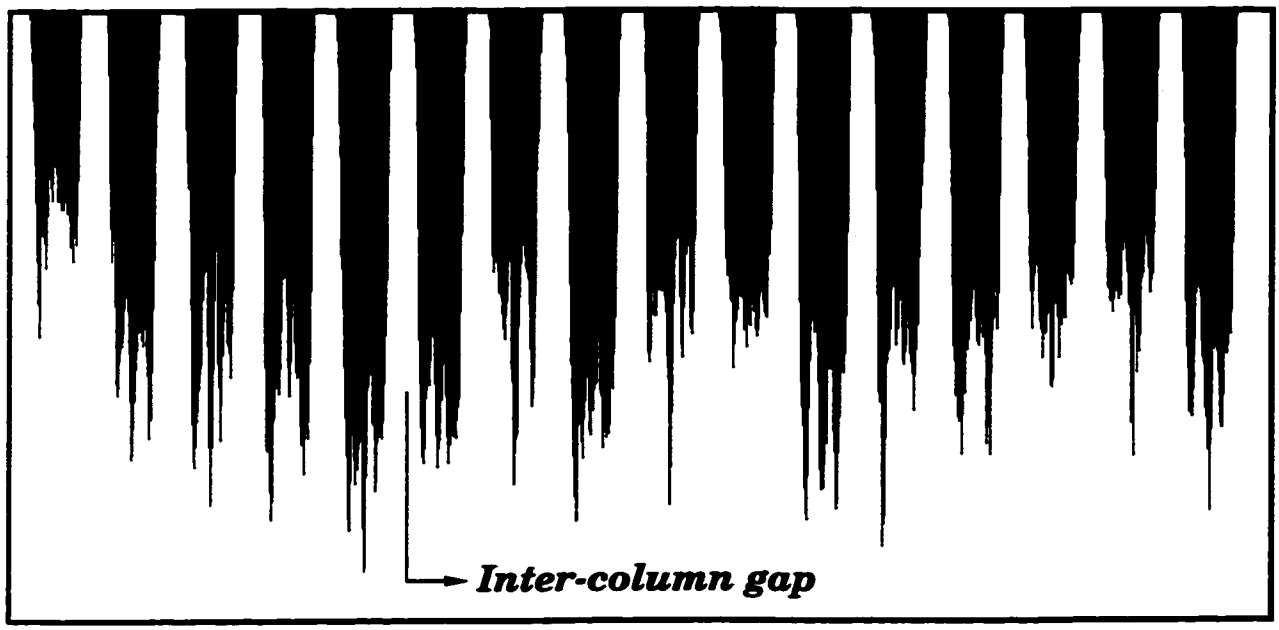

Figure 16: The vertical projection of the vertical Japanese document of Figure 13.

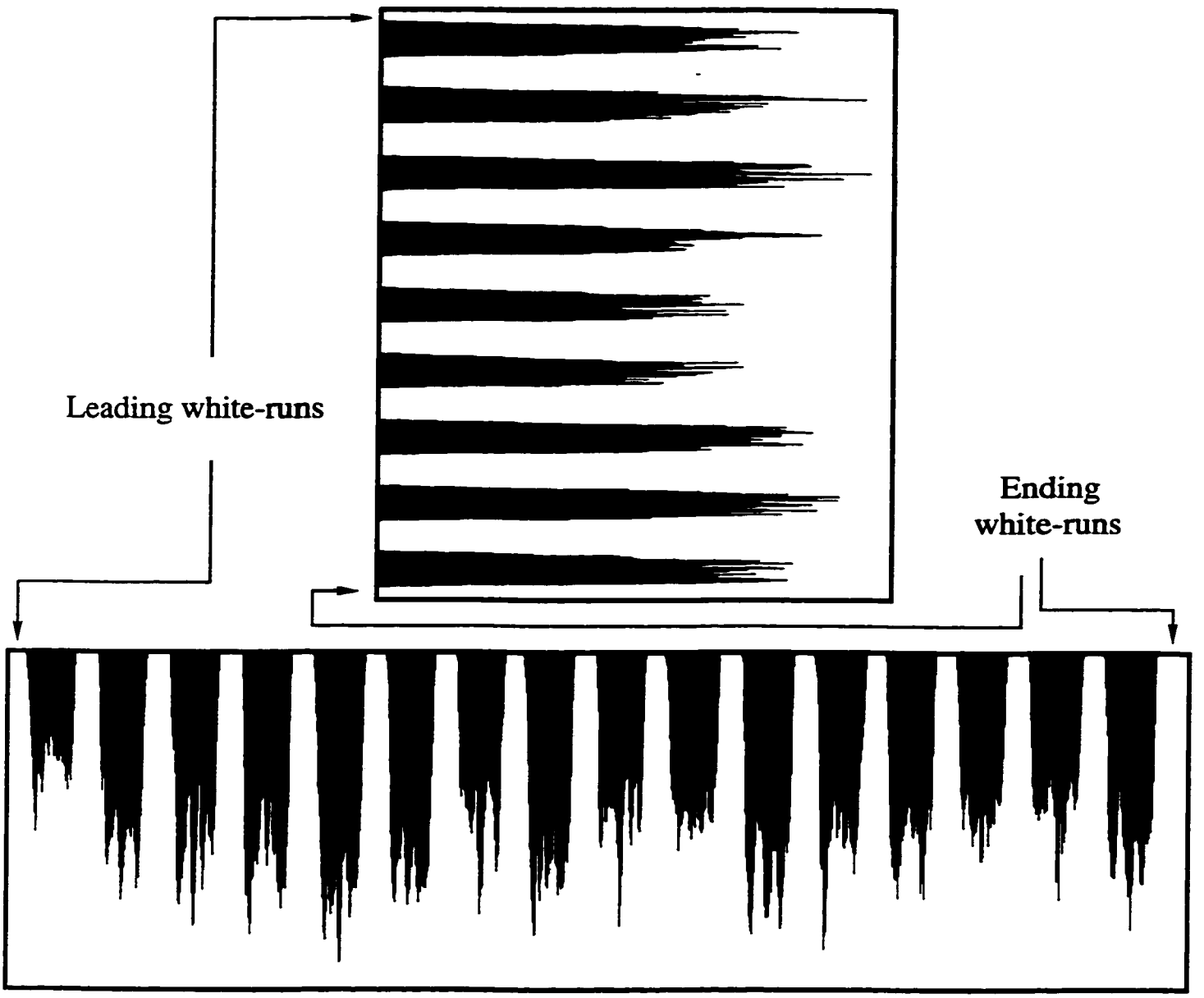

Figure 17: The leading and ending white-runs in horizontal and vertical projections.

- The document is said to be vertical in case more white pixels are found in its vertical projection than in its horizontal one.
- In case both of the projections consisted of the same number of white pixels, the document is rejected because its text orientation is undetected, which halts segmentation and thus further processing.

2.2.3 Document Segmentation

Segmentation includes line or column extraction and their segmentation into character cells. Line or column extraction is done by means of projection profiles. The inter-line or inter-column gaps are considered as line or column separators. Figure 18 shows that the characters of the first line are processed, *one at a time*, before processing the characters of the second line and so on. Thus, the system finds the first line, normalizes and matches all the characters belonging to a certain line with the various models before moving on to the next line. This process is repeated until all the lines in the image are processed.

The segmentation algorithm works as follows:

- For a horizontal image:
  1. we first start by projecting it horizontally to get the lines constituting it.
  2. The projection is scanned to locate a white space separating two consecutive peaks. The top and bottom boundaries of a line are obtained.
  3. The line just extracted is projected vertically to get its individual characters and the projection is scanned to locate a white space separating two consecutive characters. At this point, the width of a character is obtained.
  4. The character is then projected horizontally to get its height.
  5. Now that the height and the width of the character are known, it is normalized to the size of the templates used later for the search. The character is ready to be matched with the various models.

6. Steps 2 to 5 are repeated until the horizontal projection in step 1 is completely scanned, i.e., until the last line is reached.

- For a vertical image:

1. we first start by projecting it vertically to get the columns constituting it.
2. The projection is scanned to locate a white space separating two consecutive peaks. The left and right boundaries of a column are obtained.
3. The column just extracted is projected horizontally to get its individual characters and the projection is scanned to locate a white space separating two consecutive characters. At this point, the height of a character is obtained.
4. The character is then projected vertically to get its width.
5. Now that the height and the width of the character are known, it is normalized to the size of the templates used later for the search. The character is ready to be matched with the various models.
6. Steps 2 to 5 are repeated until the vertical projection in step 1 is completely scanned, i.e., until the last column is reached.

2.2.4 Size Normalization

The normalization transformation is based on linear image transform. Since the size of the image to be normalized may be larger or smaller than the standard size, we should make some mapping by checking image indices of both images. Otherwise, some pixels of the normalized image may not find their corresponding pixels from the original image.

2.3 The Proposed Identification Method

Our system automatically distinguishes Oriental documents, namely Chinese, Japanese, and Korean, stored electronically in image form. The essence of our identification approach is to look for instances of frequent characters, called models, of each script in

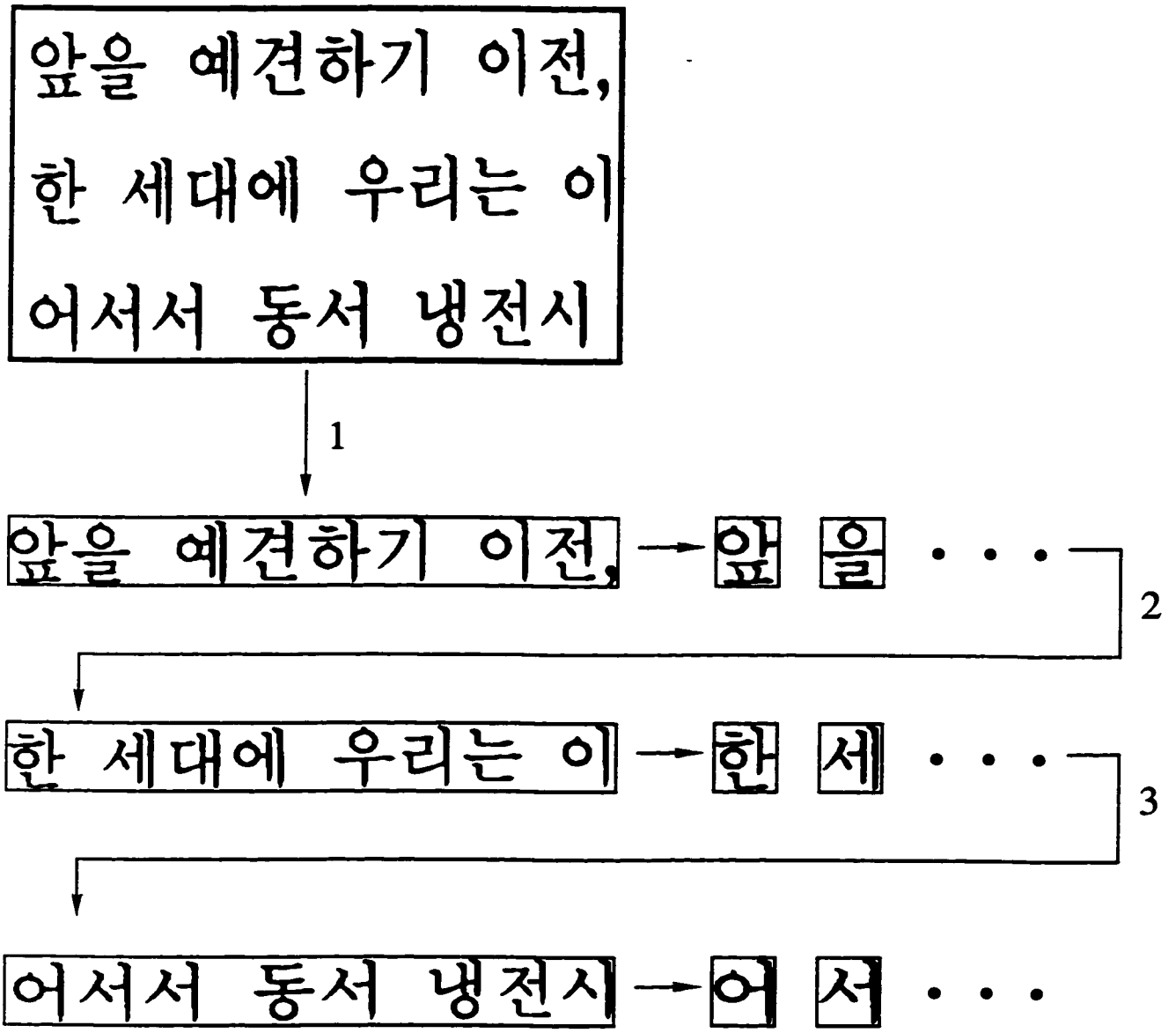

Figure 18: A sample horizontal Korean document segmented into lines and characters.

the input documents. We use the Hausdorff distance to test the shape resemblance between the representative models and the characters of the input document.

2.3.1 The Hausdorff Distance

Computer vision is the study of methods that take an image, and extract some information from the image: what objects there are in the image, their location, their shapes and so on. One interesting topic in computer vision is that of *model-based recognition.* This is the problem of locating an object, of which the computer has a model, in an image [Ruc96]. Model-based recognition is applicable to many practical task domains such as document processing applications.

Our differentiation between Chinese, Japanese and Korean documents is based on the Hausdorff distance, which is a similarity measure defined between two sets of points. In model-based recognition, one of the point sets represents a model of some object, and the other represents an image to be searched for that object.

Suppose that we have two sets of points, representing a model and an image. The Hausdorff distance between these two point sets is small when every point in the model is close to some point in the image, and every point in the image is close to some point in the model. Intuitively, this is a good property to have for model-based recognition because it defines a measure of how much the sets look like each other. The Hausdorff distance is also a metric. That is, the distance function is everywhere positive, and has the properties of identity, symmetry and the triangle inequality. These properties correspond to our intuitive notions of shape resemblance, namely that a shape is identical only to itself, the order of comparison of two shapes does not matter, and two shapes that are highly dissimilar cannot both be similar to some third shape [HKR93].

The Hausdorff distance is actually composed of two distances: the *forward distance*, h(M,I), which is the distance from the model M to the image I, and the *reverse distance*, h(I,M), the distance from the image I to the model M. h(I,M) identifies the

point $i \in I$ that is farthest from any point of M and measures the distance from i to its nearest neighbour in M. In effect, h(I,M) ranks each point of I based on its distance to the nearest point of M and then uses the largest ranked such point as the distance (the most mismatched point of I). h(I,M) will be small when exactly *every* point of I is near some point of M. Similarly, h(M,I) will be small when every point of M is near some point of I, and the *undirected Hausdorff distance* H(I,M) will be small when both of these are true, [HKR93]. The Hausdorff distance H(I,M) is the maximum of h(I,M) and h(M,I). It measures the degree of of mismatch between two sets by measuring the distance of the point of I that is farthest from any point of M and vice versa. Ideally, if I is similar to M, H(I,M) is small so that, equivalently, the degree of mismatch between the image and the model is small.

2.3.2 Definition

The Hausdorff distance measures the extent to which each point of a model set lies near some point of an image set and vice versa. Thus, this distance can be used to determine the degree of resemblance between two objects that are superimposed on one another.

The Hausdorff distance between two finite sets I = $\{i_1, \ldots, i_p\}$ and M= $\{m_1, \ldots, m_q\}$ of points in the plane is defined as

$$\mathbf{H(I,M) = max(h(I,M), h(M,I))} \tag{3}$$

where

$$\mathbf{h(I,M) = max_{i \in I}(min_{m \in M}\|i - m\|)} \tag{4}$$

and $\|.\|$ is some underlying norm on the points of I and M (e.g., The L_2 or Euclidean norm).

Computing h(M,I) necessitates the finding of the distance from each point of M to the nearest point of I, which involves many distance determinations. This computation can be made significantly more efficient by pre-computing an array of these distance values: compute an array D such that D[x,y] (for integers x, y) is the distance from the point (x,y) to the closest point of I. This array is called the *distance transformation* of I. Sub-section 2.3.5 presents the algorithm used to compute the distance transformation of a binary image.

2.3.3 The Modified Hausdorff Distance

It should be noted that the Hausdorff distance is susceptible to noise. A number of extensions [HKR93, Pau97] have been proposed to improve its resilience to noise. For example, the Hausdorff distance can be calculated on a k*th* (k is a proportion of p and q) largest, instead of the maximum, distance of two images. The selection of k is arbitrary that important information may be cut off. A better choice, proposed in [Li98], is instead of using the maximum of minimum distance, use the average minimum distance defined below:

$$\mathbf{H(I,M)} = \frac{\sum_{\mathbf{m \in M}} \mathbf{h(m,I)} + \sum_{\mathbf{i \in I}} \mathbf{h(i,M)}}{\mathbf{p+q}} \tag{5}$$

where

$$\mathbf{h(i,M)} = (\mathbf{min_{m \in M}} \|\mathbf{i-m}\|)$$

$$\mathbf{h(m,I)} = (\mathbf{min_{i \in I}} \|\mathbf{m-i}\|)$$

h(i,M) is the minimum distance from a point in I to any point in M. p and q are the number of pixels in the image and model, respectively.

This distance is called the *averaged Hausdorff distance* and it reflects the overall likeliness between two point sets and is more resilient to noise [Li98].

For each character model, we compute its distance transformation. When a character from the input image comes in for matching, we first scale it to the size of the character model, generate its distance transformation, and then match it to the character model, starting at the top left corner and moving left to right, top to bottom. The average Hausdorff distance between two images is calculated as the sum of the minimal distances for all black pixels in one image to the corresponding pixels in the other image, averaged by the total number of pixels in each image.

In equation (5), let $S1 = \sum_{m \in M} h(m, I)$, $S2 = \sum_{i \in I} h(i, M)$ and $H = H(I, M)$, for illustration purposes. Part (a) of Figure 19 shows the most frequently used Chinese model, part (b) shows an instance of the model in part (a) extracted from a document, while part (c) shows another Chinese character that is obviously different from the model in part (a). The Chinese model and the character in part (b) are said to match in case the Hausdorff distance H between them falls below the threshold value (0.05634) corresponding to the Chinese model of part (a). Otherwise, a mismatch between them is produced. Same for the Chinese character in part (c). Since the model and the image in part (b) are in fact the same, we expect the resulting Hausdorff distance to be less than or equal to the threshold value. The opposite should occur for the image in part (c), namely that the resulting Hausdorff distance should be greater than the threshold value stated above. Figure 19 shows the numerical values of $S1$ and $S2$ stated above as well as the modified Hausdorff distance H, computed according to equation (5), when both a match and a mismatch are produced. For part (b) of Figure 19, $S1 = (0 + \ldots + 1 + 1 + \ldots + 0 + \ldots + 1.4142 + \ldots)$ and $S2 = (0 + \ldots + 1 + 0 + \ldots + 1 + 0 + \ldots + 1 + \ldots, 1.4142, \ldots)$. As for part (c) of Figure 19, $S1 = (0 + \ldots + 1 + 2 + 3 + \ldots + 1.4142 + 2.8284 + \ldots + 4 + \ldots)$ and $S2 = (0 + \ldots + 1 + 1 + 1.4142 + 0 + \ldots + 2.8284 + \ldots)$. Note that $S1$ and $S2$ are divided by q and p, respectively, resulting in their mean values.

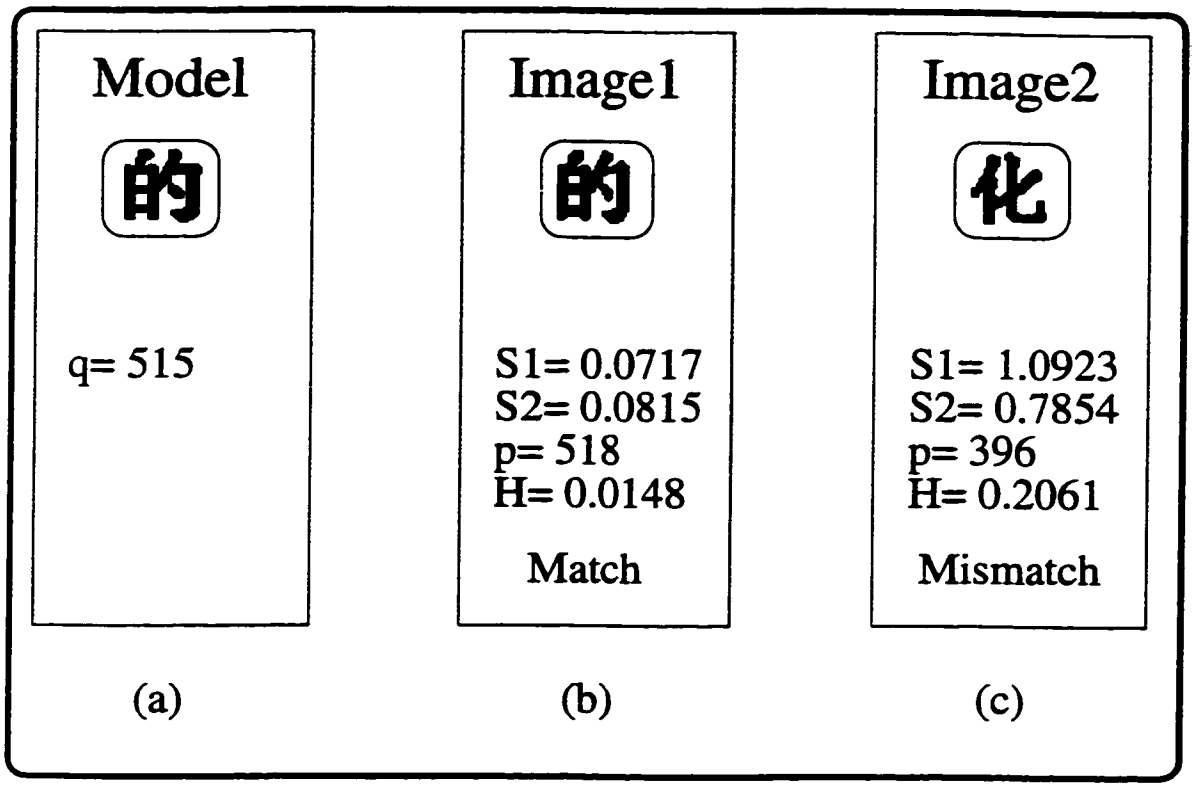

Figure 19: An example of a match and a mismatch. S1, S2, and H are explained at the end of sub-section 2.3.3. p and q are the number of pixels in the image and model, respectively.

