HISTORICAL DOCUMENT ANALYSIS BASED ON WORD MATCHING

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING

AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Damla Arifoğlu

June, 2011

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Pınar Duygulu (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Fatoş Yarman Vural

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Fazlı Can

Approved for the graduate school of engineering and science:

Prof. Dr. Levent Onural
Director of the Institute

ABSTRACT

HISTORICAL DOCUMENT ANALYSIS BASED ON WORD MATCHING

Damla Arifoğlu
M.S. in Computer Engineering
Supervisor: Asst. Prof. Dr. Pınar Duygulu
June, 2011

Historical documents constitute a heritage which should be preserved and providing automatic retrieval and indexing scheme for these archives would be beneficial for researchers from several disciplines and countries. Unfortunately, applying ordinary Optical Character Recognition (OCR) techniques on these documents is nearly impossible, since these documents are degraded and deformed. Recently, word matching methods are proposed to access these documents. In this thesis, two historical document analysis problems, word segmentation in historical documents and Islamic pattern matching in kufic images are tackled based on word matching. In the first task, a cross document word matching based approach is proposed to segment historical documents into words. A version of a document, in which word segmentation is easy, is used as a source data set and another version in a different writing style, which is more difficult to segment into words, is used as a target data set. The source data set is segmented into words by a simple method and extracted words are used as queries to be spotted in the target data set. Experiments on an Ottoman data set show that cross document word matching is a promising method to segment historical documents into words. In the second task, firstly lines are extracted and sub-patterns are automatically detected in the images. Then sub-patterns are matched based on a line representation in two ways: by their chain code representation and by their shape contexts. Promising results are obtained for finding the instances of a query pattern and for fully automatic detection of repeating patterns on a square kufic image collection.

Keywords: Historical Manuscripts, Ottoman Documents, Word Image Matching, Word Spotting, Word Segmentation, Islamic Pattern Matching.

ÖZET

KELİME EŞLEŞTİRME TABANLI TARİHSEL BELGE ANALİZİ

Damla Arifoğlu
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Asst. Prof. Dr. Pınar Duygulu
Haziran, 2011

Tarihsel belgelerin otomatik erişimi ve dizinlenmesi bir çok alandan ve ülkeden araştırmacı için faydalı olacaktır. Ne yazık ki, bu belgelerdeki yıpranma ve lekeler yüzünden, Optik Karakter Tanıma (OPT) tekniklerinin bu belgelerde başarılı olması zordur. Son zamanlarda, bu belgelerde erişim problemi kelime eşleştirme yöntemleriyle çözülmeye çalışılmıştır. Bu tezde, iki tarihsel belge analizi problemi, tarihsel belgelerin kelimelere bölütlenmesi ve Kufi resimlerinde İslami motiflerin eşleştirilmesi, kelime eşleştirme tabanlı yöntemlerle çözülmeye çalışılmıştır. Birinci problemin çözümü için çapraz belgelerde kelime eşleştirme tabanlı bir yöntem önerilmiştir. Bir belgenin kelime bölütlemenin kolay olacağı bir versiyonu kaynak veri kümesi ve de diğer başka bir yazı tarzıyla yazılan ve kelime bölütlemesinin zor olacağı bir versiyonu da hedef veri kümesi olarak kullanılmıştır. Kaynak veri kümesi basit bir yöntemle kelimelerine bölütlenmiş ve elde edilen bu kelimeler sorgu kelimeleri olarak kullanılarak hedef veri kümesindeki yerleri saptanmaya çalışılmıştır. Yapılan deneyler, çapraz belgelerde kelime eşleştirme tabanlı yöntemin tarihsel belgelerde kelime bölütlemesi için umut verici sonuçlar verdiğini göstermiştir. İkinci problemin çözümü için sunulan yöntemde, öncelikle resimlerdeki çizgiler çıkartılır ve alt-kelimeler otomatik olarak bulunur. Daha sonra alt-kelimeler, çizgi tabanlı zincir kod gösterimi eşleştirmesi ve şekil içeriği tanımlayıcısı eşleştirmesi yöntemleriyle eşleştirilir. Kare kufi resimlerinden oluşan bir veri kümesi üzerinde yapılan deneyler, sunulan kelime eşleştirme tabanlı yöntemin umut verici sonuçlar verdiğini göstermiştir.

Anahtar sözcükler: Tarihi Metinler, Osmanlıca Belgeler, Kelime Resimlerinde Eşleştirme, Kelime Erişimi, Kelime Bölütleme, İslami Motif Eşleştirme.

Dedicated to my parents...

Acknowledgement

First of all, I would like to express my gratitude to my supervisor Dr. Pınar Duygulu from whom I have learned a lot due to her supervision, patient guidance, and support during this research. Without her invaluable assistance and encouragement, this thesis would not be possible.

I am indebted to the members of my thesis committee Dr. Fatoş Yarman - Vural and Dr. Fazlı Can for accepting to review my thesis and for their valuable comments. Also, I thank to Mehmet Kalpaklı for his support.

I am thankful to Dr. Uğur Sezerman who started my academic journey by making me love research, learn and teach. His endless enthusiasm in the courses he taught and his love of research will be always in my mind throughout my academic life.

I am grateful to Mehmet, who has been more than a friend, a brother to me and show his care and endless love throughout my life.

I would like to express my special thanks to Anıl, for always being so supportive and cheerful towards me. Library hours and lodgment days would not be so enjoyable without her.

I thank to the members of EA 128 Office, Selen, Buğra and Ateş, for their good companionship.

The biggest of my love goes to my beloved family, for their endless support and unconditional love, and for giving me the inspiration and motivation. None of this would be possible without them.

My last acknowledgment must go to Armağan, who was always there for me with his patience and support when I needed the most.

This thesis is supported by TÜBİTAK grant no : 109E006.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The historical documents are valuable since they provide information about the old times of the history and they constitute a heritage which should be preserved. In recent years, the need to preserve and provide access to these ancient documents has increased [72]. Electronic access of the historical documents would shed light into a relatively unknown era and providing automatic retrieval and indexing schema for these archives would be beneficial for researchers from several disciplines and countries.

Unfortunately, studying with these ancient documents is challenging since they are degraded and deformed and not in good quality due to faded ink and stained paper and may be noisy because of deterioration [59]. Thus, applying ordinary Optical Character Recognition (OCR) techniques on these documents is nearly impossible. Most of the historical documents are kept in image format and it is easier to access these historical and cultural archives by their visual contents and treating them as images is the start point of this study.

Most of the studies such as retrieval and indexing of historical documents require word segmentation before searching a word on a document. Thus, before providing fast indexing and retrieval approaches for an easy navigation of historical documents, providing a word segmentation schema would be beneficial and make further processes easier and faster.

In the first task, a cross document word matching based approach is proposed to segment historical documents into words. Some documents may have multiple copies (versions) with different writers or writing styles. Word segmentation in some of those versions may be relatively easier than other versions. Based on this knowledge, a version of a document is used as a source data set and another version of the same document, which is difficult to segment into words, is used as a target data set. Words in the source data set are automatically extracted by a simple word segmentation method and these words are used as queries to be spotted in the target data set. Proposed approach is evaluated on an Ottoman document collection which consists of two different versions of a document in different writing styles. Experiments show that a cross document word matching based approach is a promising method on deciding the word boundaries in historical documents.

Islamic Calligraphy also constitute a heritage in the context of cultural heritage preservation. It has been widely used on many religious and secular buildings and manuscripts throughout the history and kufic is one of the oldest calligraphic forms of the various Islamic scripts [7]. Examples of kufic calligraphy can be found in a wide range region including Byzantine icons and the engravings in French, Italian Churches, on African coins and buildings [7]. It was generally used as a decorative element on tombstones, coins and rugs, as well as for architectural monuments such as some pillars, minarets, porches and on the walls of palaces [32, 6] and it is still used in some Islamic countries [7]. Thus, providing an indexing and retrieval schema for kufic calligraphy images would be beneficial for researchers.