2.3.4 Determining Thresholds

After the Hausdorff distance is computed, we must use a threshold value to determine whether or not that distance indicates a match. Fixing the threshold value, say T, heuristically is impractical because T may be appropriate to identify a match between a certain Chinese model A and a character B, but not between a Korean model C and a character D even tough C and D are alike, for example. For us to compute accurate threshold values, we interactively validated the matches found by the system, accumulated the Hausdorff distances resulting from correct matches and used these distances to *statistically* generate one threshold value for each of our models. We collected a total of *391* distances to determine thresholds for the Chinese models, *896* for the Japanese models, and *1081* for the Korean models. At the end, we determine the thresholds such that the probability that the Hausdorff distance d is less than or equal to the threshold value T is greater than or equal to $X\%$ as follows:

$$P(d \leq T) \geq X\%.$$

2.3.5 Distance Transformation

Consider a digital binary image, consisting of feature (black) and non-feature (white) pixels. A distance transformation converts a binary digital image into an image where all non-feature pixels have a value corresponding to the distance to the nearest feature pixel. An original binary image, to which the distance transformation (DT) is to be applied, consists of feature pixels with the initial value zero, and non-feature pixels with the initial value infinity, i.e., a suitably large number.

Borgefors, in [Bor86], presented various distance transformations and their use in different applications. The computation of the DT is either parallel or sequential. Borgefors suggested the algorithm published in [Dan80] for a sequential computation of the Euclidean distance transformation (EDT) and mentioned that it does not always give correct results. The author also suggested a parallel EDT algorithm, published in [Yam84] and used by this work, that always gives correct results. It is explained below.

The Euclidean Distance Transformation Algorithm

Given a binary image with two sets of pixels,

$$S = \text{set of 1's, the black points;}$$
$$S' = \text{set of 0's, the white points.}$$

A distance transformed image (or distance map) **L(S)** is an image such that for each pixel

$$\mathbf{z} = (\text{i,j}) \in S,$$

there is a corresponding pixel in **L(S)** where

$$\mathbf{L(S)} = min_{\mathbf{z}' \notin S} d_e(\mathbf{z}, \mathbf{z}').$$

Here, $\mathbf{z} = (\text{i,j})$ and $\mathbf{z}' = (\text{i}', \text{j}')$, $0 \leq i, i' \leq M-1, 0 \leq j, i' \leq N-1$, are points on a two-dimensional picture and d_e is the *Euclidean distance.* Specifically, the Euclidean distance is given as follows:

$$d_e(\mathbf{z}, \mathbf{z}') = \sqrt{(i-i')^2 + (j-j')^2}.$$

On a discrete coordinate system such that $\mathbf{z}$ is discrete, u_n of Figure 20 is used as a vector to a neighbouring point. Here, u_0 is a zero vector, so $\mathbf{z} + u_0 = \mathbf{z}$ and

$$U_8'(\mathbf{z}) = \{\mathbf{z} + u_m \mid m = 0, 1, 2, \ldots, 8\}$$

Algorithm: *Euclidean Distance Transformation*

- *Initial Condition:* The initial value of this process at the iteration time $t = 0$ is given by

$$\mathbf{L}^0(\mathbf{z}) = \begin{cases} (0,0), & \text{if } \mathbf{z} \in S \\ (Z,Z), & \text{if } \mathbf{z} \notin S, \end{cases}$$

where Z is a positive integer which is large enough.

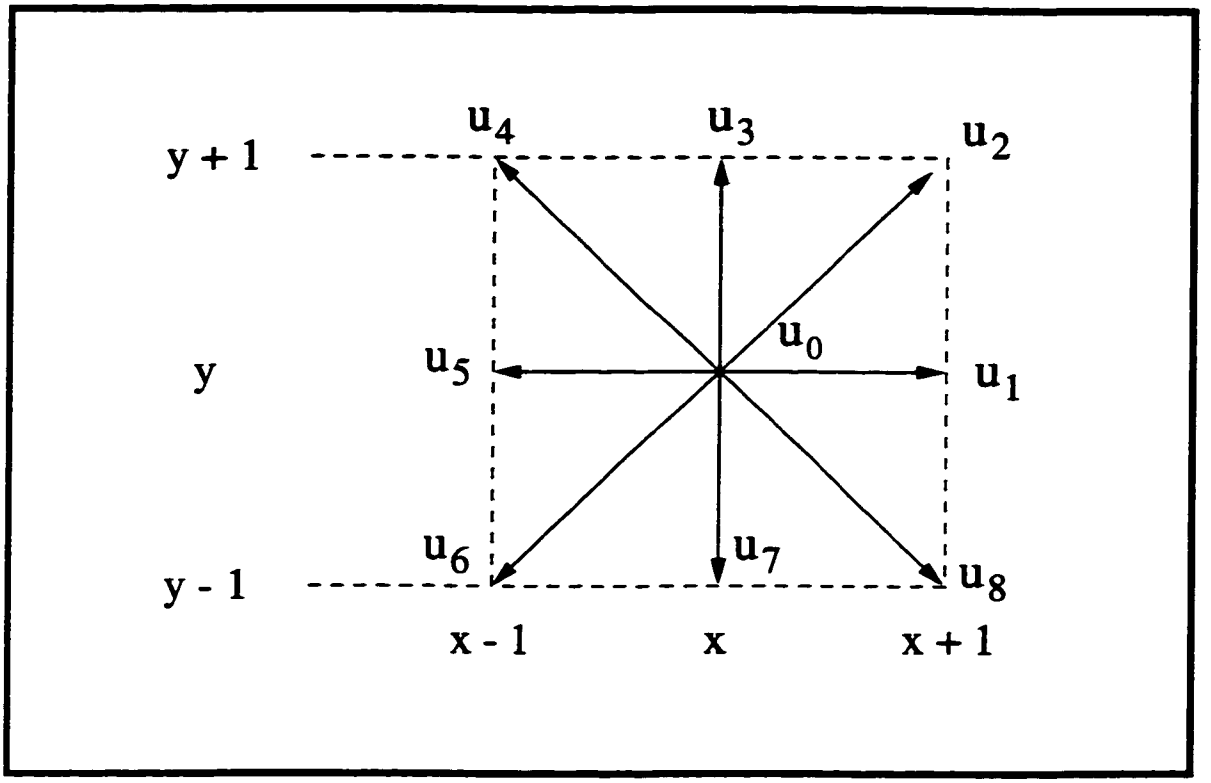

Figure 20: Definition of neighbours. The symbols u_i, i = 0, 1, 2, ..., 8, are vectors to the neighbours. u_0 is a zero vector, and so the neighbour of i = 0 means the point itself.

- *Iteration*: For $t = 1, 2, 3, \ldots$, the propagation is given by

$$\mathbf{L}^t(\mathbf{z}) = \mathbf{L}^{t-1}(\mathbf{z} + u_m) + u_m,$$

where m satisfies the equation,

$$| \mathbf{L}^t(\mathbf{z} + u_m) + u_m | = min_{u_n \in U'_k(z)}\{| \mathbf{L}^{t-1}(\mathbf{z} + u_n) + u_n |\}.$$

If there are multiple u_n's which satisfy the minimum value, the minimum n is taken as m.

- *Stop Condition*: The stop condition of the iteration is

$$| \mathbf{L}^t(\mathbf{z}) | = | \mathbf{L}^{t-1}(\mathbf{z}) |, \text{ for all } \mathbf{z}.$$

Then, the distance transformed picture is

$$g(\mathbf{z}) = | \mathbf{L}^T(\mathbf{z}) | = \sqrt{(\mathbf{L}_i^T(\mathbf{z}))^2 + (\mathbf{L}_j^T(\mathbf{z}))^2},$$

where T is the final value of t.

When k = 8, that is $U'_k = \{ u_0, u_1, u_2, \ldots, u_8\}$, it is called the propagation with 8-neighbours.

2.3.6 Templates Used in the Search

The essence of our model-based approach is to locate instances of certain templates, or models, in an input document. These templates are a subset of the most frequently appearing characters in each language [SMR98]. The seven mostly used characters in each of the three Asian languages were sufficient to locate enough matches to identify a new document since the seven most frequently appearing Chinese, the seven Japanese, and the seven Korean characters alone constitute approximately 14.98%, 16.92%, and 20.00%, [SMR98], of a Chinese, Japanese, and Korean text, respectively. However,

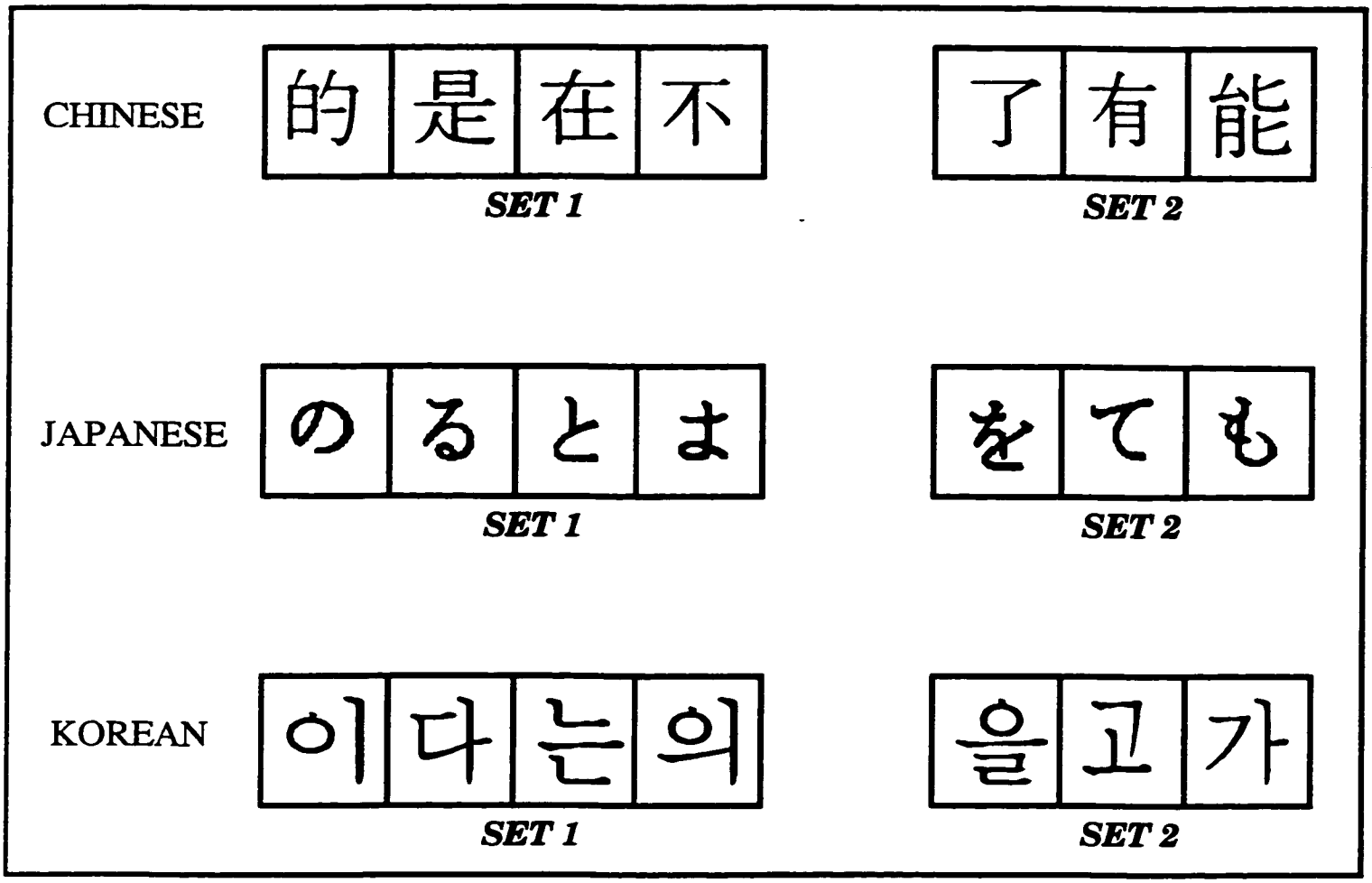

Figure 21: The Chinese, Japanese, and Korean templates.

the system can be made to look for any number of models as estimated necessary. Our models are normalized to 30x30 pixels. Figure 21 shows the Chinese, Japanese and Korean models looked for in the input document. In that figure, the Chinese models belong to the *Song* font, the Japanese belong to the *Mincho* font, and the Korean belong to the *New Myung Jo* fonts. These are the fonts used to print the majority of documents. The reason the models are split into two sets, SET1 and SET2 will be explained in section 2.4.

2.3.7 Handling Multiple Fonts

The documents printed in fonts markedly different from the ones used in training present a problem for systems based on template matching techniques because the templates on which the system is trained do not match with the characters in the input document obviously due to remarkable shape difference. To handle variations in fonts, we trained our system on a total of 5 commonly used fonts for Chinese, Japanese, and Korean other than the ones used for the majority of documents, namely Song, Mincho, and New Myung Jo.

The various fonts on which the system is trained are the following (the figures enclosed in brackets show sample documents of each font):

- 2 Chinese: *Kai* (Figure 22), and *Black* (Figure 23),
- 1 Japanese: *Gothic* (Figure 24), and
- 2 Korean: *Gyun Myung Jo* (Figure 25), and *Graphic* (Figure 26).

To handle font variations, our system not only looks for the basic models depicted in Figure 21 but also looks for the same models in different fonts in case it could not locate instances of the basic templates. The performance of the system tested on documents printed in different fonts will be presented in chapter 3 covering the results achieved. Figure 27 shows the Chinese models on which the system is trained but in the Kai and Black fonts. Figure 28 shows the Japanese models in Gothic font and Figure 29 depicts the Korean models in Gyun Myung Jo and Graphic fonts.

2.4 The Method

The essence of our identification approach is to locate instances of some frequent character models in the input documents. As noted in Figure 21, two sets of models are looked for in the image, SET1 and SET2. The purpose of having 2 sets of templates makes the method more robust since the system is made to look for another set of

一番事业，但经亲属和朋友的归劝，最后还是选择“提前退休”这条路子。退休后他办起了一家电子公司，当了老板，在电子世界一展他的才能，干起来顺顺当当，实现了他自身的价值。

无奈的退却。经朋友介绍，笔者认识了一位38岁就“退休”的年轻人，他在一家不太景气的企业工作，为了经济上宽松一点，上班以外他干起第二职业，经过几年的“地下工作”，他对经营买卖上了路子。于是，他向厂里提出停薪留职，但厂里因种种理由要他每月交回100多元的停薪留职的费用。迫于无奈，最后他通过多种途径，去医院办了一张因病不能坚持工作的证明。就这样，38岁便顺顺当当地办了“提前退休”手续，挤进了退休人员的行列，与那些老年退休工人一样，反到按月退休费。而在岗职工就要根据经营情况而定。孙某虽然38岁，却是已有20年工龄的老职工了，尽管她每天起早贪黑，按时上下班，但每月只能拿到60%的工资，而且有时拖好长时间才能兑现。后来经过多方努力，她也很快加入了“提前退休”者的行列。退岗后，她每月领到的退休费超过在岗时的报酬，并且能按时领取。

舍“卒”保“帅”。她是一名国有大型企业的女工，丈夫是厂里的领导，为了企业的经营发展，她的丈夫不是出差走南闯北，就是在厂里忙不完的开会和解决厂里扯皮纠纷问题，每天总是早上7点左右就出家门，经常很晚才归。为了给丈夫分忧，照顾他的起居生活，照顾家庭，离正常退休年龄还有15年的她，也办理了“提前退休”手续。

Figure 22: A sample Chinese document printed in Kai font.

各位先生、女士：馬克思的確是一條「硬漢」。這位日爾曼天才高高凌駕在他那個時代的科學知識之上。他只管創造關於哲學的哲學，全然不顧有多少人能明白。結果呢？是一系列完整的、高層次的著述面世，對於普通讀者，這些厚重得難以消化。馬克思，一塊難啃的骨頭。

本册子嘗試爲你提供一些較易下咽的東西——馬克思思想的扼要說明。自知水平有限(僅小學五年級程度！)，但書寫認眞，能讓讀者明白。拙著若還不至於通篇語無倫次，作者便知足矣。

Figure 23: A sample Chinese document printed in Black font.

(10) 外務省中国課監修、前掲書、一一一～一一二頁。言うまでもなく、日米安保条約に「台湾条項」などというものは存在しない。一九六九年一一月の佐藤・ニクソン共同声明における「台湾条項」と混同しているのであろう。いかにも田中角栄らしい問答ではある。日米安保の存在意義について、積極的理由をなんら展開せず、ただ単に「中国がいいと言っている」というのは、積極的理由を述べることを意図的に避けたものなのか、それとも田中角栄自身、積極的理由を見いだせなかったのか、不明である。後者だとすると、この時期、やはり日米安保はきわめて危うい状態にあったことになり、中国が日米安保を救ったということになるのかもしれ

Figure 24: A sample Japanese document printed in Gothic font.

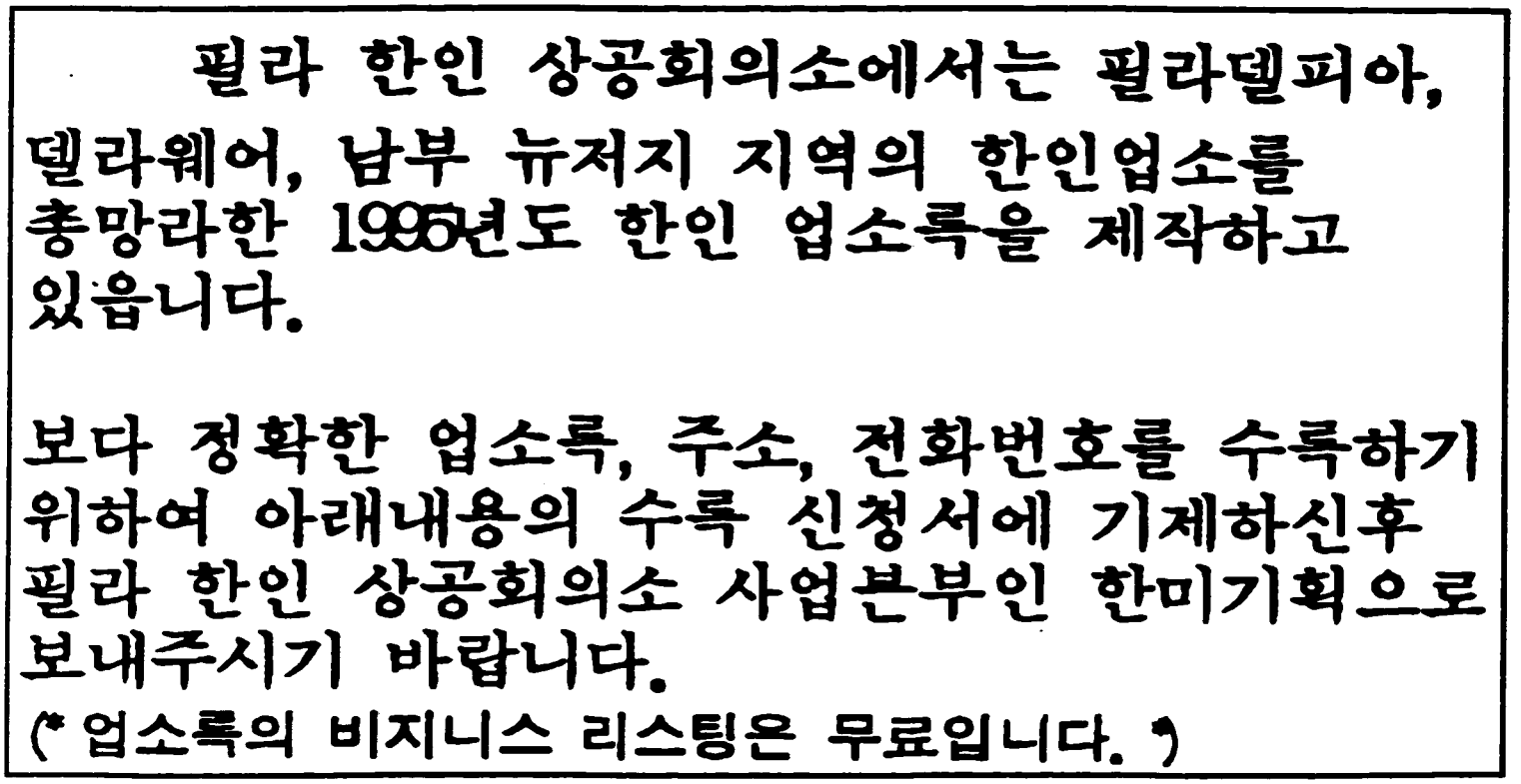

Figure 25: A sample Korean document printed in Gyun Myung Jo font.