In the second task, two methods are proposed for matching Islamic patterns in square kufic images. The methods are based on a line representation, in which lines are extracted by Douglas-Peucker (DP) line simplification algorithm. Then, sub-patterns in the images are automatically detected by finding the connected lines resulting in a polygonal shape. In the first method, chain code representations for all sub-patterns are extracted and sub-patterns are matched by a string matching algorithm, in which chain code representations are considered as strings. In the second method, the similarity between sub-patterns are calculated

by their shape contexts. On a square kufic image collection, promising results are obtained for finding the instances of a query pattern and for fully automatic detection of repeating patterns.

In the following, firstly cross document word matching based word segmentation task is explained and followed by experiments on an Ottoman data set. Then, Islamic pattern matching task is presented and followed by experiments on a square kufic image data set.

Chapter 2

Segmentation of Historical Documents

2.1 Motivation

In recent years, as an alternative to Optical Character Recognition (OCR) techniques, word spotting methods have been proposed for the easy access and navigation of historical documents [73, 56, 57]. These studies are inspired from the cognitive studies that have observed the human tendency to read a word as a whole [55] and generally use the image properties of a word to match it against other words. Most of the word spotting techniques require word segmentation before searching a word [56, 18, 19]. Although there are some segmentation-free approaches [13, 38], their computational cost is high to search a query word on a document. Thus, providing a word segmentation schema would be beneficial and make further processes easier and faster. On the other hand, word segmentation is difficult in historical documents where the words may touch each other due to handwriting style or due to the high noise levels.

Most of the word segmentation algorithms are based on analysis of character space distances [79, 41, 47, 78, 54]. But these methods are likely to fail in

languages such as Ottoman, Persian, Arabic, etc., in which a word may be composed of one or more sub-words (a sub-word is a connected group of characters or letters, which may be meaningful individually or only meaningful when it comes together with other sub-words). This means there are also inter-word gaps as well as intra-word gaps and it is not easy to decide whether a gap refers to an intra-word gap or an inter-word gap in these languages (see Figure 2.1). When intra-word gaps are as large as inter-word gaps or when both gaps are very little, the segmentation gets harder and error-prone. Word segmentation of documents in these type of languages usually requires a knowledge of the context, which can be learned by recognition, but on the other hand, most of the time word segmentation is required before recognition.

Some historical documents have been copied with multiple people due to the unavailability of the printing technology, resulting in different versions of the same document in different writing styles. Also, recently printed copies of historical handwritten documents become available. In this study, we make use of the multiple copies (versions) of the same document for segmenting words in historical documents which is difficult, if not impossible, using traditional word segmentation techniques.

The main idea behind our approach is that, some of the versions are easier to segment, and the difficult versions can be segmented by carrying out the information from these simpler ones. For this purpose, a cross document word matching based approach is proposed. The words in a source document which are easier to segment are spotted in a target document in which the identification of the word boundaries is difficult.

While our approach is related to the recognition based character segmentation studies in the literature [87, 80, 26, 85] and word spotting studies on multi-writer data sets [73, 40, 12, 22], as far as we know, there is no recognition based word segmentation method and our study is the first application of word matching idea for segmentation of words across documents.

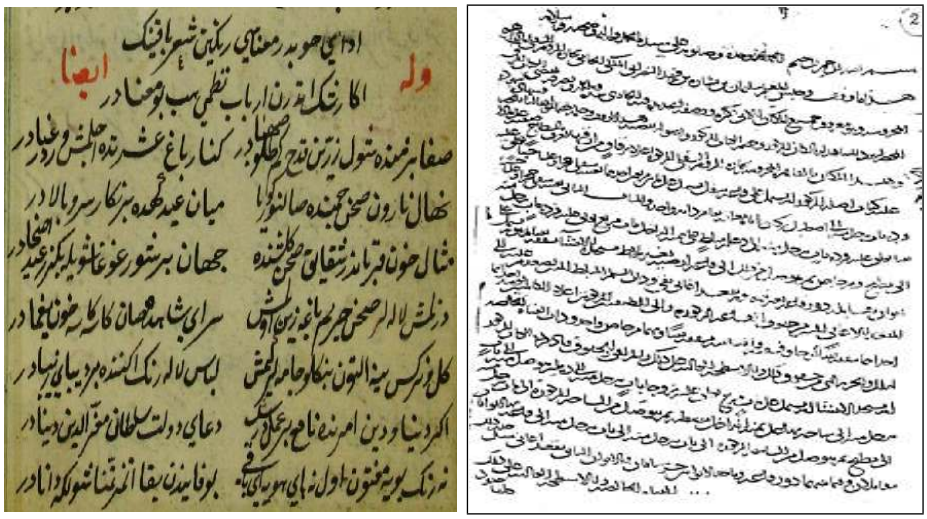


Figure 2.1: First one is an example page in Ottoman and second one (image courtesy of [9]) is in Arabic. Note that identification of word boundaries on these pages is very difficult since intra and inter word gaps can not be easily discriminated.

In this task, our contributions are three-fold. Firstly, we propose a cross document word matching method. Since the words in different versions can have different writing styles and noise levels, this is a more difficult task compared to word matching in the same document. We propose a matching scheme based on possible ordered combinations of sub-words. Holistic word matching methods may not be successful in inter-document word matching since words may be broken into smaller units in different versions of historical documents. Thus, sub-word (robust units of a word) matching would be more successful in inter-document word matching. It is easy to divide a word into units of sub-words and during word matching process, all combinations of sub-words are considered in order to spot a word. Similar approach is adapted in [60], where words are modeled as a concatenation of Markov models and a statistical language model is used to compute word bigrams.

Secondly, we use the word matching idea for segmentation of target documents

which are difficult, if not impossible, to segment by traditional approaches. Target documents are segmented into words using the information carried from the source documents which are easier to segment. At the same time with word segmentation, also an indexing scheme can be provided for the target dataset if labels are ready for the source data set.

Thirdly, we use the context information for word matching. When the words are spotted across documents individually, they can be mismatched, but we consider words as ordered sequences, in form of lines or sentences. In this way, a word and its consecutive and preceding words are considered as a whole and this context information is used during word matching. Our experiments show that word segmentation success rates are low when query words are spotted independently from their left/right neighbor words, while success rates increase when a query word is considered in a neighborhood.

The proposed approach is language independent and can be applied to any pair of documents. In this study, we focus on Ottoman divans, which in general have multiple copies by nature. In the following section, first we describe the general characteristics and challenges of the data set used.

2.2 Ottoman Data Set

Archives of Ottoman empire, which had lasted for more than 6 centuries and spread over 3 continents, attract the interests of scholars from many different countries all over the world [10]. Because of the importance of preserving and the need for efficient access, in recent years the amount of studies tackling with indexing and retrieval problem of Ottoman documents has increased [18, 19, 85, 77, 86, 24, 23]. Most of the Ottoman documents not only contain text, but also drawings, miniatures, portraits, signs and ink smears, etc., which have also historical value and relation with the corresponding text. Thus, accessing this documents by their visual content is important.

The Ottoman is a connected script which is actually a subset of Arabic alphabet. Also it has additional vocals and characters from Persian and Turkish scripts [46]. Since Ottoman is a cursive language and the Ottoman characters are composed of interconnected symbol groups [68], determining the word boundaries in Ottoman documents is not easy. Ottoman is very different from languages such as English in which there are explicit word boundaries. In Ottoman, a word may be composed of many individual sub-words and a space does not always correspond to a word boundary. Even a word can comprise one, two or more sub-words, without explicit indication where a word ends and another begins.

Another difficulty is that Ottoman characters may take different shapes according to their position in the sentence; being at beginning, middle, at the end and in isolated form [16, 25, 45]. Most of the characters are only distinguished by adding dots and zigzags (called as diacritics), which makes the discrimination harder.