찬 사람을 불러 이르시기를 너는 예루살렘 성읍 중에 순행하여 그 가운데서 행하는 모든 가증한 일로 인하여 탄식하며 우는 자의 이마에 표하라 하시고 나의 듣는데 또 그 남은 자에게 이르시되 너희는 그 뒤를 좇아 성읍 중에 순행하며 아껴보지도 말며 긍휼을 베풀지도 말고 쳐서 늙은 자와 젊은 자와 처녀와 어린아이와 부녀를 다 죽이되 이마에 표 있는 자에게는 가까이 말라 내 성소에서 시작할찌니라 하시매 그들이 성전 앞에 있는 늙은 자들로부터 시작하더라."

Figure 26: A sample Korean document printed in Graphic font.

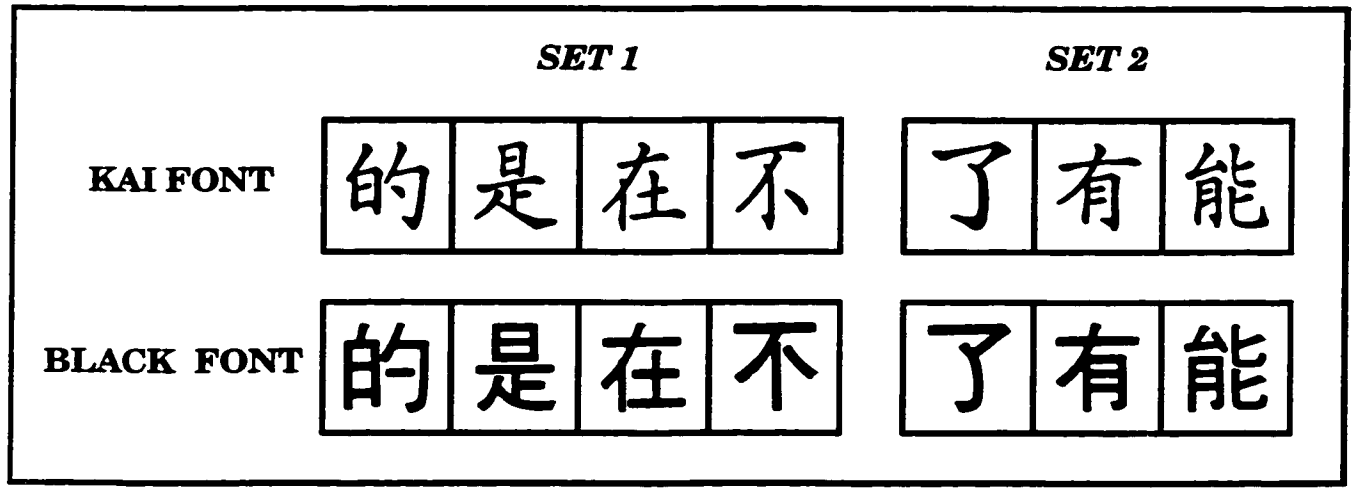

Figure 27: The Chinese models in Kai and Black fonts.

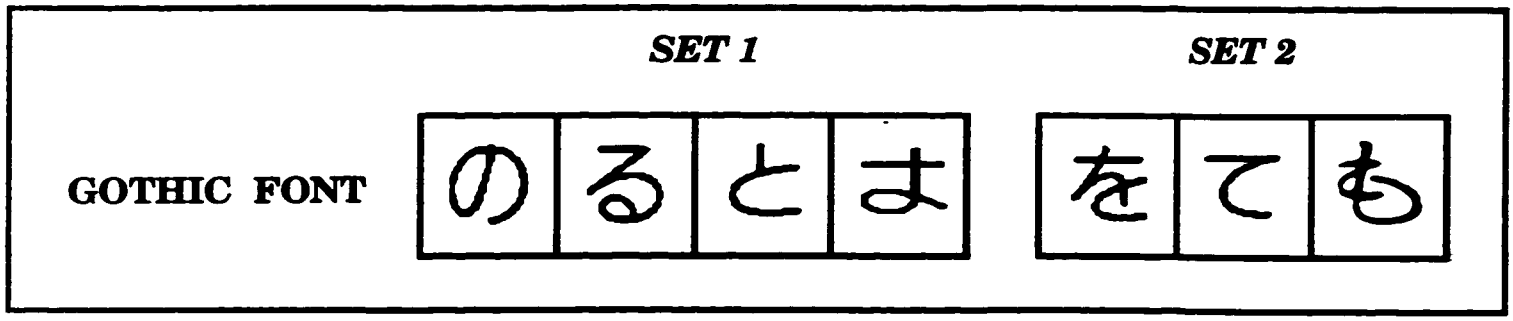

Figure 28: The Japanese models in Gothic font.

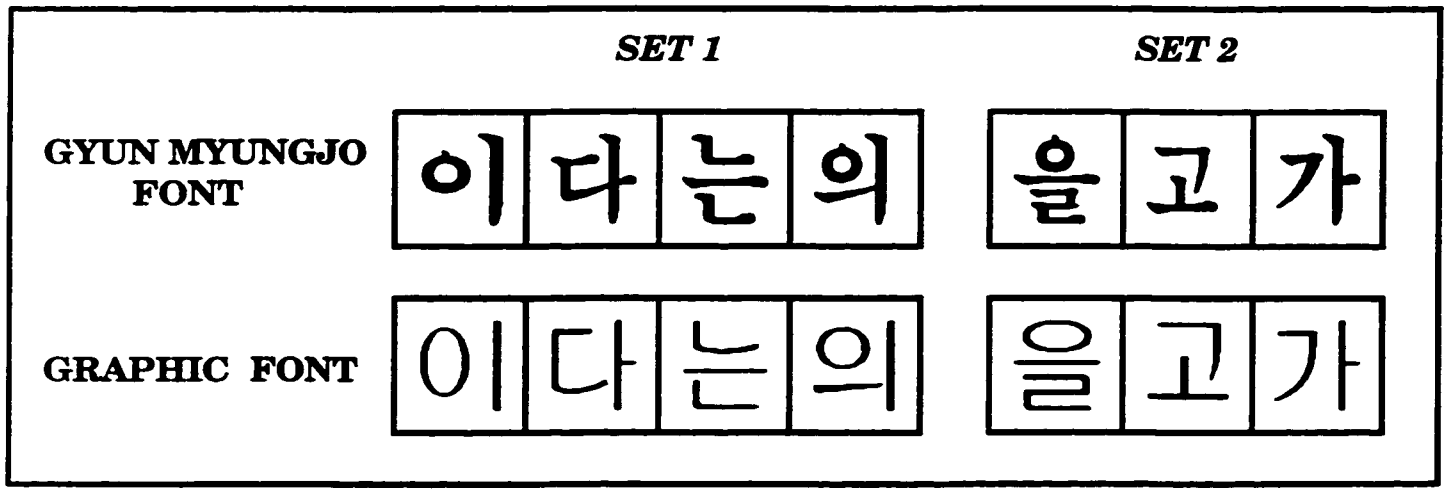

Figure 29: The Korean models in Gyun Myung Jo and Graphic fonts.

templates in case the first one did not generate any matches, instead of prematurely rejecting the document since no matches were found at all. The number of character models in each is flexible and depends on the classification problem. In our case, experiments have shown that a total of seven models in both sets was enough to reliably identify documents. Increasing the number of templates in each of the two sets undoubtedly increases the identification rate of the system because it will definitely find enough matches but on the account of a longer processing time. The method can also be extended to allow the system to keep searching for small sets of frequently appearing models until it finds enough information to identify the input document. Our system goes through 2 iterations: the models in SET1 are looked for in the first iteration and only when no instances of the templates in SET1 were located in the document that the models in SET2 are searched for in a second iteration. The system rejects the document if could not find a sufficient number of matches upon the completion of the second search iteration.

Our method of identification has the following steps:

1. Manually extract a block of text from a new document image because Oriental documents are complex in structure and very large in size.

2. Deskew, detect the text orientation, and segment the extracted block of text to character cells. For the deskewing algorithm to work properly, the extracted block of text should consist of enough lines or columns with sufficient characters on each.

Start of the first iteration. SET1 is used in the search.

3. Match each character in the block to the Korean templates printed in the commonly used New Myung Jo font, found in Figure 21.

   The document is identified as Korean if at least 3 Korean matches were found.

   If no or fewer than 3 matches were located, step 4 is executed.

4. Match each character in the block of text to the same Korean templates but presented to the system in a different font, namely the Gyun Myung Jo font, which is used more frequently than the Graphic font. The Korean templates in Gyun Myung Jo and Graphic fonts are depicted in Figure 29.

   The document is identified as Korean if at least 3 Korean matches were found.

   If no or fewer than 3 matches were located, step 5 is carried out.

5. Match each character in the block of text to the Graphic Korean models.

   The document is identified as Korean if at least 3 Korean matches were found.

   If no or fewer than 3 matches were located, step 6 is executed.

6. Match each character in the block of text to the Japanese templates in the commonly used Mincho font, found in Figure 21.

   The document is identified as Japanese if at least 3 Japanese matches were found.

   If no or fewer than 3 matches were located, step 7 is executed.

7. Match each character again to the Japanese Gothic templates, found in Figure 28.

   The document is identified as Japanese if at least 3 Japanese matches were found.

   If no or fewer than 3 matches were located, step 8 is executed.

8. Match each character in the block of text to the Chinese templates in the commonly used Song font, found in Figure 21.

   The document is identified as Chinese if at least 3 Chinese matches were found.
   If no or fewer than 3 matches were located, step 9 is carried out.

9. Match each character again to the Chinese Kai templates, found in Figure 27.

   The document is identified as Chinese if at least 3 Chinese matches were found.
   If no or fewer than 3 matches were located, step 10 is executed.

10. Match each character again to the Chinese Black models, found in Figure 27.

    The document is identified as Chinese if at least 3 Chinese matches were found. If no or fewer than 3 matches were located, step 11 is performed.

    Start of the second iteration. SET2 is used in the search.

11. Steps 3 through 10 are repeated on the same document but with the second set of models, SET2, rather than SET1.

12. Upon the completion of step 10, if no or fewer than 3 Chinese matches were located, the document is *rejected* and the system terminates its operations.

In addition to showing the results of the above method, Chapter 3 will illustrate the results when 5 then 7 matches (instead of 3) are used for classification. Furthermore, it will pictorially present the classification and error rates of short, medium, and long passages as 3, 5, or 7 matches are used. Note that by 3, 5, or 7 matches, we do not mean 3, 5, or 7 *templates* matched with a character from the input document but the minimum number of matches that needs to be located by the system for the document at hand to be classified.

2.5 Some Special Models

Some of the most frequently appearing Korean and Chinese character templates used in the search are made up of 2 or more components, split horizontally or vertically *by the segmentation algorithm*, rather than one single entity; this is not the case with the Japanese models as can be noticed in Figures 21 and 28. Parts (a) and (b) of Figure 30 depict Korean characters split vertically or horizontally, respectively, whereas part (c) of the same figure shows one Chinese character split vertically. These models are always split regardless of the font in which they are printed. Such character models do not generate any matches because as a result of the segmentation method, if the document is horizontal, characters whose components are vertically-split are divided into their constituent parts and thus will be considered as two or more characters

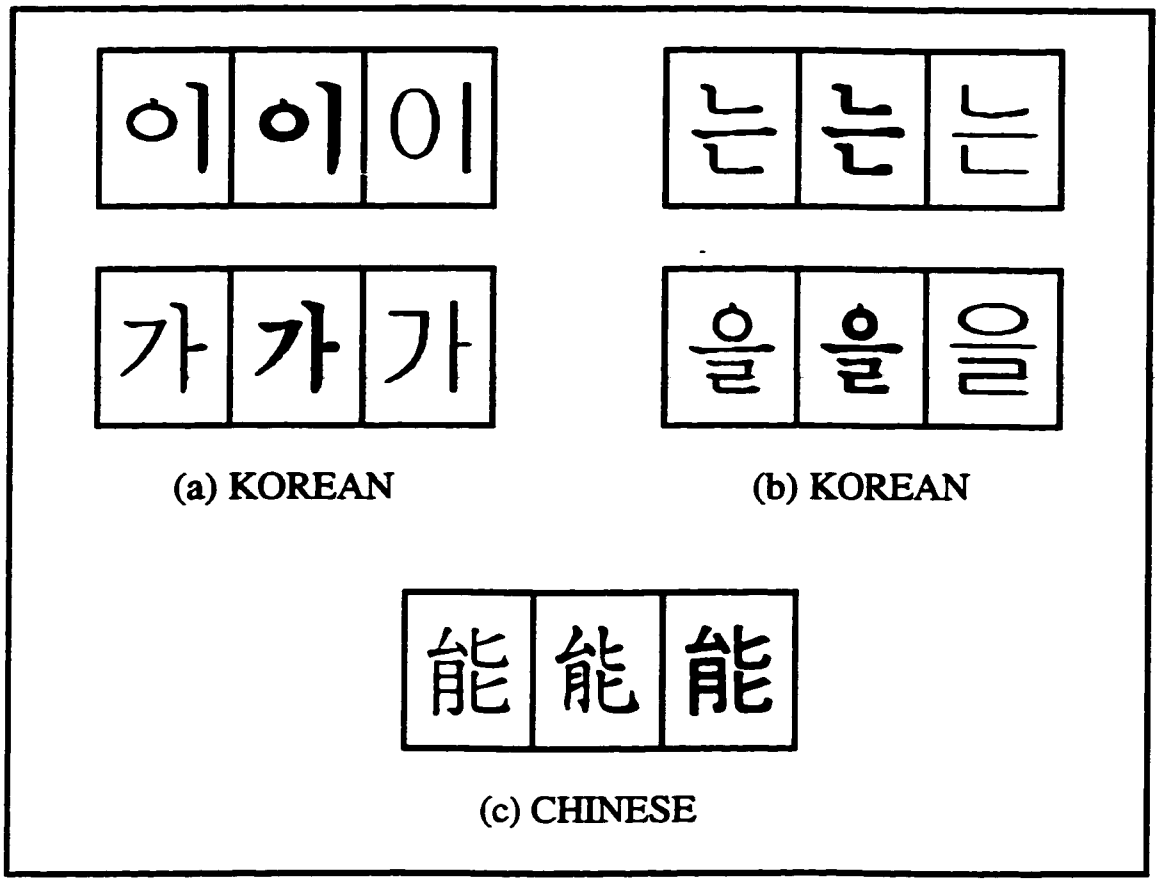

Figure 30: Some Korean and Chinese models always split regardless of the fonts in which they are printed. Each character is displayed in its 3 corresponding fonts.

rather than one. This is due to the horizontal projection that we use to divide a line into characters based on the white gap found in between the characters of the same line. It is the same for vertical documents with horizontally-split character models. This is also due to the vertical projection used to divide a column into its constituent characters. To solve that problem, our method looks for the individual parts of the character successively. If all the character's components are located one right after the other, a match is considered to be found.

Furthermore, we have to deal with other models used in the search that are normally connected in regular fonts but split in certain special fonts, and vice versa.

Part (a) of Figure 31 shows 2 Korean models that are normally connected but

will be considered as 2 separate characters due to segmentation in case the input document was a Korean printed in the Graphic font. Our method handles such a situation by looking for the connected models when the New Myung Jo and Gyun Myung Jo Korean model sets are used in the search and by looking for the constituent parts of the disconnected character model successively in the document image when the Graphic model set is used. The first Chinese character model of part (b) of Figure 31 is connected only in the Black font. In case the document to be classified is printed either in the Song or the Kai font, that character will be split into 2 by the segmentation algorithm. Its 2 individual components will be located successively as the Song and the Kai fonts are used for the search and it will be looked for as an entire character only when the Black model set is used. On the contrary, the second Chinese model in part (b) of Figure 31 is split only in the Black font. It will be searched for as a connected entity only when the system is using the Black model set in the search.

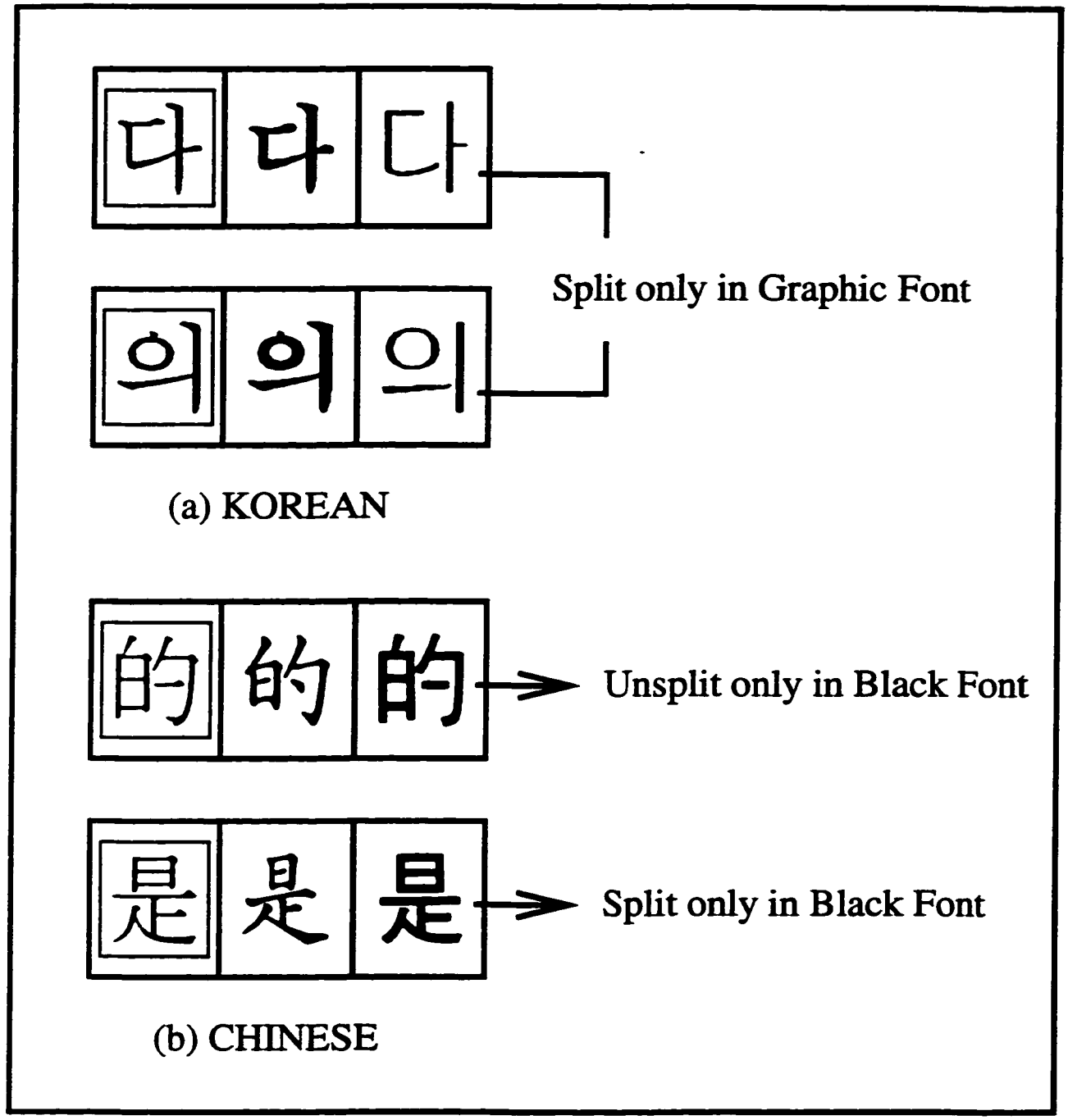

Figure 31: Some Korean and Chinese models that are split in a certain font but connected in another.

Chapter 3

Experimental Results

This chapter illustrates the results achieved when the system is tested on Mr. Hamanaka's and Ms. Ding's datasets at CENPARMI. Both datasets consist of a total of 839 documents.

3.1 Hamanaka's Dataset

The documents in this dataset were scanned at 400 dpi (dots per inch). They were collected by Hamanaka from books, magazines, newspapers, *etc.*

This dataset is pretty comprehensive; it provides the following:

- vertically and horizontally printed Chinese, Japanese, and Korean documents,
- documents in traditional and in simplified Chinese, most of them printed in Song font while others in special fonts like Kai, Black and a few others in Xin Yuan and Big Biao Song fonts,
- documents in the Mincho font and a few others in Gothic and Futo Gothic, and
- documents in *plain* Korean, i.e., containing no Chinese characters as well as others mixed with some Chinese characters. Documents in various fonts are also found, namely in Gyun Myung Jo and Graphic fonts.

	New Myung Jo	Gyun Myung Jo	Graphic	Mixture[1]
Number of Docs	61	2	12	3

Table 1: The composition of the *plain* Korean set.

	New Myung Jo
Number of Korean Docs	30

Table 2: The number of Korean documents containing Chinese characters.

Table 1 shows the distribution of plain Korean documents in New Myung Jo, Gyun Myung Jo, Graphic fonts as well as the documents containing characters in New Myung Jo and Graphic fonts. Table 2 illustrates that no special fonts exist in the Korean set in which some Chinese characters are contained in the Korean documents. Table 3 depicts the composition of the Japanese set which consists of documents in Mincho, Gothic and only one in Futo Gothic. The number of traditional and simplified Chinese documents in Song, Kai, Black, Xin Yuan, Big Biao Song, and a mixture of Song and Kai are found in Table 4.

In Hamanaka's database, there are 108 Korean, 95 Japanese, and 188 Chinese, a total of 391 documents.

[1]These documents contain a mixture of characters in New Myung Jo and Graphic fonts.

	Mincho	Gothic	Futo Gothic
Number of Docs	89	5	1

Table 3: The composition of the Japanese set.

	Song	Kai	Black	Xin Yuan	Big Biao Song	Mixture[2]
No. Traditional Chinese Docs	62	7	4	8	2	0
No. Simplified Chinese Docs	75	18	1	5	0	6

Table 4: The composition of the traditional and simplified Chinese sets.