Divans are collections of all poems written by the same poet and there may be many versions of divans in different writing styles. Therefore, they fit very well to be used as a data set for applying our approach. In this study, source data set is selected as a machine printed version and target data set is selected as a *lithography* printed version of a document. Target data set is not in a good quality and it has some deformations and noise. *Lithography* is a method in which a stone or a metal plate with a completely smooth surface is used to print text onto paper [43]. It was the first fundamentally new printing technology and invented in 1798. In this method, letters or characters are not ordered by machine, they are put on stone by humans. Spaces between characters are sometimes very large and sometimes very little. While in a normal printed text, by classifying space between characters, word boundaries can be easily determined, in *lithography* printed documents, it is not easy to distinguish inter-word and intra-word distances. Also, as it was mentioned, in Ottoman documents there are inter-word gaps as well as intra-word gaps, which makes the inter-word and intra-word gap distance classification harder. Thus, word segmentation methods which are based on gap distances can not be successful in these documents, while proposed word matching based word segmentation method does not depend on intra and

inter-word distances and it can tolerate any gap distance between sub-words and words.

Although source and target data sets are different writing versions of an Ottoman Divan and the content of the divan in these documents is the same, the documents may show some differences (see Figure 2.2). For example, they may have different number of words and lines. Because in different versions of divans, some words and lines may be omitted or new words and lines may be added. This is normally a case in handwritten documents, but at early times of progression from handwriting to printed era, this trend was also observed in *lithography* printed documents and this shows that handwriting culture is also continuing in *lithography* printed documents [43, 50]. For example, some lines of a gazel (gazel is a specific type of a poem in Ottoman literary and the data sets in this study are composed of gazels), may be omitted and sometimes new lines may me added in target version. Sometimes a word of a line in source data set may be omitted in target version and most of the times, it may be changed with a different word as well as position of it may be changed. Besides, even if the same word is used in different versions, without changing its meaning, this word may be written in a different form, by connecting some characters. The most common difference between source and target data sets is their character shape styles, which is named as writing style difference. For example, while a character may have a longer curve in source data set, its curve may be kept shorter, or additional curves may be added to it in target version. Moreover, historical documents may have some broken characters in some versions of a document because of deterioration and this makes word matching more difficult. Thus, these differences between multi copies of a divan make cross document word matching more challenging.