[2]These documents contain a mixture of characters in Song and Kai fonts.

3.2 Ding's Dataset

The documents in this dataset were scanned at 1200 dpi (dots per inch). Unlike Hamanaka's database which consists of many documents in a multitude of fonts, Ding's database is composed of documents in the most commonly used fonts in all three languages but only two special Chinese Kai and Xin Yuan fonts. Table 5 illustrates the composition of Ding's dataset. In Ding's database, there are 190 Chinese, 94 Japanese, and 164 Korean documents.

3.3 Experimental Results

In this section, we start by showing the performance of the system when tested on Korean, Japanese, and Chinese documents printed in the commonly used font from Hamanaka's database. Furthermore, to show the individual influence of *each* font on the performance of the system, we gradually illustrate the results achieved as documents written in fonts known to the system are introduced to the testing set. Then, we display the behaviour of the system as documents in new fonts, i.e., of which the system does not have models, are added to the testing set. Last, we present the overall results when the complete Hamanaka database is used for testing. We will follow the same path to illustrate the results achieved on Ding's database, presented in sub-section 3.3.3.

	Number of Docs
Korean	164
Japanese	94
Chinese (Song)	168
Chinese (Kai)	21
Chinese (Xin Yuan)	1
Total	**448**

Table 5: The composition of Ding's dataset.

3.3.1 Notations Used in Tables

Throughout all the tables in this chapter, except those that illustrate Ding's results (i.e., Tables 27, 28, 29, and 30), the following notations are used:

- **TOT**: the total number of documents used in the testing phase.
- **REJ**: the number of unclassified and thus rejected documents because no matches were located.
- **OREJ**: the number of unclassified and thus rejected documents because the text Orientation of the document was undetected.
- **CLA**: the number of correctly classified documents.
- **ERR**: the number of misclassified documents.
- **CLA%**: the classification rate.
- **ERR%**: the error rate.
- **REJ%**: the rejection rate.
- **OREJ%**: the rejection rate due to text orientation undetection.
- **RLBTY%**: the reliability of the system. This is the classification rate of the system *including* misclassified documents but *excluding* the rejected ones.
- **OVRL**: short for overall.
- **CHI**: short for Chinese.
- **JAP**: short for Japanese.
- **KOR**: short for Korean.

	TOT	REJ	OREJ	CLA	ERR
OVRL	317	10	1	305	1
CHI	137	6	1	129	1
JAP	89	2	0	87	0
KOR	91	2	0	89	0

	CLA%	ERR%	REJ%	OREJ%	RLBTY%
OVRL	96.21	0.32	3.15	0.32	99.67
CHI	94.16	0.73	4.38	0.73	99.24
JAP	97.75	0.00	2.25	0.00	100.00
KOR	97.80	0.00	2.20	0.00	100.00

Table 6: The performance of the system on documents printed in regular fonts.

3.3.2 Results Achieved on Hamanaka's Database

As depicted in Tables 1 and 2, there are *91* (61 + 30) Korean documents printed in the regular New Myung Jo font. Table 3 shows that *89* Japanese documents are printed in the commonly used font, namely the Mincho font, and from Table 4, we notice that there are *137* (62 + 75) Chinese documents printed in the Song font. Table 6 illustrates the performance of the system when tested on the *317* (91 + 89 + 137) documents mentioned above.

It can be noticed that, of all *317* documents, the text orientation of only one horizontal Chinese document could not be detected by the system simply because the number of white pixels in its vertical projection and its horizontal one were the same, as was explained in sub-section 2.2.2 in Chapter 2. In addition, only one traditional horizontal Chinese document was misclassified as Japanese. No Japanese or Korean documents were misclassified. However, 2 Japanese and 2 Korean ones were rejected by the system.

	TOT	REJ	OREJ	CLA	ERR
OVRL	319	10	1	307	1
CHI	137	6	1	129	1
JAP	89	2	0	87	0
KOR	**91**	**2**	**0**	**91**	**0**

	CLA%	ERR%	REJ%	OREJ%	RLBTY%
OVRL	96.24	0.31	3.13	0.31	99.68
CHI	94.16	0.73	4.38	0.73	99.24
JAP	97.75	0.00	2.25	0.00	100.00
KOR	**97.85**	**0.00**	**2.15**	**0.00**	**100.00**

Table 7: The performance of the system when 2 Korean documents in Gyun Myung Jo font are added to the testing set.

The System's Performance on the Korean Dataset

Table 7 illustrates the results as *2* documents printed in the Gyun Myung Jo font are added to the testing set, resulting in *93* (91 + 2) Korean documents. The 2 added documents were correctly classified and the same 2 Korean documents shown in Table 6 were still rejected.

Table 8 shows the results of adding *12* Korean documents printed in the Graphic font. Please note that the 2 documents in Gyun Myung Jo font added earlier are now removed so that we measure the influence of each font on the system. Hereafter, as documents in a new special font are added, the ones corresponding to the font added previously will be removed from the testing set. Table 8 shows that, in addition to the 2 rejected Korean documents printed in regular font (please refer to Table 6), 6 out of the 12 new documents in Graphic font were rejected because the system found only 1 or 2 Korean matches in them, 3 were misclassified as Chinese, which leaves out 3 correctly classified ones.

	TOT	REJ	OREJ	CLA	ERR
OVRL	329	16	1	308	4
CHI	137	6	1	129	1
JAP	89	2	0	87	0
KOR	**103**	**8**	**0**	**92**	**3**

	CLA%	ERR%	REJ%	OREJ%	RLBTY%
OVRL	93.62	1.22	4.86	0.30	98.72
CHI	94.16	0.73	4.38	0.73	99.24
JAP	97.75	0.00	2.25	0.00	100.00
KOR	**89.32**	**2.91**	**7.77**	**0.00**	**96.84**

Table 8: The performance of the system when 12 Korean documents in Graphic font are added to the testing set. The 2 documents in Gyun Myung Jo are now removed from the set.

Table 9 depicts the results of the system when 3 documents whose contents are mixed with New Myung Jo and Graphic fonts. This results in *94* (91 + 3) Korean documents to be presented for classification. All 3 documents were correctly classified as Korean.

The System's Performance on the Japanese Dataset

As shown in Table 3, there are 89 Japanese documents printed in the regular Mincho font. The performance of the system on those 89 documents is depicted in Table 6. Currently, 5 Japanese documents in the Gothic font will be introduced to the testing set, resulting in 94 (89 + 5) documents to be identified. The results are depicted in Table 10 where it is shown that of the 5 Gothic documents, only one was rejected because the system found only one Japanese character in it. The other 2 rejected

	TOT	REJ	OREJ	CLA	ERR
OVRL	320	10	1	308	1
CHI	137	6	1	129	1
JAP	89	2	0	87	0
KOR	**94**	**2**	**0**	**92**	**0**

	CLA%	ERR%	REJ%	OREJ%	RLBTY%
OVRL	96.25	0.31	3.12	0.31	99.68
CHI	94.16	0.73	4.38	0.73	99.24
JAP	97.75	0.00	2.25	0.00	100.00
KOR	**97.87**	**2.13**	**0.00**	**0.00**	**100.00**

Table 9: The performance of the system when 3 Korean documents in mixed New Myung Jo and Graphic fonts are added to the testing set. The 12 documents in Graphic font are now removed from the set.

documents are in regular fonts as can be noticed from Table 6.

We will next add to the testing set one Japanese document whose font, called Futo Gothic, is not one on which the system was trained. Table 11 illustrates that this single document got rejected because no matches could be produced in it due to its thick characters. That document is depicted in Figure 32 which shows that this font differs significantly from the Mincho and the Gothic fonts on which the system was trained.

	TOT	REJ	OREJ	CLA	ERR
OVRL	322	11	1	309	1
CHI	137	6	1	129	1
JAP	**94**	**3**	**0**	**91**	**0**
KOR	91	2	0	89	0

	CLA%	ERR%	REJ%	OREJ%	RLBTY%
OVRL	95.96	0.31	3.42	0.31	99.68
CHI	94.16	0.73	4.38	0.73	99.24
JAP	**96.81**	**0.00**	**3.19**	**0.00**	**100.00**
KOR	97.80	0.00	0.00	2.20	100.00

Table 10: The performance of the system when 5 Japanese documents in Gothic font are introduced to the testing set. The number of Korean documents is back to 91.

	TOT	REJ	OREJ	CLA	ERR
OVRL	318	11	1	305	1
CHI	137	6	1	129	1
JAP	**90**	**3**	**0**	**87**	**0**
KOR	91	2	0	89	0

	CLA%	ERR%	REJ%	OREJ%	RLBTY%
OVRL	95.91	0.31	3.46	0.31	99.67
CHI	94.16	0.73	4.38	0.73	99.24
JAP	**96.67**	**0.00**	**3.33**	**0.00**	**100.00**
KOR	97.80	0.00	0.00	2.20	100.00

Table 11: The performance of the system when 1 Japanese document in Futo Gothic font is introduced to the testing set. The 5 Gothic fonts are now removed from the testing set.

	TOT	REJ	OREJ	CLA	ERR
OVRL	322	10	1	310	1
CHI	**142**	**6**	**1**	**134**	**1**
JAP	89	2	0	87	0
KOR	91	2	0	89	0

	CLA%	ERR%	REJ%	OREJ%	RLBTY%
OVRL	96.27	0.31	3.11	0.31	99.68
CHI	**94.37**	**0.70**	**4.23**	**0.70**	**99.26**
JAP	97.75	0.00	2.25	0.00	100.00
KOR	97.80	0.00	0.00	2.20	100.00

Table 12: The performance of the system when 5 Chinese documents in Black font are introduced to the testing set. The Japanese fonts are now removed from the testing set.

The System's Performance on the Chinese Dataset

Here, let us recall that the system's performance on the *137* Chinese documents in regular font illustrated in Table 6. After augmenting the testing set by 5 Chinese documents printed in the Black font, the system performed as demonstrated by Table 12. All 5 Black Chinese documents were correctly classified and the 6 rejected documents are the same ones printed in regular font shown previously in Table 6. The other rejected document is the one whose text orientation could not be detected.

After removing the Chinese documents in Black font, we add to the testing set *25* documents in Kai font. The results are given by Table 13. Of the 25 Kai documents, only one was rejected because the system found only one Chinese match. No Kai documents were misclassified.

As far as the *6* documents that contain a mixture of characters in Song and Kai

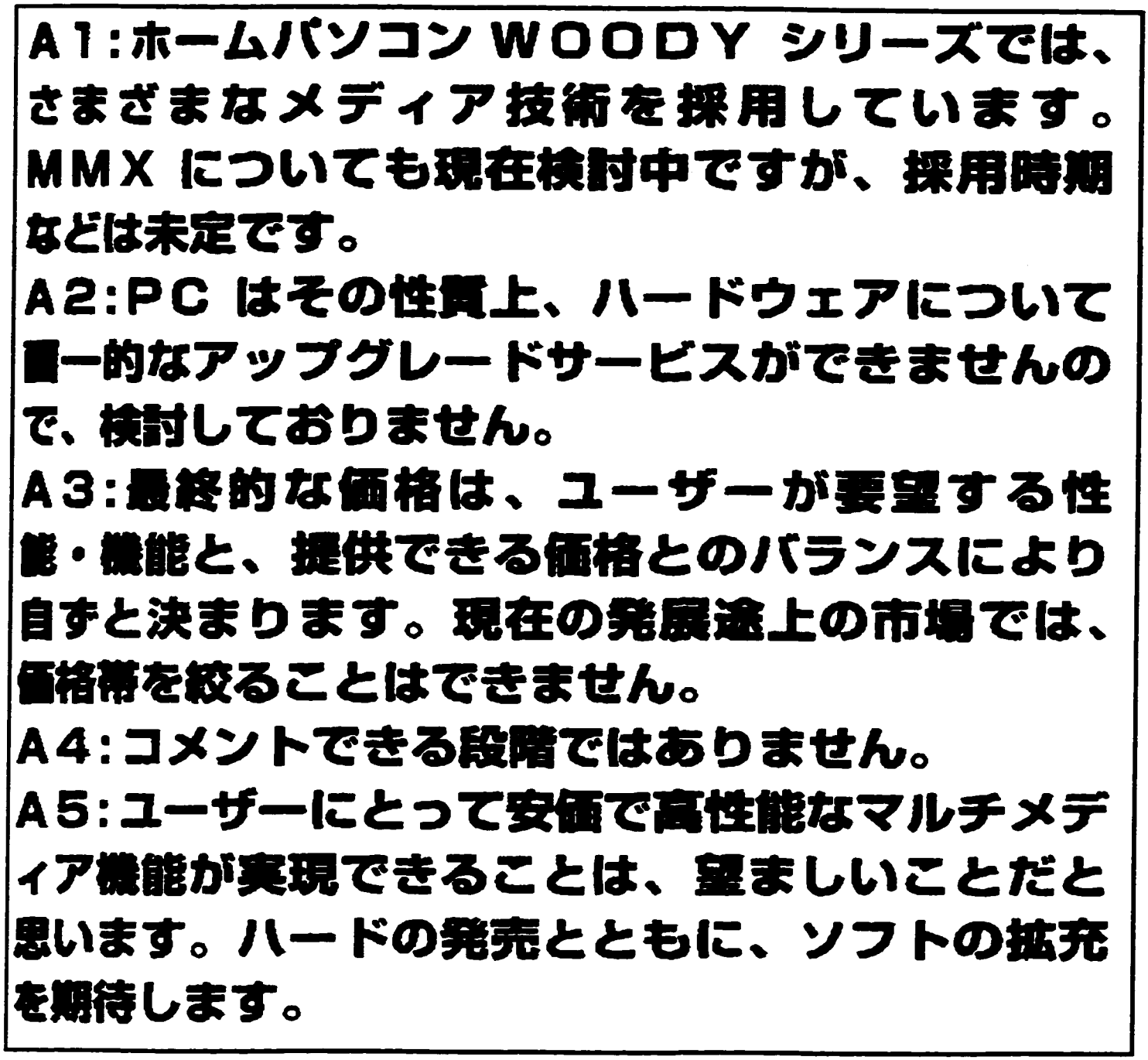

Figure 32: A sample Japanese document printed in Futo Gothic font on which the system was not trained.

	TOT	REJ	OREJ	CLA	ERR
OVRL	342	11	1	329	1
CHI	**162**	**7**	**1**	**153**	**1**
JAP	89	2	0	87	0
KOR	91	2	0	89	0

	CLA%	ERR%	REJ%	OREJ%	RLBTY%
OVRL	96.20	0.29	3.22	0.29	99.70
CHI	**94.44**	**0.62**	**4.32**	**0.62**	**99.35**
JAP	97.75	0.00	2.25	0.00	100.00
KOR	97.80	0.00	0.00	2.20	100.00

Table 13: The performance of the system when 25 Chinese documents in Kai font are introduced to the testing set.

are concerned, the system classified all of them correctly. The results of the mixed documents are presented in Table 14.

The Korean Gyun Myung Jo and Graphic, the Japanese Gothic, the Chinese Kai and Black are all fonts on which the system has been trained. Table 11 shows that the Japanese document in the new Futo Gothic font was rejected. This is because the font differs considerably from Japanese ones known to the system (Futo Gothic is a bold font with very wide strokes). It is simply a font on which the system was not trained. In this database, there are no Korean documents printed in new fonts. However, there are *13* Chinese documents in Xin Yuan font and only *2* others in Biao Song font. Figures 33 and 34 show a Chinese document in Xin Yuan and Biao Song, respectively. Tables 15 and 16 illustrate the behaviour of the system when documents in Xin Yuan and Biao Song are fed to it. For the 13 Xin Yuan documents, only one got rejected because the system found only one match in it. The other 12 were correctly classified since the font is not too different from the other Chinese ones.

	TOT	REJ	OREJ	CLA	ERR
OVRL	323	10	1	311	1
CHI	**143**	**6**	**1**	**135**	**1**
JAP	89	2	0	87	0
KOR	91	2	0	89	0

	CLA%	ERR%	REJ%	OREJ%	RLBTY%
OVRL	96.28	0.31	3.10	0.31	99.68
CHI	**94.41**	**0.70**	**4.20**	**0.70**	**99.27**
JAP	97.75	0.00	2.25	0.00	100.00
KOR	97.80	0.00	0.00	2.20	100.00

Table 14: The performance of the system when 6 Chinese documents containing a mixture of Song and Kai fonts are added to the testing set.

The overall system's performance

Table 17 presents the behaviour of the system as it processes all *391* documents in Hamanaka's database.

Table 18 sheds some light on the effect of introducing commonly used as well as new fonts to the performance of the system in terms of number of documents while Table 19 shows the rates in percentage before and after the same fonts as well.

The confusion matrix resulting from Hamanaka's database is depicted in Table 20.

	TOT	REJ	OREJ	CLA	ERR
OVRL	330	11	1	317	1
CHI	**150**	**7**	**1**	**141**	**1**
JAP	89	2	0	87	0
KOR	91	2	0	89	0

	CLA%	ERR%	REJ%	OREJ%	RLBTY%
OVRL	96.06	0.30	3.33	0.30	99.69
CHI	**94.00**	**0.67**	**4.67**	**0.67**	**99.30**
JAP	97.75	0.00	2.25	0.00	100.00
KOR	97.80	0.00	0.00	2.20	100.00

Table 15: The performance of the system when 13 Chinese documents in a new font called Xin Yuan are added to the testing set.

	TOT	REJ	OREJ	CLA	ERR
OVRL	319	11	1	306	1
CHI	**139**	**7**	**1**	**130**	**1**
JAP	89	2	0	87	0
KOR	91	2	0	89	0

	CLA%	ERR%	REJ%	OREJ%	RLBTY%
OVRL	95.92	0.31	3.45	0.31	99.68
CHI	**93.53**	**0.72**	**5.04**	**0.72**	**99.24**
JAP	97.75	0.00	2.25	0.00	100.00
KOR	97.80	0.00	0.00	2.20	100.00

Table 16: The performance of the system when 2 Chinese documents in a new font called Biao Song are added to the testing set.

國內現時民用航空公司相繼成立，一票難求的情況已有所改善。機票一般預售15天，早3、4天前預訂，應無問題。但遇上特殊情況，例如潑水節期間，昆明至思茅的機位就會十分緊張，且會加價，欲乘搭飛機便要及早安排了。

在香港的中旅社，和中國航空公司(CNAC)，可購買由香港、深圳或廣州出發往全國各地的機票，詳細地址可參閱本章最後附註，另港龍和國泰，亦有航班來往國內部分城市。

至於在國內購買機票可直接到航空公司辦理，機場往往離市區頗遠，但會有小巴、巴士等連接機場和市區民航

Figure 33: A sample Chinese document printed in Xin Yuan font on which the system was not trained.

第十六問　能不能用和平手段廢除私有制？

答：但願如此。共產黨人絕不會阻止使用和平方法。共產黨人十分清楚謀反活動不但無益，反而有害。他們十分清楚革命不是任何人可以隨心所欲地製造的，無論何時何地，革命都是各種因素作用的必然結果，不以個別政黨或整個階級的意志和領導爲轉移。但他們也認識到，幾乎所有文明國家的無產階級的發展都受到强烈的壓制，共產主義的敵人這樣做無異於竭盡全力引發革命。一旦受壓迫的無產者最終被推向革命，我們共產主義者將毫不猶豫用實際行動捍衛無產階級事業，就像我們現在用語言捍衛它一樣。

Figure 34: A sample Chinese document printed in Biao Song font on which the system was not trained.

	TOT	REJ	OREJ	CLA	ERR
OVRL	391	21	1	365	4
CHI	188	9	1	177	1
JAP	95	4	0	91	0
KOR	108	8	0	97	3

	CLA%	ERR%	REJ%	OREJ%	RLBTY%
OVRL	93.35	1.02	5.37	0.26	98.92
CHI	94.15	0.53	4.79	0.53	99.44
JAP	95.79	0.00	4.21	0.00	100.00
KOR	89.81	2.78	7.41	0.00	97.00

Table 17: The performance of the system on all documents from Hamanaka's database.

		TOT	REJ	OREJ	CLA	ERR
OVRL	Before Fonts	317	10	1	305	1
	After Fonts	391	21	1	365	4
CHI	Before Fonts	137	6	1	129	1
	After Fonts	188	9	1	177	1
JAP	Before Fonts	89	2	0	87	0
	After Fonts	95	4	0	91	0
KOR	Before Fonts	91	2	0	89	0
	After Fonts	108	8	0	97	3

Table 18: The performance (in terms of number of documents) of the system before and after documents in special fonts were introduced to the testing set.

		CLA%	ERR%	REJ%	OREJ%	RLBTY%
OVRL	Before Fonts	96.21	0.32	3.15	0.32	99.67
	After Fonts	93.35	1.02	5.37	0.26	98.92
CHI	Before Fonts	94.16	0.73	4.38	0.73	99.24
	After Fonts	94.15	0.53	4.79	0.53	99.44
JAP	Before Fonts	97.75	0.00	2.25	0.00	100.00
	After Fonts	95.79	0.00	4.21	0.00	100.00
KOR	Before Fonts	97.80	0.00	2.20	0.00	100.00
	After Fonts	89.81	2.78	7.41	0.00	97.00

Table 19: The rates achieved by the system before and after documents in special fonts are introduced to the testing set.

	CHI	JAP	KOR	REJ
CHI	0	1	0	9
JAP	0	0	0	4
KOR	3	0	0	8

Table 20: The confusion matrix resulting from Hamanaka's database.

3.3.3 Results Achieved on Ding's Database

In this sub-section, we will follow the same path as in sub-section 3.3.2 in the sense that we will show the performance of the system first on the documents printed in the regular fonts and then gradually introduce the other fonts on which the system was trained, and terminate by demonstrating the behaviour of the system as documents in new fonts, on which it has not been trained, are added to the testing set.