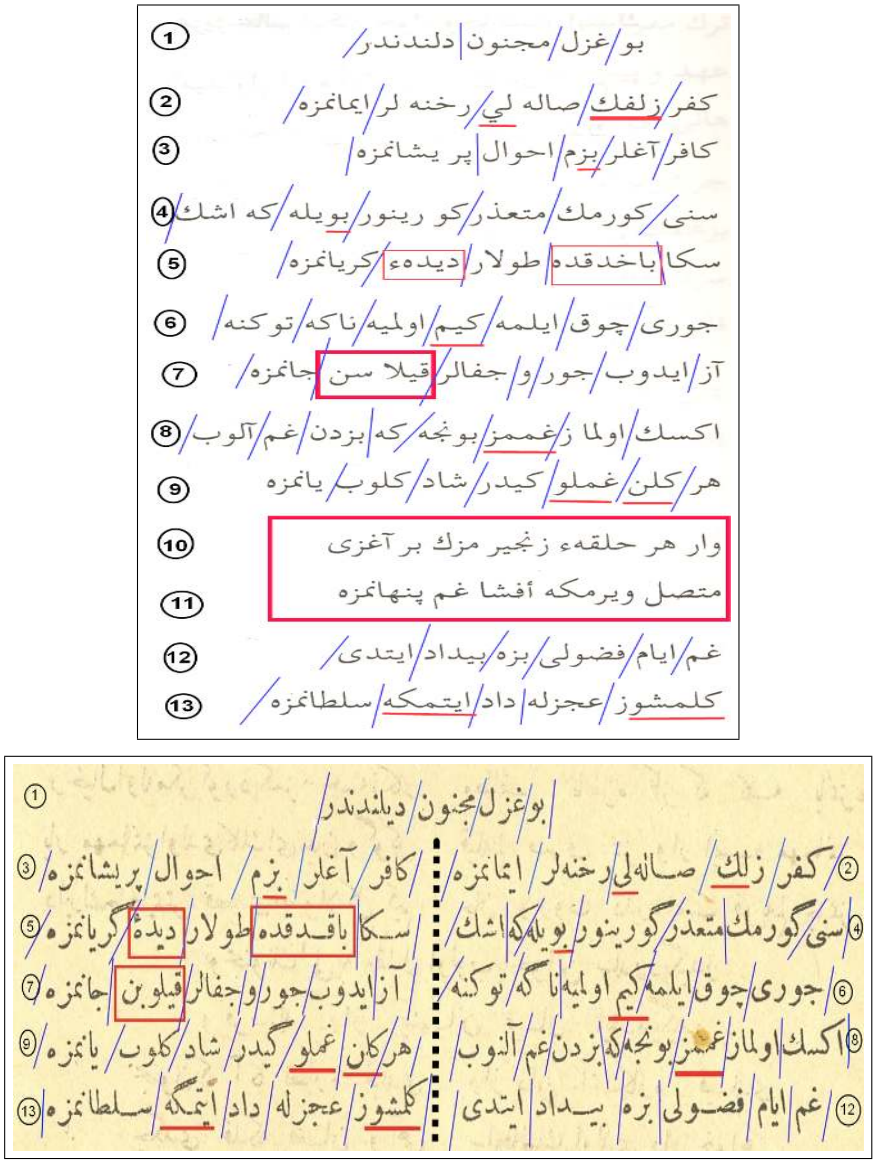


Figure 2.2: First image is an example page from source data set, which is machine printed and second one is from target data set, which is *lithography* printed. In the first image, it is easy to separate words from each other, while it is hard to define intra and inter word gaps in the second one. The lines show the correct word boundaries in pages. Lines 10 and 11 are missing in the second page. Also, the words in rectangles are different or written in a different form and the sub-words underlined are same in two pages, but their characters have different shapes which makes the matching process harder.

2.3 Related Work

In this section, we review the related studies for (i) word segmentation and (ii) word matching.

2.3.1 Word Segmentation

Most of the proposed segmentation algorithms such as X-Y cuts or projection profiles, Run Length Smoothing Algorithm (RLSA), component grouping, document spectrum, whitespace analysis, constrained text lines, Hough transform, Voronoi tessellation, projection profiles and scale space analysis etc. are mainly designed for contemporary documents [64]. But in historical documents, because of the degradation due to old printing quality and ink diffusion, these approaches may not be useful [64].

Previous work on segmenting historical documents into words includes many approaches for both printed and handwritten documents and these studies are generally based on analysis and classification of distance relationship of adjacent components [79, 41, 47, 78, 54].

In [79, 41], to decide whether the gap between two adjacent connected components in a line is a word gap or not, an artificial neural network is designed with features characterizing the connected components. In another study [78], three algorithms are combined to separate text lines into words. The first algorithm decides the distance between adjacent connected components. Eight methods (bounding box, Euclidean method, the minimum run-length method etc.) are tried in order to measure the distances between pairs of connected components. In the second algorithm, using a set of fuzzy features and a discrimination function, punctuation marks are detected since they are useful for segmentation. The third algorithm combines the first two algorithms to rank the gaps from the

largest to the smallest and determines which gaps are inter-word and which are intra-word gaps.

The classification method of distance measures as intra or inter word gaps is as important as distance metric selection and in [47], three clustering techniques are used and sequential clustering is shown to be the best when three different distance metrics (bounding box, run length/Euclidean, Convex hull) are used.

In [54], word segmentation of handwritten documents are done based on a distinction of inter and intra-word gaps using combination of two metrics (Euclidean and convex hull distance). Deciding on word boundary is considered as an unsupervised clustering problem, in which Gaussian mixture theory is used to model two classes.

Classical word segmentation methods which are based on a single value threshold, can not be successful when the gap distance between words varies from page to page in a document. In study [82], single value threshold problem is solved by applying a tree based structure. The gaps between words are not only decided based on a single threshold. Also context information in form of sizes of adjacent gaps, is also used to decide the gap distances. It is claimed that this method outperforms classical word segmentation methods. The proposed method is an extension of traditional threshold based methods and instead of a single threshold value, a more flexible threshold is used. Nevertheless, the method is still based on a threshold like traditional methods and this does not eliminate the need of a space threshold.

In [48], word segmentation problem in historical machine-printed documents is solved based on a run length smoothing in the horizontal and vertical directions. For each direction, white runs having length less than a threshold are eliminated in order to form word segments. As a result, a binary image where characters of the same word become connected to a single connected component is retrieved.

In [64], traditional Run Length Smoothing Algorithm (RLSA) is adapted for word segmentation. Firstly, all connected components are extracted and sorted according to their X coordinates and adjacent components with a distance smaller

than a threshold are considered to belong to the same word.

Manmatha and Rothfeder [58], used a technique for automatically segmenting documents into words, in which scale space ideas are used to smooth images to produce blobs. At small scales, these blobs correspond to character-like entities and at larger scales, they correspond to word-like entities.

In [18, 19], Ottoman documents are segmented into words by vertical projection profiles.

Adapting the methods above to the problem of word segmentation in Ottoman archives is hard because Ottoman words may consist of more than one sub-word, which means there are inter and intra word gaps in these documents. The space between two sub-words of a word may be bigger than the space between two words. In a case like this, classical word segmentation methods may not work very well. On the other hand, a matching based method may tolerate the large variations in inter and intra word spaces since word matching is not based on any distance between words or sub-words.

2.3.2 Word Matching

In word spotting, different instances of a word are clustered using some word image matching techniques. And in the literature, there are many features used to calculate the similarity of visual words [57, 18, 19, 12, 48, 69, 83, 33, 21, 71].

Dynamic Time Warping (DTW) is one of the most famous methods to calculate the similarity of words [69, 83, 33, 21, 71], in which two words are treated as signal samples and they are tried to be matched dynamically. In [36], frequency and position of the words are used as features for similarity measurement in DTW algorithm, while in the study [69], some features (word profiles, background to ink transition and vertical projection) are extracted and used in DTW algorithm to calculate the similarity between two words.

In [70], five different matching algorithms are used to index a digital library

of George Washington's manuscripts. These five matching methods are XOR, Euclidean Distance Mapping (EDM), Sum of Squared Differences (SSD), SLH and the authors claim that EDM is the best performing algorithm in their data set. But these methods are sensitive to spatial variation since pixels are directly compared. On the other hand, DTW features are not pixel based and they can tolerate spatial variations since DTW compares feature vectors.

Also there are some studies which use the well-known Shape Context [20] descriptor in order to match visual words [53]. But shape context method is very time consuming and the authors of [57] claimed that shape context method was not successful on George Washington collection. Because of these reasons, shape context descriptor is not preferred in this study.

In [12], a new approach is proposed to holistic word recognition for historical handwritten manuscripts. This method is based on matching word contours instead of whole images or word profiles. In this way, contour-based descriptors can capture a word's important details in a better way and also eliminate the need for skew and angle normalization.

In [74], a statistical framework is introduced to spot words in handwritten documents and a hidden Markov model is employed to model keywords and Gaussian Mixture model is used for score normalization. They showed that statistical methods outperform traditional DTW methods for word image distance computation.

Besides the studies above, also there are gradient based studies to match visual words [79, 74, 88, 52]. In study [88], gradient based binary features, namely gradient, structural and concavity are used. Authors compared their method with profile based methods, such as DTW and showed that their proposed method is faster than the existing best handwritten word retrieval algorithm DTW. In study [52], instead of comparing pixels directly, gradient fields are compared, since they provide more information than the grey levels.

In [76], word corners are obtained by Harris corner detector and points between word images are tried to be recovered by minimizing Sum of Squared

Distance (SSD) error. It is a simple method to apply but Harris corner detector is sensitive to noise and this method is not applicable to historical data sets which are noisy.