As depicted in Table 5, there are *164* Korean documents printed in the regular New Myung Jo font, *94* Japanese documents are printed in the commonly used font, namely the Mincho font, and *168* Chinese documents printed in the Song font. Table 21 illustrates the performance of the system when tested on the *426* (164 + 94 + 168) documents printed in the most commonly used fonts. One Chinese document was rejected because it found 2 Japanese matches in it. In such a case, the system rejects, does not misclassify as Japanese, the input document because of only 2 Japanese matches, which is estimated by our method not to be enough for proper classification. The 4 Korean documents were rejected because just 1 or 2 Korean matches were found in them.

The System's Performance on the Chinese Dataset

In Ding's database, only Chinese has documents printed in special fonts, namely, Kai and Xin Yuan. All 94 Japanese and 164 Korean documents are printed in commonly used fonts.

TABLE 22 shows the performance of the system as *21* documents in Kai font are added to the testing set, resulting in 189 (168 + 21) Korean documents. Of the 21 introduced Kai documents, 3 were rejected because only 1 or 2 Chinese matches were found by the system.

Unfortunately, given Ding's database, we cannot test the effect of introducing new

	TOT	REJ	OREJ	CLA	ERR
OVRL	426	5	0	421	0
CHI	168	1	0	167	0
JAP	94	0	0	94	0
KOR	164	4	0	160	0

	CLA%	ERR%	REJ%	OREJ%	RLBTY%
OVRL	98.83	0.00	1.17	0.00	100.00
CHI	99.40	0.00	0.60	0.00	100.00
JAP	100.00	0.00	0.00	0.00	100.00
KOR	97.56	0.00	2.44	0.00	100.00

Table 21: The performance of the system on documents from Ding's database printed in commonly used fonts.

	TOT	REJ	OREJ	CLA	ERR
OVRL	447	8	0	439	0
CHI	**189**	**4**	**0**	**185**	**0**
JAP	94	0	0	94	0
KOR	164	4	0	160	0

	CLA%	ERR%	REJ%	OREJ%	RLBTY%
OVRL	98.21	0.00	1.79	0.00	100.00
CHI	**97.88**	**0.00**	**2.12**	**0.00**	**100.00**
JAP	100.00	0.00	0.00	0.00	100.00
KOR	97.56	0.00	2.44	0.00	100.00

Table 22: The performance of the system as 21 Chinese documents in Kai font from Ding's database are introduced to the testing set.

	TOT	REJ	OREJ	CLA	ERR
OVRL	427	5	0	422	0
CHI	**169**	**1**	**0**	**168**	**0**
JAP	94	0	0	94	0
KOR	164	4	0	160	0

	CLA%	ERR%	REJ%	OREJ%	RLBTY%
OVRL	98.23	0.00	1.17	0.00	100.00
CHI	**99.41**	**0.00**	**0.59**	**0.00**	**100.00**
JAP	100.00	0.00	0.00	0.00	100.00
KOR	97.56	0.00	2.44	0.00	100.00

Table 23: The performance of the system as 1 Chinese document (from Ding's database) in Xin Yuan font, on which the system was not trained, are introduced to the testing set.

fonts, on which the system was not trained, to the same extent as with Hamanaka's database since there is one single Chinese document in Xin Yuan font. Adding the Xin Yuan document to the testing set generates the results in Table 23.

The overall system's performance

We will now present the behaviour of the system as it processes all *448* documents in Ding's database. Table 24 illustrates.

Since, in Ding's database, documents printed in special fonts can only be found in Chinese, Tables 25 and 26 illustrate the overall performance of the system as well as its performance on the Chinese dataset before and after fonts are introduced to the testing set. No confusion matrix will be displayed for Ding's set because no documents were misclassified in any of the three languages.

	TOT	REJ	OREJ	CLA	ERR
OVRL	448	8	0	440	0
CHI	190	4	0	186	0
JAP	94	0	0	94	0
KOR	164	4	0	160	0

	CLA%	ERR%	REJ%	OREJ%	RLBTY%
OVRL	98.21	0.00	1.79	0.00	100.00
CHI	97.89	0.00	2.11	0.00	100.00
JAP	100.00	0.00	0.00	0.00	100.00
KOR	97.56	0.00	2.44	0.00	100.00

Table 24: The performance of the system on all documents from Ding's dataset.

		TOT	REJ	OREJ	CLA	ERR
OVRL	Before Fonts	426	5	0	421	0
	After Fonts	448	8	0	440	0
CHI	Before Fonts	168	1	0	167	0
	After Fonts	190	4	0	186	0

Table 25: The performance (in terms of number of documents) of the system before and after documents in Chinese special fonts were introduced to the testing set from Ding's database.

		CLA%	ERR%	REJ%	OREJ%	RLBTY%
OVRL	Before Fonts	98.83	0.00	1.17	0.00	100.00
	After Fonts	98.21	0.00	1.79	0.00	100.00
CHI	Before Fonts	99.40	0.00	0.60	0.00	100.00
	After Fonts	97.89	0.00	2.11	0.00	100.00

Table 26: The rates achieved by the system before and after documents in Chinese special fonts from Ding's database are introduced to the testing set.

3.4 Comparison of Methods and Results

Ding's complete database consists of *190 Chinese*, *94 Japanese*, and *164* Korean documents. To train the identification system described in her thesis [Din99], Ms. Ding used *76* Chinese, *45* Japanese, and *58* Korean documents, leaving **114** Chinese, **49** Japanese, and **106** Korean documents for the testing set.

Note that the notations used in Tables 27, 28, 29, 30, which illustrate Ding's results, are according to her thesis work in [Din99]. The notations are intentionally kept as they are to draw the reader's attention that these tables illustrate the results of another work on the same test samples. The tables' layout is also different from the ones that illustrate our work for the same reason.

In Ding's work (previously explained in section 1.3.1), the classification of the test samples is performed by two methods: by using the ranges of C (complex structure), K (Korean circles), and V (long vertical strokes) values for the training set, and by clustering. The K-means clustering algorithm is used to generate cluster centers, or codebooks of the training data. Ding found that complex structure is important for separating Japanese texts from Chinese, long vertical strokes for differentiating Korean from Chinese and Japanese texts, and Korean circles for the separation of Korean from Chinese and Japanese texts [Din99]. Ding obtained the classification

Language	# Samples	Not Processed	Recognition (%)	Error (%)	Rejection (%)
Chinese	114	1	94.69	4.43	0.88
Japanese	49	0	95.92	0.00	4.08
Korean	106	1	93.33	6.67	0.00

Table 27: Ding's results of Oriental language classification by using C, K, and V values.

	Chinese	Japanese	Korean	Reject
Chinese	107	5	0	1
Japanese	0	47	0	2
Korean	0	7	98	0

Table 28: Ding's confusion matrix when using C, K, and V values.

results, shown in Table 27, according to C, K, and V values. Table 28 shows the confusion matrix when using C, K, and V values.

Ding's classification results after applying the clustering algorithm are depicted in Table 29, while the confusion matrix from clustering using C, K, and V features is depicted in Table 30. These are the results that will be used to compare the performance of Ding's system with ours.

When all three features are used for clustering, five Chinese testing samples are classified as Japanese documents. All but one of them are printed in Chinese Kai font. As the strokes in this font are smooth and do not touch each other, there are fewer complex structures in this font, leading to the misclassification of these images unto Japanese. The misclassification of the other sample is due to poor quality and the existence of many broken strokes [Din99].

Language	# Samples	Not Processed	Recognition (%)	Error (%)	Reject (%)
Chinese	114	1	94.69	4.43	0.88
Japanese	49	0	97.96	0.00	2.04
Korean	106	1	97.14	1.91	0.00

Table 29: Ding's results from clustering using C, K, and V features.

	Chinese	Japanese	Korean	Reject
Chinese	107	5	0	1
Japanese	0	48	0	1
Korean	0	2	102	1

Table 30: Ding's confusion matrix from clustering using C, K, and V features.

The Korean documents misclassified as Japanese are due to problems with the detection of Korean circles. Most of the Korean circles in these documents cannot be detected by Ding's circle extraction method because in one document the ellipses look more like rectangles than Korean circles while in another document, there are sharp turns in the inner contour of the circles, which significantly reduce the number of Korean circles [Din99].

While Ding's method is statistically-based, our method is based on template matching. The essence is to look for frequently used templates of each language in a new document. The method was explained in detail in Chapter 2. To provide a common ground for the comparison of our results with Ding's, we have used exactly the same test samples used by Ms. Ding to test the performance of our system. Table 31 shows the results achieved by our system on the same testing set used by Ding in [Din99].

	TOT	REJ	CLA	ERR
CHI	114	2	112	0
JAP	49	0	49	0
KOR	106	4	102	0

	CLA%	ERR%	REJ%	RLBTY%
CHI	98.25	0.00	1.75	100.00
JAP	100.00	0.00	0.00	100.00
KOR	96.23	0.00	3.77	100.00

Table 31: The performance of our system on the same test samples used by Ding in [Din99].

Out of the 114 Chinese samples used, 14 samples are printed in Chinese Kai. 2 of the 14 Kai documents were rejected, the first because only 1 Chinese match was found in it, and the second because 1 Korean match was located in it. 4 Korean documents were also rejected. In 3 of them, the system found only 2 Korean matches, while it found 1 match in the fourth sample. No samples were rejected for Japanese.

3.4.1 Comparison of Results

Table 32 depicts the results of Ding's system as well as ours for comparison purposes. We can draw the following conclusions from it:

- For *Chinese*: our system achieved a classification rate 3.56% higher than Ding's system, a 0.00% error rate, 4.43% lower than Ding's system, and a rejection rate 0.87% higher than Ding's system.
- For *Japanese*: our system achieved a classification rate 2.04% higher than Ding's system, a 0.00% error rate, same as Ding's system, and a rejection rate 2.04% lower than Ding's system.

		CLA%	ERR%	REJ%
CHI	Ding's System	94.69	4.43	0.88
	Our system	98.25	0.00	1.75
JAP	Ding's System	97.96	0.00	2.04
	Our system	100.00	0.00	0.00
KOR	Ding's System	97.14	1.91	0.00
	Our system	96.23	0.00	3.77

Table 32: The classification, error, and rejection rates achieved by Ding's system and by ours on the same test samples.

- For *Korean*: our system achieved a classification rate 0.91% lower than Ding's system, a 0.00% error rate, 1.91% lower than Ding's system, and a rejection rate 3.77% higher than Ding's system.

The merit of our system is that it reported a 0.00% error rate for all three languages, which makes the system 100% reliable. Another merit is that our system processes horizontal and vertical documents, as compared to Ding's system which processes horizontal documents only.

3.5 More Results

The purpose of this section is to show the overall results of using 5 or 7 matches to classify a document in comparison with 3 matches. It also intends to present the length in terms of number of characters in the documents and then the classification and error rates of short, medium, and long passages as 3, 5, or 7 matches are used for classification. As a reminder, 3, 5, or 7 does not refer to the number of templates used in the search but to the minimum number of matches that needs to be located by the system for the new document to be classified.

	Dataset	0-125	126-250	251-500	501-
CHI	Ding	3	33	147	7
	Hamanaka	4	59	119	6
JAP	Ding	18	27	14	35
	Hamanaka	2	13	74	6
KOR	Ding	2	6	29	127
	Hamanaka	0	31	74	3

Table 33: The number of documents in Ding's and Hamanaka's datasets in each of the 4 categories.

We divided Ding's and Hamanaka's datasets into the following 4 categories, depending on the number of characters found in the documents:

- Category 1: between *0* and *125* characters
- Category 2: between *126* and *250* characters
- Category 3: between *251* and *500* characters
- Category 4: *501* characters and more

Figures 35 and 36 depict the overall classification rates of the system as 3, 5, or 7 matches are used for the classification of Chinese, Japanese, and Korean on Hamanaka's and Ding's datasets, respectively. It can be noticed from both Figures that the classification rates for all 3 languages drop as the number of matches increases from 3 to 5 and finally to 7, except for Japanese in Figure 36 because 1 document, which was correctly classified with 3 matches, got rejected with 5 matches and thus with 7 matches, which is why the classification rates for 5 and 7 matches for Japanese are the same in Ding's dataset. The drop of classification rates is expected since the rejection rates increase as we move in the same direction. On the other hand, some of those documents that used to be misclassified become rejected by the system as

the number of matches increases, which will possibly cause the error rates to drop as well, as illustrated by Figure 37 which shows the overall error rates obtained on Hamanaka's dataset. The error rate for Japanese is 0% for 3, 5, and 7 matches. For Chinese, with 3 matches, there was only 1 misclassified document that was later rejected with 5 matches and thus with 7 matches. This justifies the 0% error rate with 5 and 7 matches. No similar graph is available for Ding's dataset since the error rates for all 3 language are 0%. Figures 38 and 39 illustrate the overall reliability of the system on Hamanaka's and Ding's datatsets, respectively, as 3, 5, and 7 matches are used for classification.

Let's now examine the classification and error rates for the documents found in the 4 categories mentioned previously. Only in the case where error rates are observed in a certain category that a graph will be presented for illustration purposes.

3.5.1 More Results on Hamanaka's Dataset

Category 1: between 0 and 125 characters

Figure 40 shows the classification rates of 4 Chinese and 2 Japanese in the first category of Hamanaka's dataset. The classification rate of Japanese dropped to 0% in this category since the 2 Japanese documents that were correctly classified with 3 matches got rejected by the system with 5 matches and thus with 7 matches and since there are only 2 Japanese documents in this category, the classification rate would be 0% for 5 as well as for 7 matches. For Chinese, among the 4 available documents, 1 was rejected with 3 matches because its text orientation could not be detected by the system. The other 3 were properly classified. With 5 matches, 2 other documents got rejected, and thus only 1 classified and finally, with 7 matches, the document that was classified with 5 matches got rejected with 7 matches, which causes the Chinese classification rate to be 0% with 7 matches.

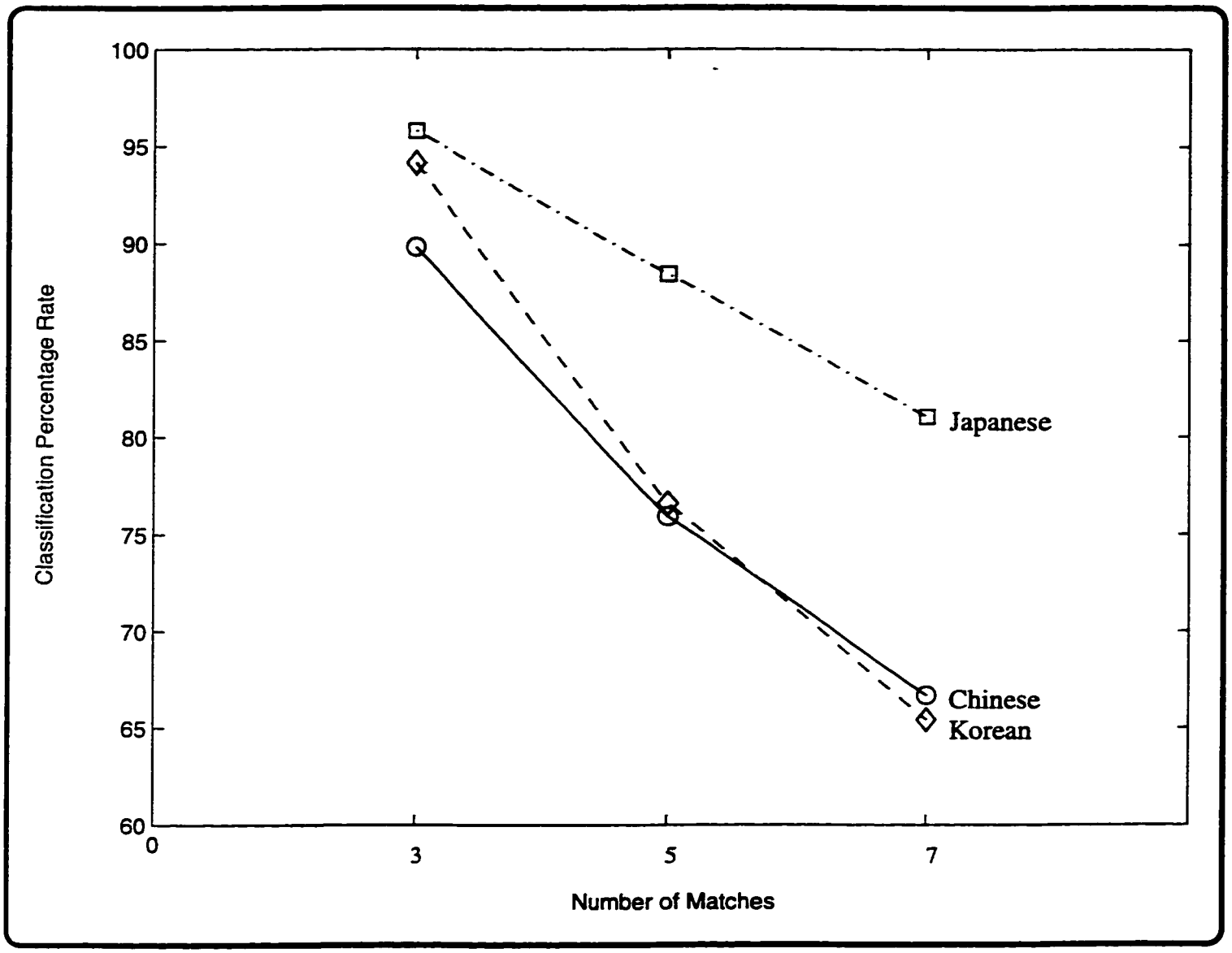

Figure 35: The overall classification rates of Chinese, Japanese, and Korean in Hamanaka's dataset for 3, 5, and 7 matches.

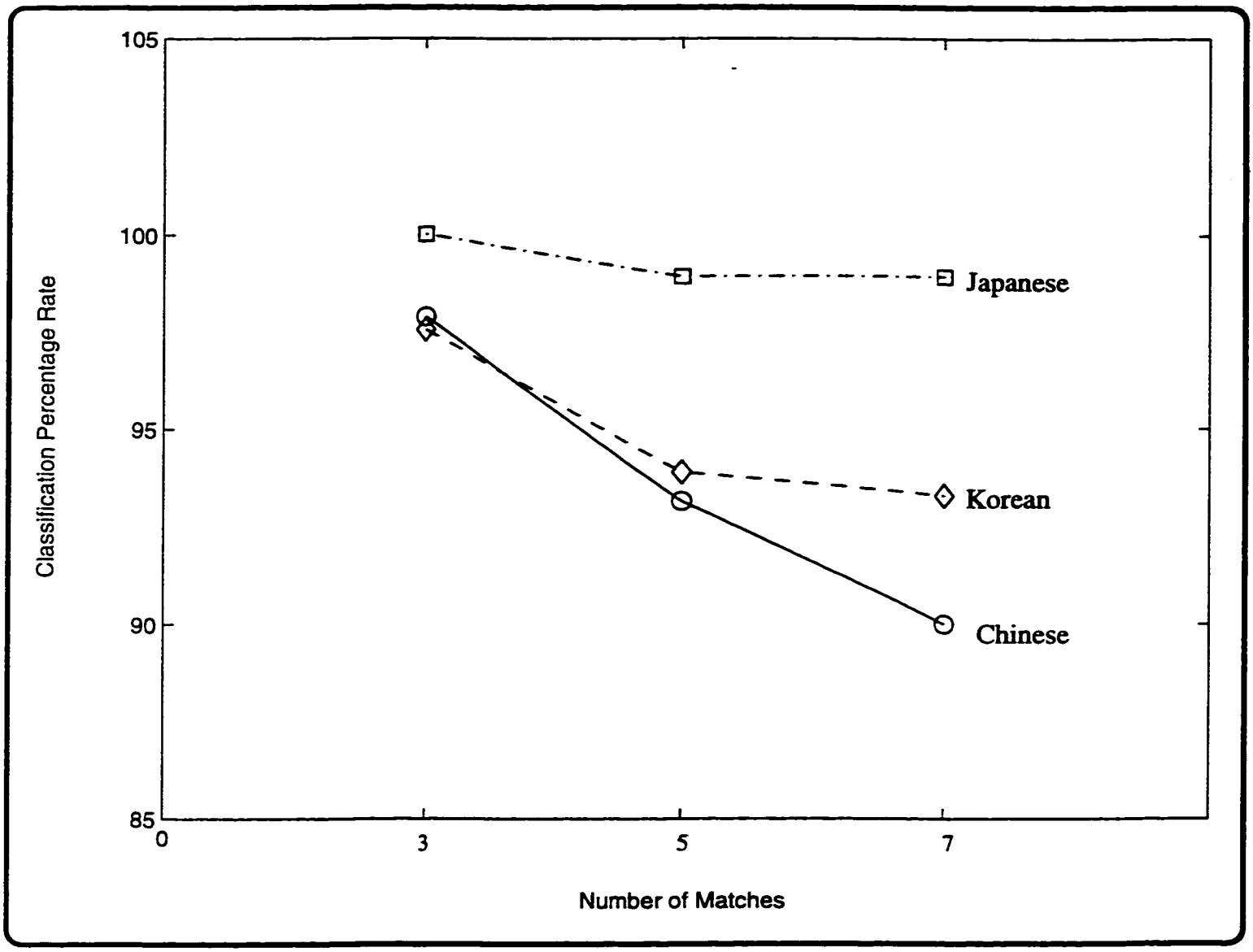

Figure 36: The overall classification rates of Chinese, Japanese, and Korean documents in Ding's dataset with 3, 5, and 7 matches.

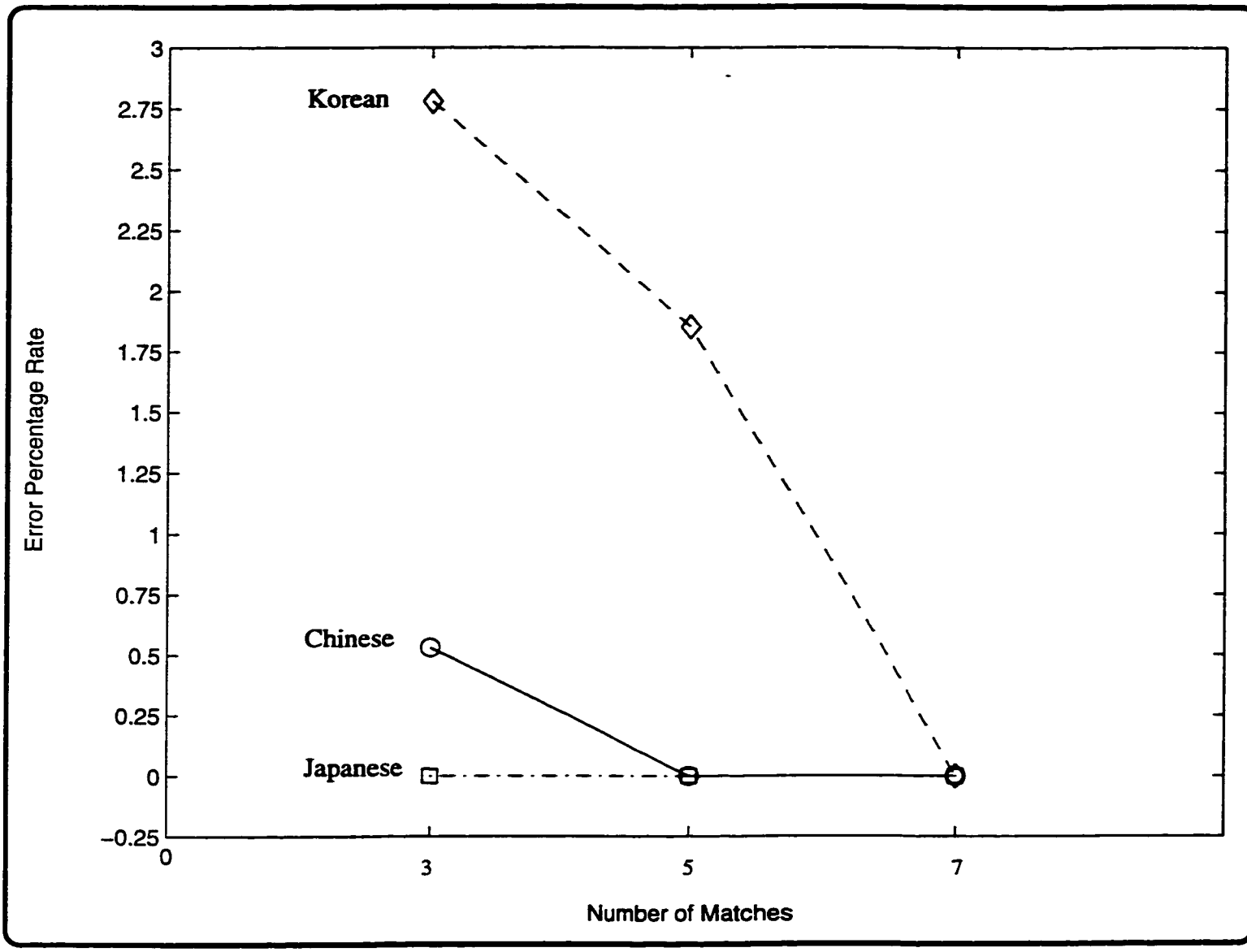

Figure 37: The overall error rates of Chinese, Japanese, and Korean documents in Hamanaka's dataset with 3, 5, and 7 matches.

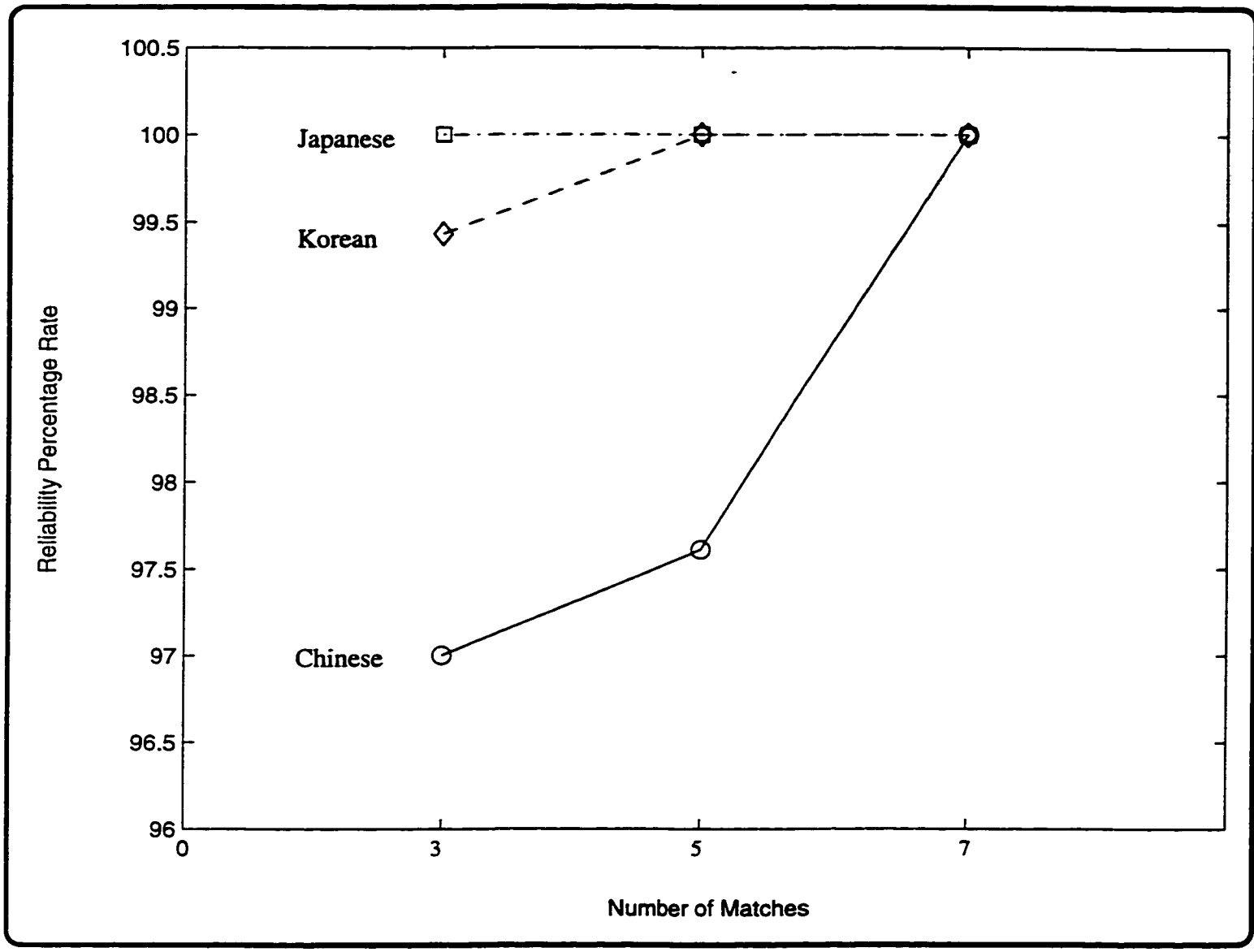

Figure 38: The overall reliability rates of Chinese, Japanese, and Korean documents in Hamanaka's dataset with 3, 5, and 7 matches.

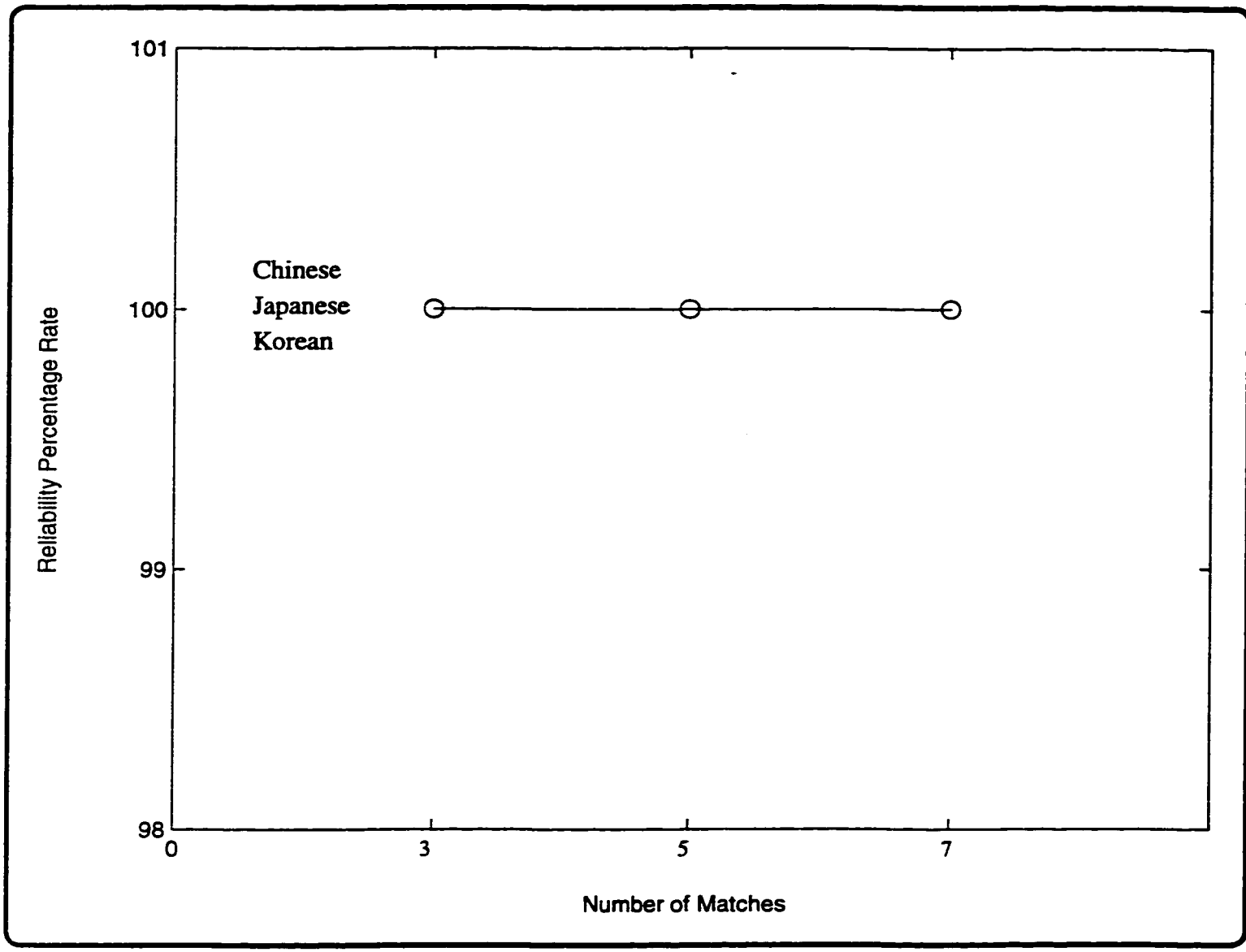

Figure 39: The overall reliability rates of Chinese, Japanese, and Korean documents in Ding's dataset with 3, 5, and 7 matches.

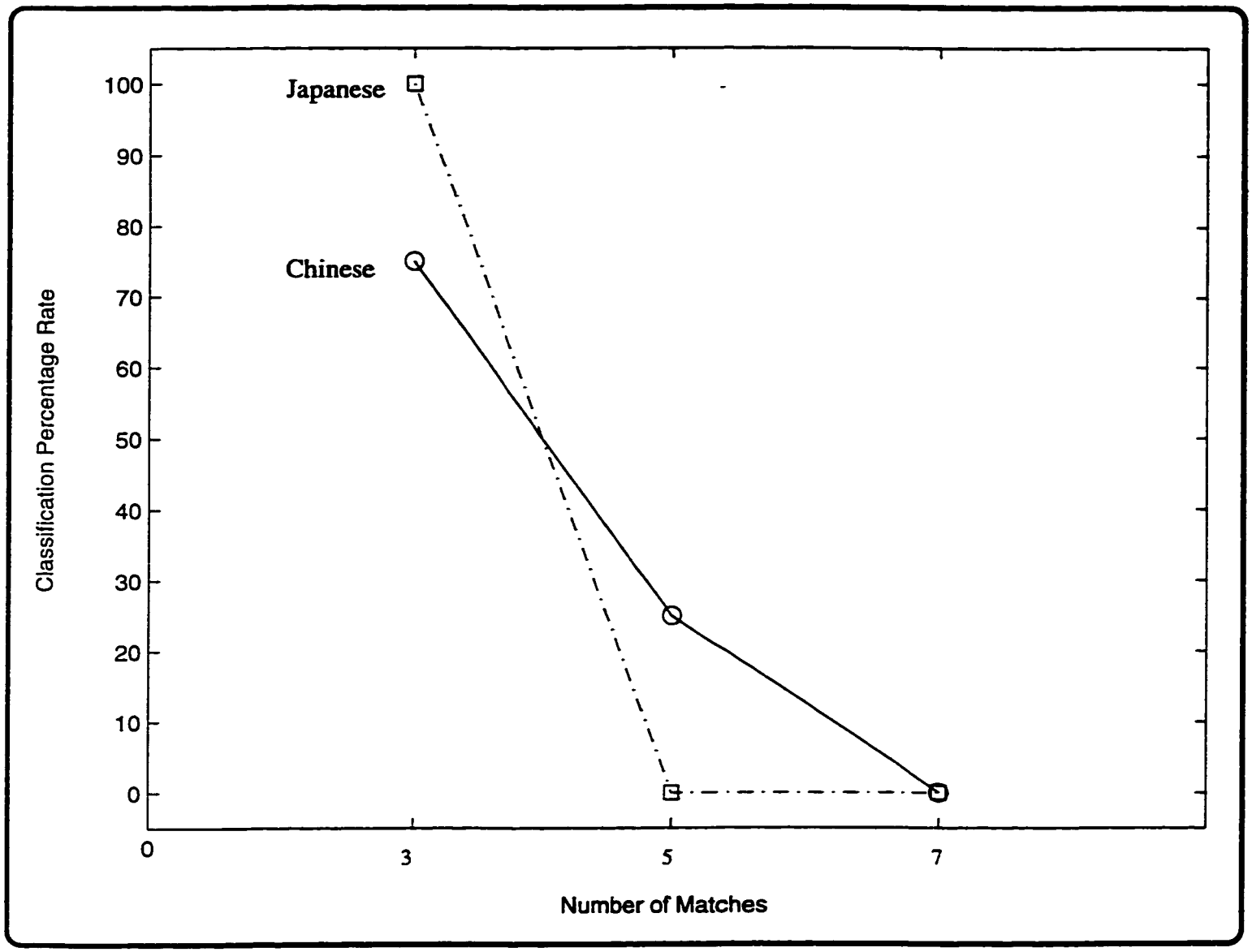

Figure 40: The classification rates of Chinese and Japanese documents in category 1 of Hamanaka's dataset with 3, 5, and 7 matches. There are no Korean documents in this category.

Category 2: between 126 and 250 characters

Figure 41 shows the classification rates of 59 Chinese, 13 Japanese, and 31 Korean documents in the second category of Hamanaka's dataset. We notice a normal decrease in the classification rates in all 3 languages. Figure 42 depicts the error rates of Chinese, Japanese, and Korean. The Chinese and the Japanese error rates with 3, 5, and 7 matches are all 0% since no Chinese and Japanese documents, only Korean, were misclassified in this category. With 3 matches, 2 Korean documents were misclassified, with 5 matches, one of the 2 misclassified got rejected and the other one still misclassified, and with 7 matches, the document that was misclassified with 5 matches got rejected, which makes the Korean error rate 0% with 7 matches.

Category 3: between 251 and 500 characters

Figure 43 shows the classification rates of 119 Chinese, 74 Japanese, and 74 Korean documents in the third category of Hamanaka's dataset. Figure 44 shows the error rates of Chinese, Japanese, and Korean documents in this category. The Japanese error rate is 0% since no Japanese documents were misclassified. The Korean error rate with 3 matches is the same as the one with 5 matches because no additional documents were misclassified as the number of matches was increased from 3 to 5. With 7 matches, the Korean error rate dropped to 0% because the Korean document that was misclassified with 5 matches got rejected with 7 matches, which leaves no misclassified documents. On the other hand, the Chinese error rates with 5 and 7 matches are both 0% since the only misclassified Chinese document was rejected by the system with 5 matches and thus with 7 matches causing the Chinese error rates to be 0% with 5 and 7 matches.

Category 4: 501 characters and more

Figure 45 illustrates the classification rates of 6 Chinese, 6 Japanese, and 3 Korean documents in the fourth category of Hamanaka's dataset. The Chinese, Japanese, and Korean classification rates are all 100% with 3 and 5 matches. The Japanese and the Korean error rates dropped with 7 matches while Chinese maintained the same

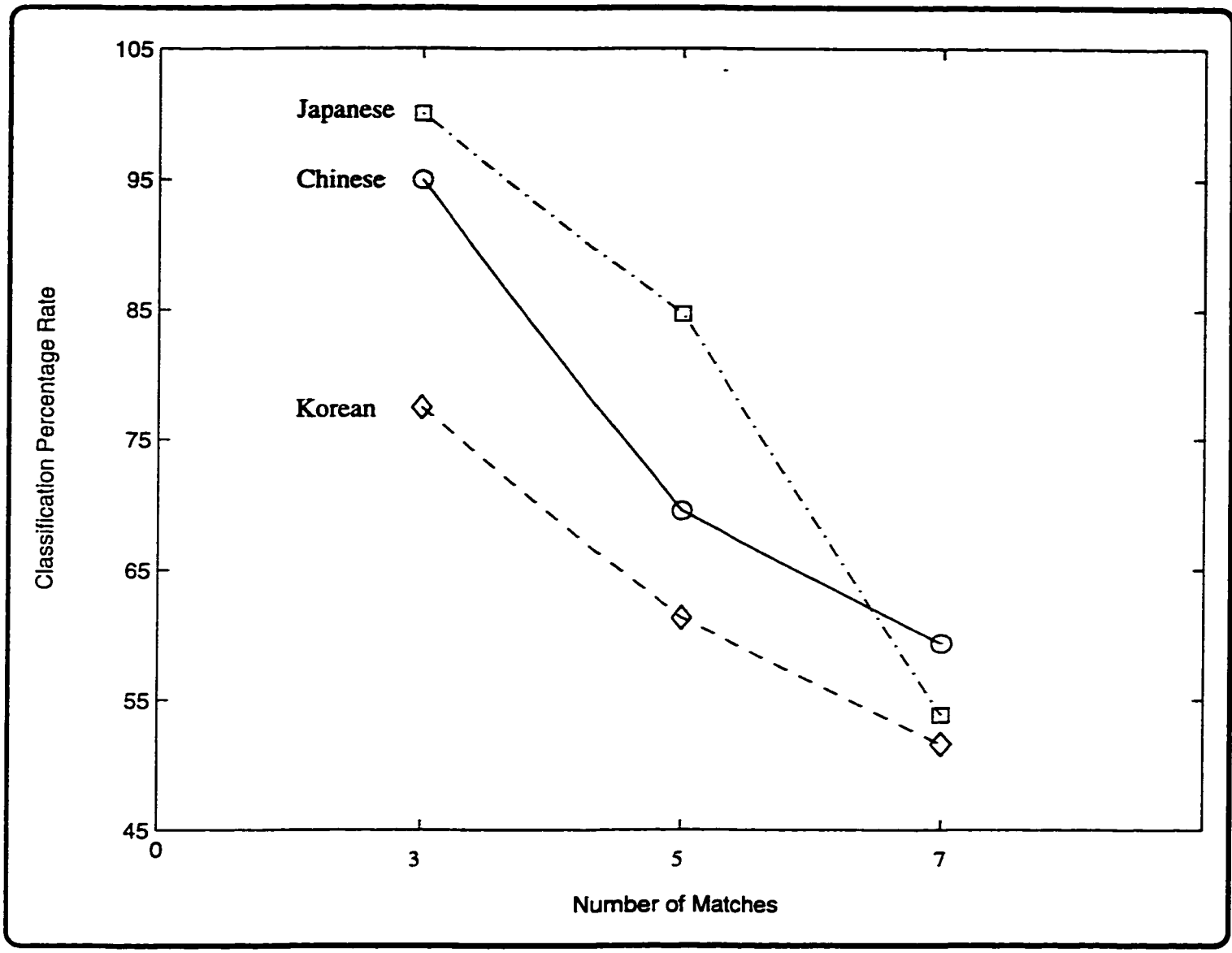

Figure 41: The classification rates of Chinese, Japanese, and Korean documents in category 2 of Hamanaka's dataset with 3, 5, and 7 matches.

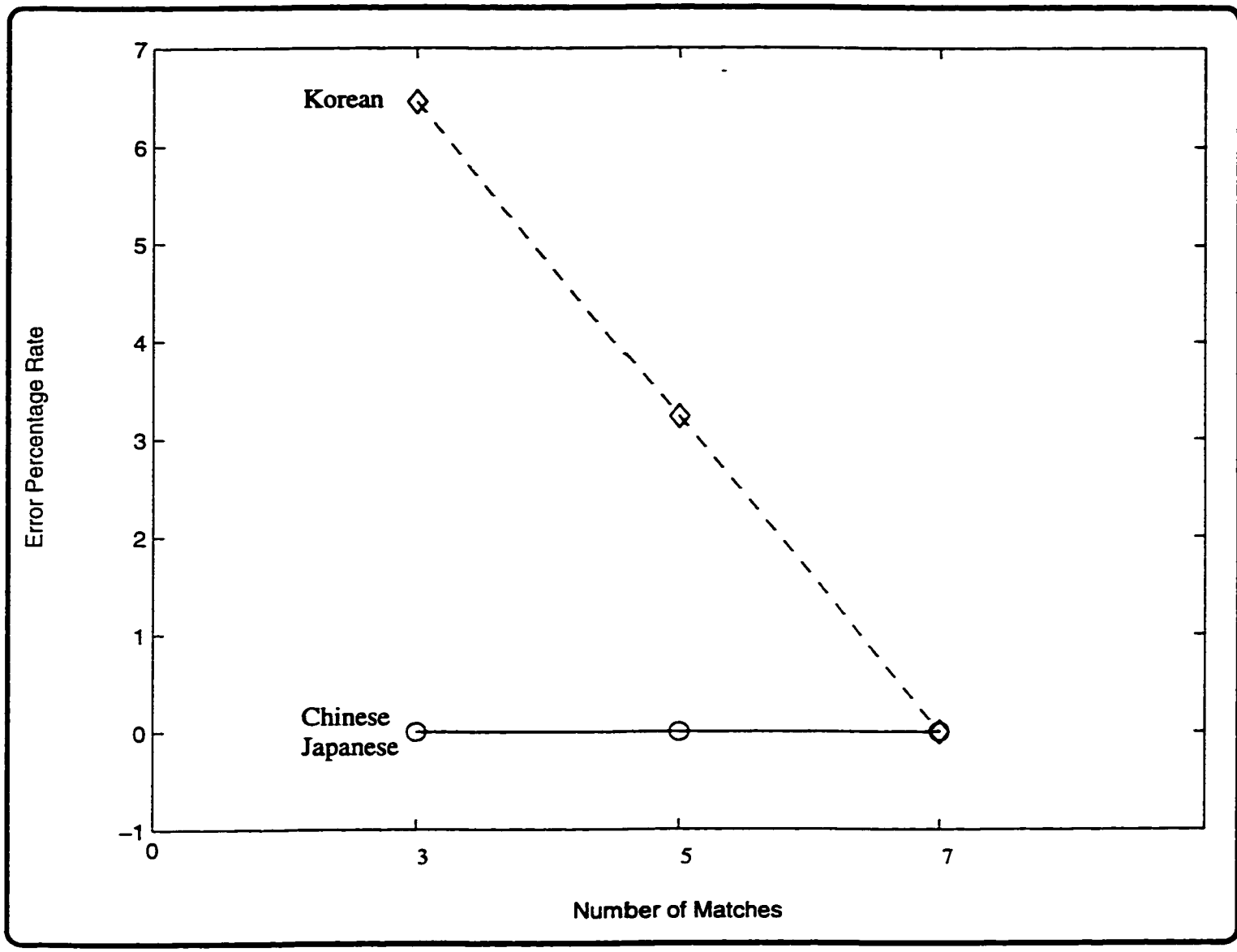

Figure 42: The error rates of Chinese, Japanese, and Korean documents in category 2 of Hamanaka's dataset with 3, 5, and 7 matches.

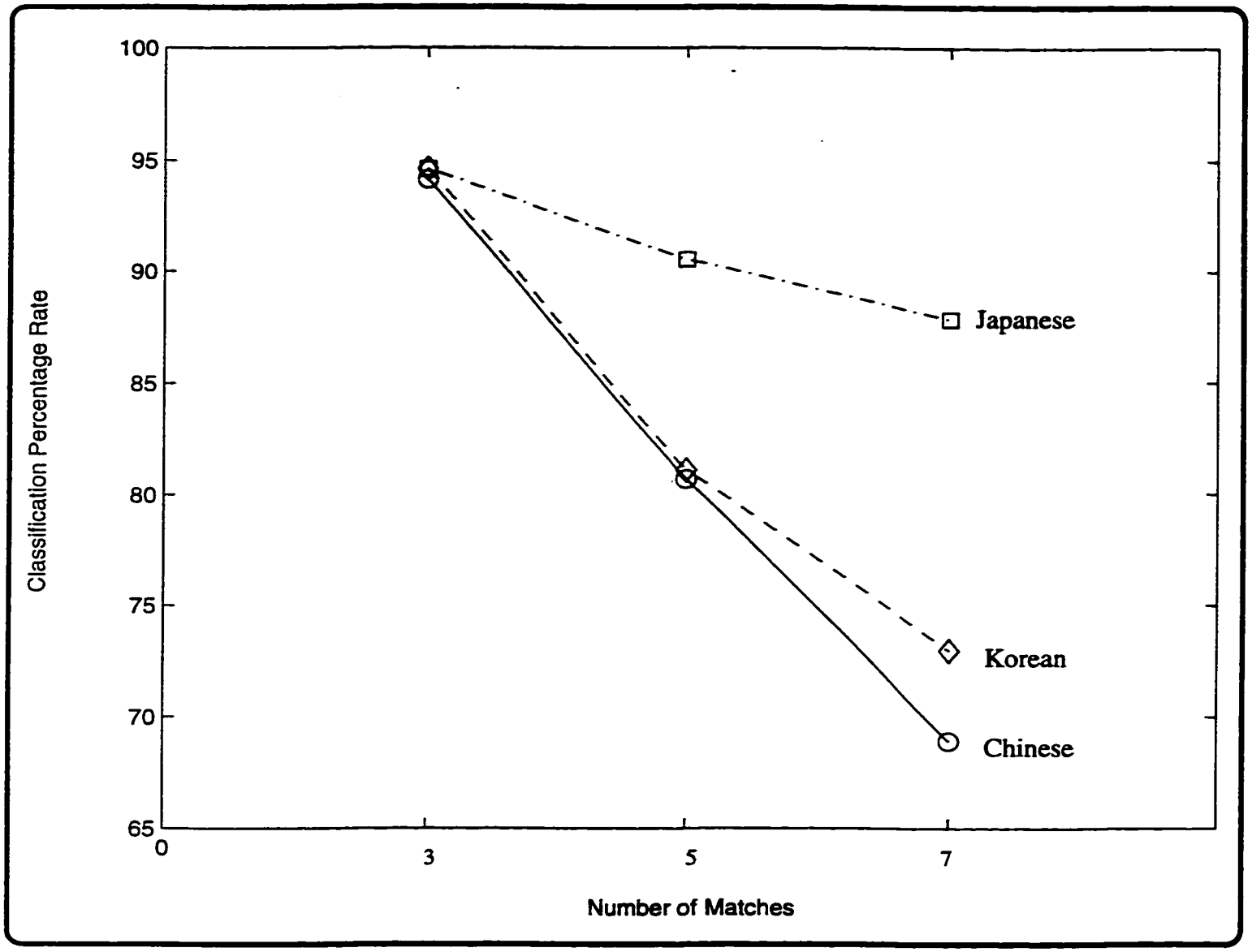

Figure 43: The classification rates of Chinese, Japanese, and Korean documents in category 3 of Hamanaka's dataset with 3, 5, and 7 matches.

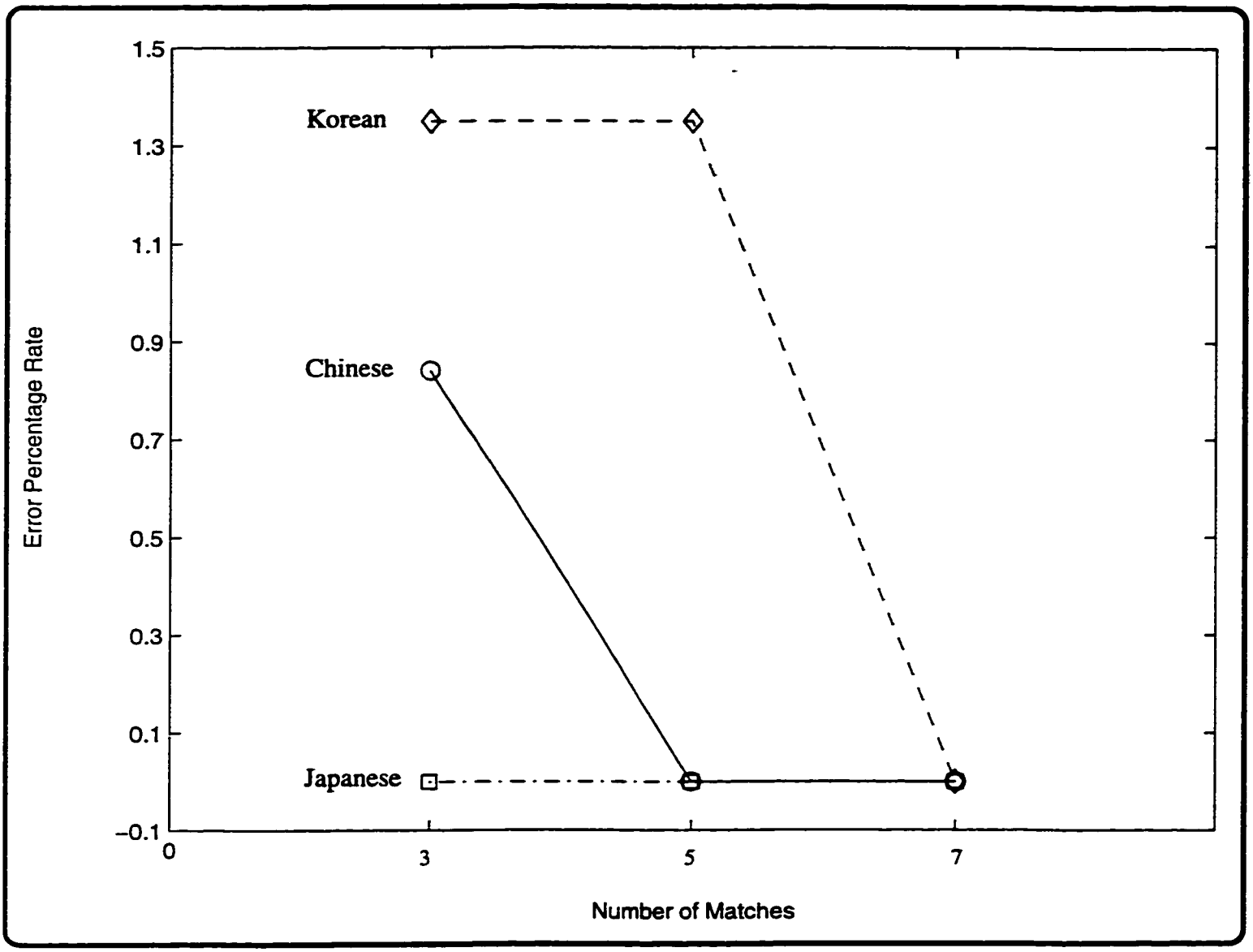

Figure 44: The error rates of Chinese, Japanese, and Korean documents in category 3 of Hamanaka's dataset with 3, 5, and 7 matches.

classification rate of 100% in this category.

3.5.2 More Results on Ding's Dataset

Since the Chinese, Japanese, and Korean error rates obtained on Ding's dataset are all 0% for 3, 5, and 7 matches, no graphs representing error rates will be presented in this sub-section.

Category 1: between 0 and 125 characters

The classification rates of 3 Chinese, 18 Japanese, and 2 Korean documents in the first category of Ding's dataset are presented in Figure 46. The Chinese and Japanese classification rates remained 100% with 5 and 7 matches while the Korean classification rate dropped to 0% because the only 2 Korean documents in this category were correctly classified with 3 matches and later rejected with 5 and thus 7 matches causing the classification rate to drop to 0% with 5 and 7 matches.

Category 2: between 126 and 250 characters

Figure 47 shows the classification rates of 33 Chinese, 27 Japanese, and 6 Korean documents in the second category of Ding's dataset. Here, a normal decrease of the Chinese classification rates can be noticed. As far as Japanese is concerned, with 3 matches, all documents were correctly classified. With 5 matches, 1 document got rejected and later with 7 matches, no other documents were rejected, which causes the classification rates with 5 and 7 matches to be the same. The same applies to Korean.

Category 3: between 251 and 500 characters

Figure 48 illustrates the classification rates of 147 Chinese, 14 Japanese, and 29 Korean documents in the third category of Ding's dataset. The Chinese and Korean classification rates dropped with 5 and 7 matches while Japanese maintained its level at 100%.

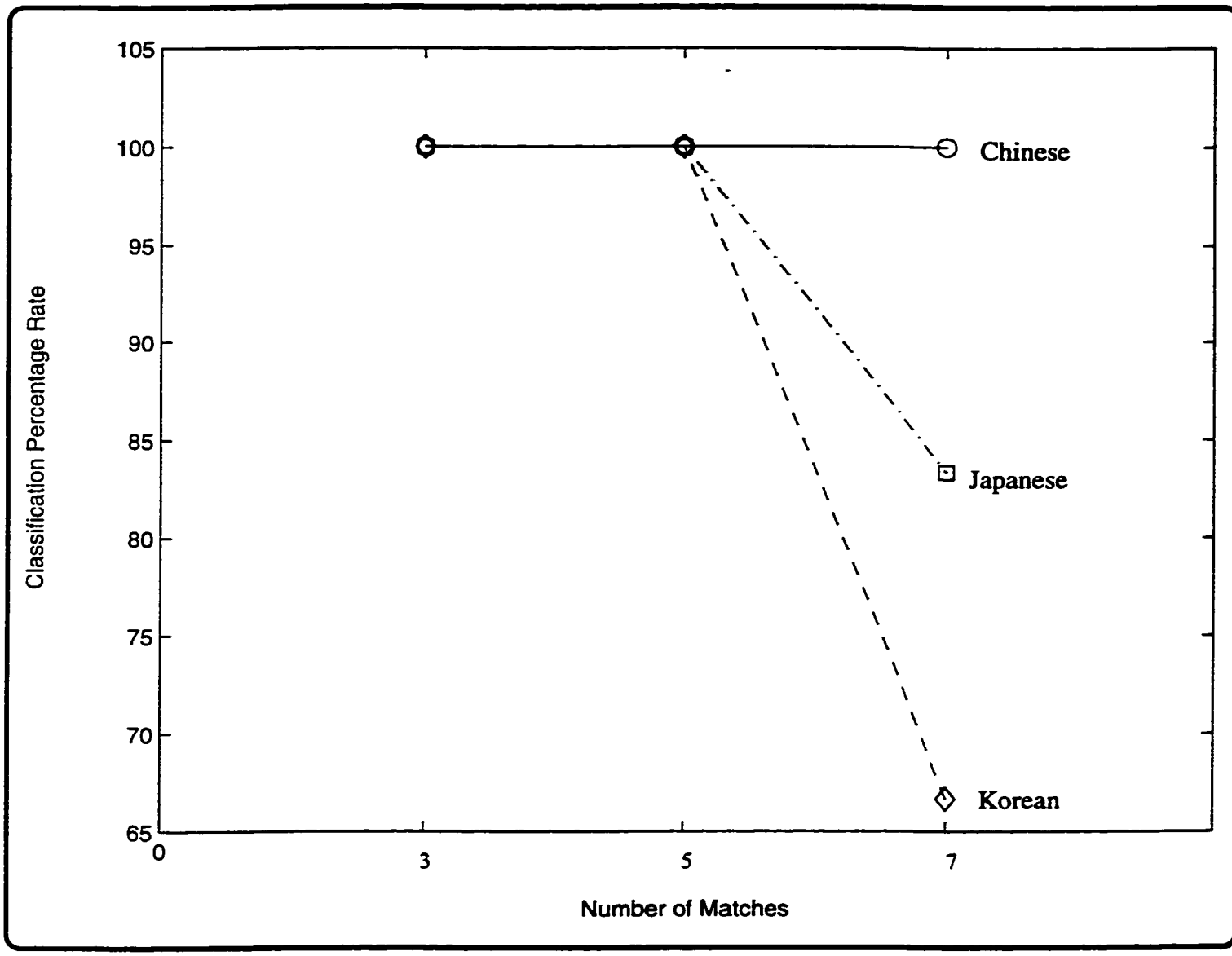

Figure 45: The classification rates of Chinese, Japanese, and Korean documents in category 4 of Hamanaka's dataset with 3, 5, and 7 matches.

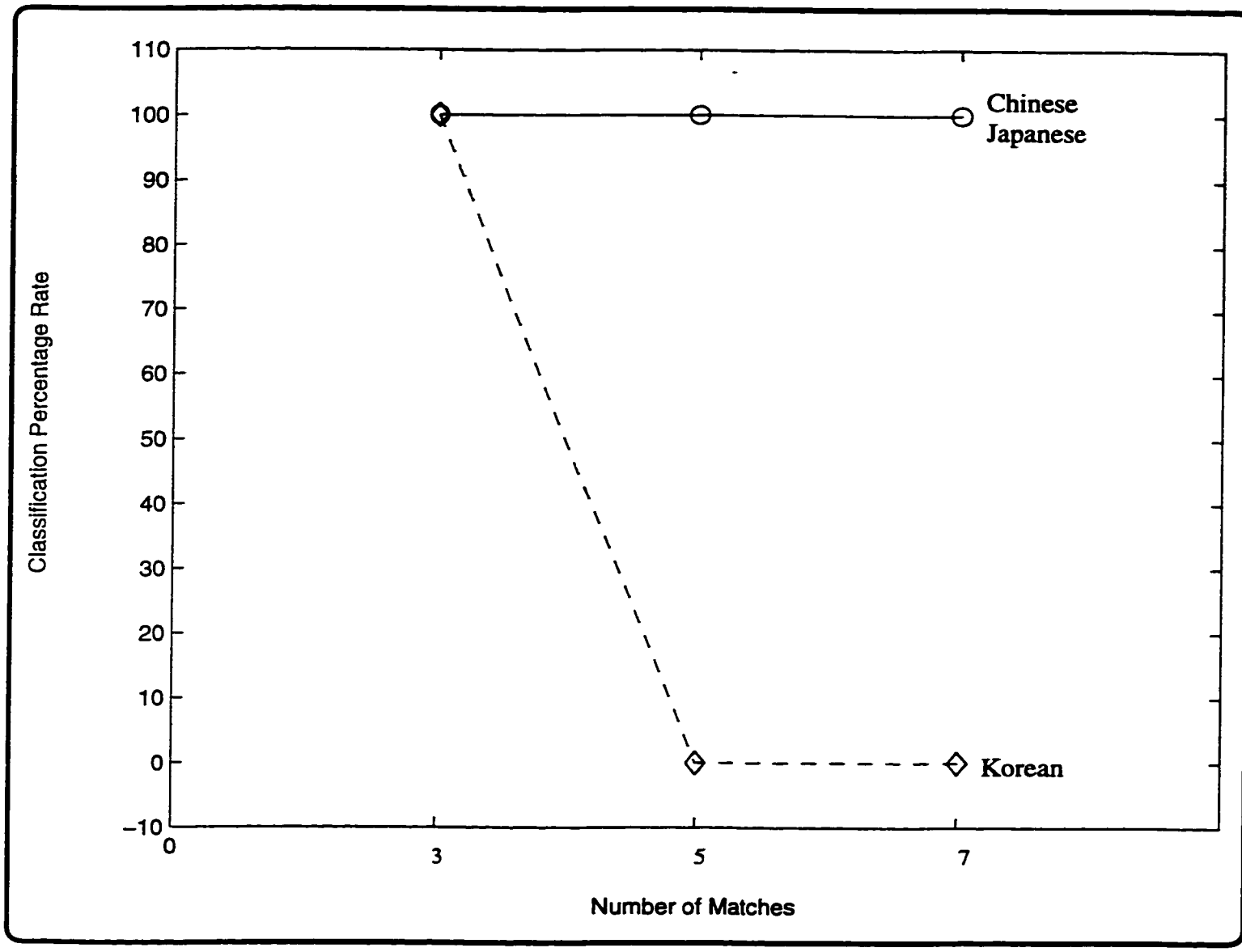

Figure 46: The classification rates of Chinese, Japanese, and Korean documents in category 1 of Ding's dataset with 3, 5, and 7 matches.

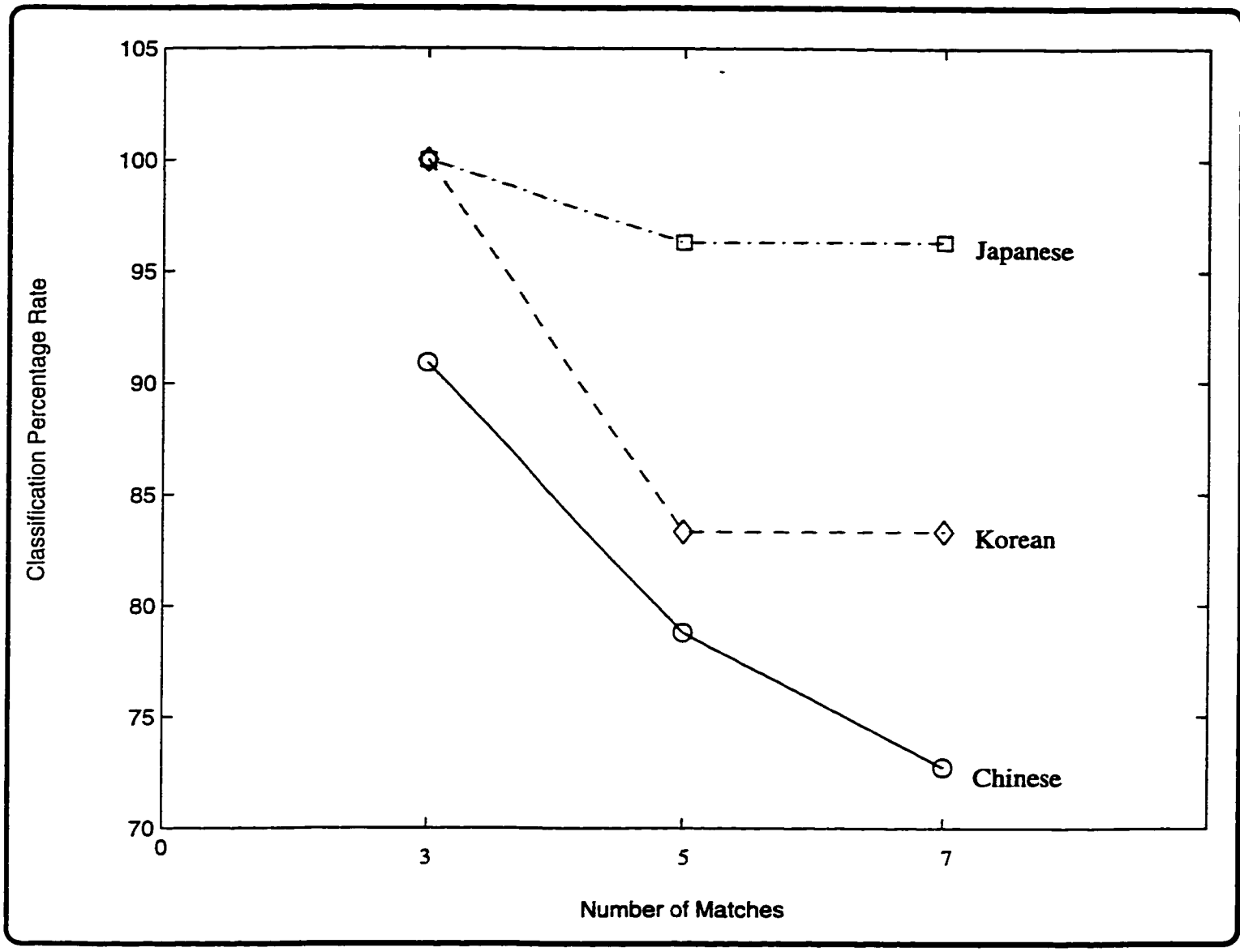

Figure 47: The classification rates of Chinese, Japanese, and Korean documents in category 2 of Ding's dataset with 3, 5, and 7 matches.

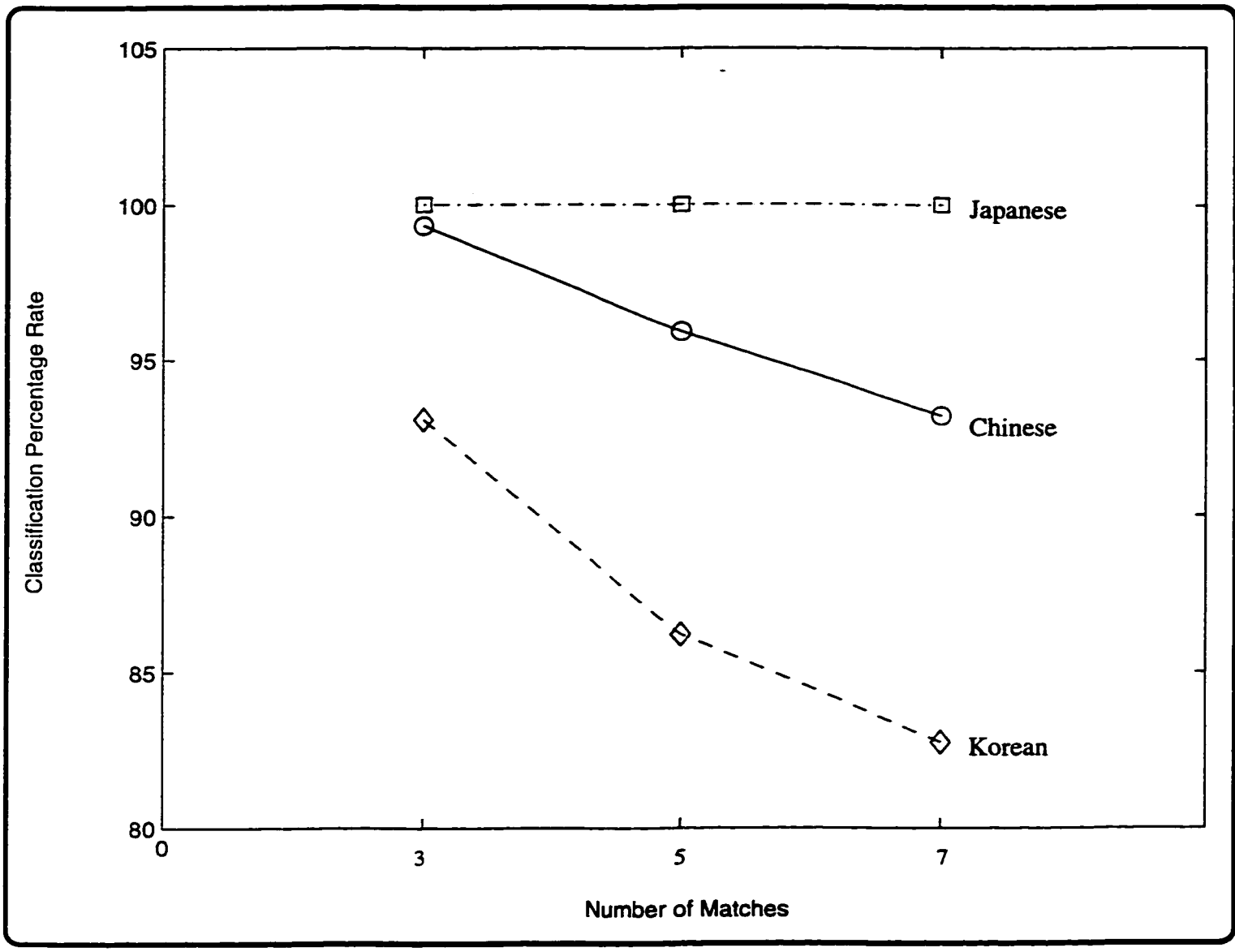

Figure 48: The classification rates of Chinese, Japanese, and Korean documents in category 3 of Ding's dataset with 3, 5, and 7 matches.

Category 4: 501 characters and more

Figure 49 shows the classification rates of 7 Chinese, 35 Japanese, and 127 Korean documents in the fourth category of Ding's dataset. The Chinese and Japanese classification rates are 100% with 3, 5, and 7 matches. On the other hand, the Korean rate dropped because the document that was properly classified with 3 matches got rejected with 5 matches. No other documents were rejected with 7 matches causing the Korean classification rate to be the same with 5 and 7 matches.

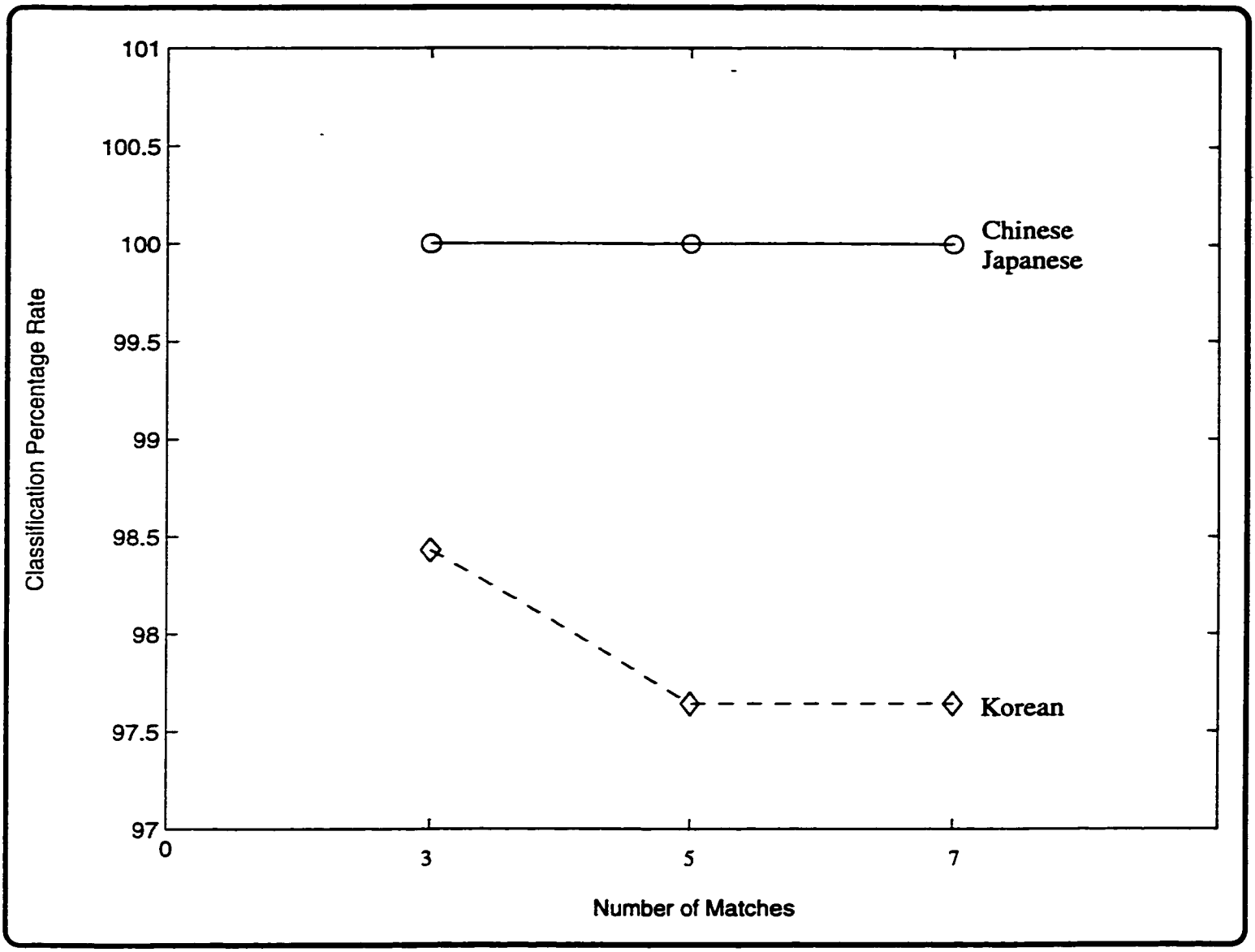

Figure 49: The classification rates of Chinese, Japanese, and Korean documents in category 4 of Ding's dataset with 3, 5, and 7 matches.

Chapter 4

Conclusion and Future Directions

As more and more documents are available electronically, the automatic identification of the language used on documents becomes useful and necessary. It also supports a wide variety of applications ranging from the selection of the appropriate algorithm for optical character recognition to translation.

In this thesis, we have proposed a system that can process Oriental documents and classify them into Chinese, Japanese, and Korean. It is quite challenging to teach computers to identify the script used on Oriental documents because of their complexity and their very large character sets. Even though this area of research is still in its infancy, a number of papers were published on separating European from Oriental scripts, a few others on Oriental document classification into Chinese, Japanese, and Chinese based on statistical features. However, the literature on Oriental language identification is really scarce when the identification process relies on template matching techniques.

Experiments have shown that our method achieved very good results when tested on Hamanaka's database which contains a total of *391* Oriental documents printed in the most commonly used fonts as well as famous and less regularly fonts. Few documents printed in different fonts also existed. Horizontally and vertically printed documents can also be found in this database. The overall classification rate on

Hamanaka's dataset is 93.35% with a 98.92% reliability, keeping in mind that the documents were scanned at 400 dpi. In addition, the method was tested on Ding's database which contains *448* documents, scanned at 1200 dpi, horizontally printed only. The system achieved an overall classification rate of 98.21% with a 100% reliability.

4.1 Major Contributions of the Thesis

4.1.1 Contribution 1: New system to classify Oriental documents

The major contributions of the thesis are research and creation of a new system capable of automatically classifying printed Oriental documents electronically stored in image form. Unlike most identification methods which are statistically based, our method is template based, which is hard to find in the current literature. The basic idea was to assemble templates for a set of the most frequently appearing characters in Chinese, Japanese, and Korean and look for them in new documents. The merit of the method is that it searches for a subset of the chosen models in one iteration and in case no matches were generated, the method looks for the remaining models, obviously the ones that were not used in the first iteration. The method can be easily extended to look for as many models as needed by the application; the same applies to the number of iterations which increases the accuracy of the classification process.

4.1.2 Contribution 2: Horizontal and vertical documents classified

Since Oriental documents can be printed either horizontally or vertically, a general system should be able to process not only horizontal documents but vertical ones as well. We have created an algorithm for the detection of the document's text orientation. Out of *839* processed documents in both databases, only one Chinese document's text orientation could not be detected. The text orientation's error rate

is 0.26% on Hamanaka's set and 0% on Ding's set.

Language identification is a challenging field of research. In this thesis, we proposed a system for the differentiation between Oriental documents. The method proved to be effective based on the test results achieved on *839* documents containing a variety of fonts and printing styles.

Our method has the following merits:

- It has been designed to handle documents printed in various fonts, other than the ones used frequently. It works well on new fonts that do not differ significantly from the regular ones, as was the case with Chinese documents in Xin Yuan font. The method rejects, does not miscalssify, documents if little information was found in them.
- It works well in the processing of *mixed* documents containing a variety of fonts. In most publications, researchers have not reported their comments on this aspect.
- It is based on the averaged Hausdorff distance which is more resilient to noise than the other modified Hausdorff distances found in the literature.

4.2 Future Work

Future study can proceed in the following directions:

- The algorithm currently handles a total of 5 fonts other than the ones normally used to print Oriental documents. As there are more fonts available, the algorithm should be able to identify more than the fonts it currently knows.
- The processing time could be reduced if we try to guess the font prior to the identification process and try to look for the models corresponding to the font identified by the system.

Bibliography

[Bor86] G. Borgefors, *Distance Transformations in Digital Images*, **Computer Vision, Graphics, and Image processing**, Vol. 34, pp. 344-371, 1986.

[Cry94] D. Crystal, **An Encyclopedic Dictionary of Language and Languages**, Penguin Books, London, England, 1994.

[Dan80] P. E. Danielsson, *Euclidean Distance Mapping*, **Computer Vision, Graphics, and Image processing**, Vol. 14, pp. 227-248, 1980.

[DGM98] M. Daoudi, F. Ghorbel, A. Mokadem, O. Avaro, and H. Sanson, *Shape Distances for Contour Tracking and Motion Estimation*, **Pattern Recognition**, Vol. 32, No. 7, pp. 1297-1306, 1998.

[Din99] J. Ding, *Automatic Classification of Multi-Lingual Documents*, Master's thesis, Concordia University, March 1999.

[DLS97] J. Ding, L. Lam, and C.Y. Suen, *Classification of Oriental and European Scripts using Characteristic Features*, **Proceedings of the 4th International Conference on Document Analysis and Recognition**, pp. 1023-1027, Ulm, Germany, 1997.

[DH75] R.O. Duda and P. E. Hart, *Use of the Hough transform to detect lines and curves in pictures*, **Communications of the Association for Computer Machinery**, Vol.15, pp. 11-15, 1975.

[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**, Vol.10, No. 6, pp. 910-918, 1988.

[GT98] B. Günsel and M. Tekalp, *Shape Similarity Matching for Query-by-Example*, **Pattern Recognition**, Vol. 31, No. 7, pp. 931-944, 1998.

[HKK95] J. Hochberg, L. Kerns, P. Kelly, and T. Thomas, *Automatic Script Identification from Images Using Cluster-Based Templates*, **Proceedings of the 3rd International Conference on Document Analysis and Recognition**, pp. 378-381, Montréal, Queébec, Canada, 1995.

[Hul98] J. Hull, *Document Image Similarity and Equivalence Detection*, **International Journal on Document Analysis and Recognition**, Vol. 1, pp. 37-42, 1998.

[HKR93] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, *Comparing Images Using the Hausdorff Distance*, **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Vol. 15, No. 9, pp. 850-863, 1993.

[LNB96] D. S. Lee, C. R. Nohl, and H. S. Baird, *Language Identification in Complex, Unoriented, and Degraded Document Images*, **Proceedings of Document Analysis Systems**, pp. 76-98, Malvern, Pennsylvania, USA, 1996.

[LKB96] J. S. Lee, O. J. Kwon, and S. Y. Bang, *Highly Accurate Recognition of Printed Korean Characters through an Improved Grapheme Recognition Method*, **Proceedings of the 13th International Conference on Pattern Recognition**, pp. 447-451, Vienna, Austria, 1996.

[LTC97] C. H. Leung, W. C. Tam, and Y. S. Cheung, *Recognition of Handwritten Chinese Characters by Structural Matching*, **Proceedings of the 17th International Conference on Computer Processing of Oriental Languages**, Vol. 1, pp. 624-629, Hong Kong, China, 1997.

[Li98] D. Li, **Road Sign Recognition**, Master's thesis, Concordia University, September 1998.

[Nak80] A. Nakanishi, **Writing Systems of the World - Alphabets, Syllabaries, Pictograms**, Charles E. Tuttle Company, Rutland, Vermont and Tokyo, 1980.

[NS93] T. Nakayama and A. L. Spitz, *European Language Determination from Images*, **Proceedings of the 2nd International Conference on Document Analysis and Recognition**, pp. 159-162, Tsukuba, Japan, 1993.

[NBS97] N. Nobile, S. Bergler, and C. Y. Suen, *Language Identification of On-Line Documents Using Word Shapes*, **Proceedings of the 4th International Conference on Document Analysis and Recognition**, pp. 258-262, Ulm, Germany, 1997.

[Pag92] D. W. Paglieroni, *Distance Transforms: Properties and Machine Vision Applications*, **Graphical Models and Image Processing**, Vol. 54, No. 1, pp. 56-74, 1992.

[Pau97] J. Paumard, *Robust Comparison of Binary Images*, **Pattern Recognition Letters**, No. 18, pp. 1057-1063, 1997.

[RSL95] E. Reiher, F.N. Said, Y. Li, and C.Y. Suen, *Map Symbol Recognition Using Directed Hausdorff Distance and a Neural Network Classifier*, **Proceedings of the 18th ISPRS Congress**, Vol. XXXI, Part B3, pp. 680-685, Vienna, Austria, 1996.

[RK82] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, 2nd ed, Academic Press, New York, 1982.

[Ruc95] W. J. Rucklidge, *Efficient Computation of the Minimum Hausdorff Distance for Visual Recognition*, Ph.D. thesis, Cornell University, 1995.

[Ruc96] W. J. Rucklidge, **Efficient Visual Recognition Using the Hausdorff Distance**, Lecture Notes in Computer Science, Springer, 1996.

[SS94] P. Sibun and A. L. Spitz, *Language Determination: Natural Language Processing from Scanned Document images*, **Proceedings of the 4th Conference on Applied Natural Language Processing**, pp. 15-21, Stuttgart, Germany, 1994.

[SR96] P. Sibun and J. C. Reynar, *Language Identification:Examining the Issues*, **In 5*th* Symposium on Document Analysis and Information Retrieval**, pp. 125-135, Las Vegas, Nevada, USA, 1996.

[Spi94] A. L. Spitz, *Script and Language Determination from Document Images*, **Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval**, pp. 229-235, Las Vegas, Nevada, 1994.

[Spi97] A. L. Spitz, *Determination of the Script and Language Content of Document Images*, **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Vol. 19, No. 3, pp. 235-245, 1997.

[Str93] N. W. Strathy, *A method for Segmentation of Touching Handwritten Numerals*, Master's thesis, Concordia University, 1993.

[SBN98] C. Y. Suen, S. Bergler, N. Nobile, B. Waked, C. P. Nadal, and A. Bloch, *Categorizing Document Images into Script and Language Classes*, **International Conference on Advances in Pattern Recognition**, pp. 297-306, Plymouth, UK, 1998.

[SH98] C. Y. Suen and M. Hamanaka, *Visual Cues for Automatic Identification of Languages*, **Proceedings of Vision Interface**, pp. 365-372, Vancouver, British Columbia, Canada, 1998.

[SMR98] C. Y. Suen, S. Mori, H. Rim, and P. S. P. Wang, *Intriguing Aspects of Oriental languages*, **International Journal of Pattern Recognition and Artificial Intelligence**, Vol. 12, No. 1, pp. 5-29, 1998.

[T98] B. Takács, *Comparing Face Images Using the Modified Hausdorff Distance*, **Pattern Recognition**, Vol. 31, No. 12, pp. 1873-1881, 1998.

[Tan96] T. N. Tan, *Written Language Recognition Based on Texture Analysis*, **Proceedings of the International Conference on Image Processing**, Vol. 2, pp. 185-188, Lauzanne, Switzerland, 1996.

[TS97] C. Tzomakas and W. V. Seelen *An Object Recognition Scheme Using Knowledge and the Hausdorff Distance*, **Proceedings of Vision Interface**, pp. 108-113, Kelowna, British Columbia, Canada, 1997.

[WBS98] B. Waked. S. Bergler, C. Y. Suen and S. Khoury *Skew Detection, Page Segmentation, and Script Classification of printed Document Images*, **IEEE International Conference on Systems, Man, and Cybernetics**, pp. 4470-4475, San Diego, California, USA, 1998.

[Wan88] P. S. P. Wang, *Intelligent Chinese Language, Pattern, and Speech Processing*, **World Scientific Publishing**, Singapore, 1988.

[WS93] M. Worring and A. W. M. Smeulders, *Digital Curvature Estimation*, **CVGIP: Image Understanding**, Vol. 58, No. 3, pp. 366-382, 1993.

[Yam84] H. Yamada, *Complete Euclidean Distance Transformation by Parallel Operation*, **Proceedings of the 7th International Conference on Pattern recognition**, pp. 69-71, Montréal, Québec, Canada, 1984.

Index