In [18], quantized vertical projection profiles and in [19], bag-of-visterms approach is used for matching word images in Ottoman documents. Also, in [24, 23], word images are represented with a set of line segments and retrieved by a line based matching algorithm in Ottoman manuscripts as well as in George Washington documents.

Multi-writer word retrieval studies [73, 40, 12, 22], also tackle the problem of writing style variations on multi writer data sets. But these studies are generally based on finding an instance of a query word and words are in isolated forms. Cross document word matching is difficult because of the writing style variations across documents, but it is even more challenging since words are not segmented and not in isolated form.

As a result, since DTW features are not pixel based, they can tolerate spatial variations. In cross documents even same words may have spatial differences in different copies, thus DTW features are preferred for word matching in this study.

2.4 Our Approach

In this study, a cross document word matching based method is presented to segment historical documents into words. As it is depicted in figure 2.3, the proposed approach has the following steps: i) A version of a document, which is easy to segment into words, is chosen as a source data set and another version of the same document in a different writing style is used as a target data set. ii) All sub-words in source and target data sets are extracted. iii) Features are extracted for each sub-word in the data set. iv) From the source data set, words are extracted by a simple word segmentation method. v) Extracted words are tried to be spotted in the target data set by a word matching method. vi) In the target data set, word boundaries are decided based on the spotted words of the

source data set.

In the following, the each step will be described in detail.

2.4.1 Preprocessing

After the original documents are converted into gray scale, they are binarized by adaptive binarization method [42]. Small noises such as dots and other blobs are cleaned by removing connected components which are smaller than a predefined threshold. Then, pages are segmented into lines by a Run Length Smoothing algorithm [54].

Most of the historical documents are degraded and there may be some broken characters in these documents. In order to connect the broken characters, a Manhattan distance based approach is used, in which Manhattan distances between adjacent ink pixels are calculated. These pixels are connected if their distances are smaller than a predefined threshold.

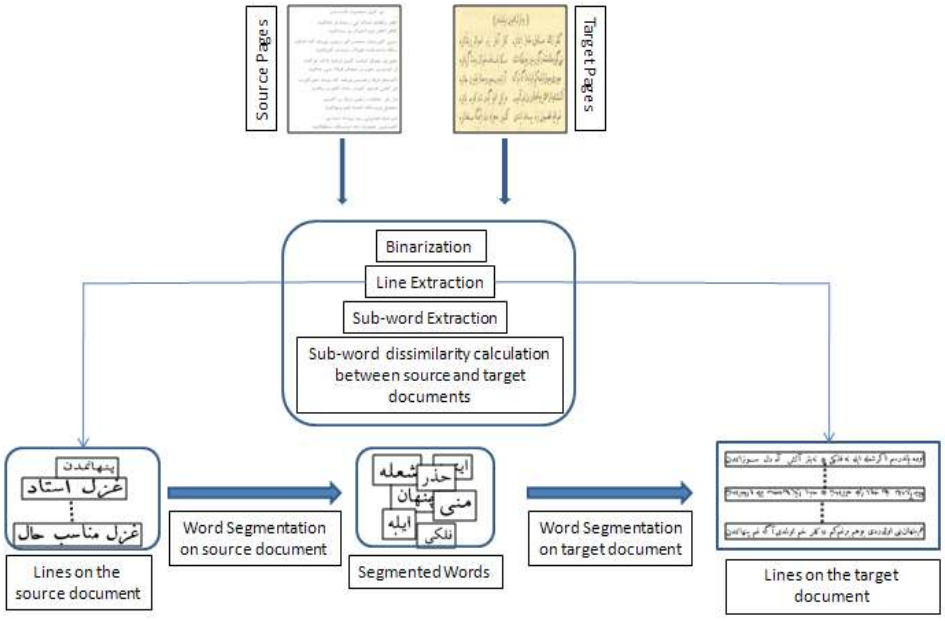


Figure 2.3: Overview of our approach

2.4.2 Sub-word Extraction

Firstly, all connected components in both source and target data sets are extracted by a boundary detection algorithm. A connected component is defined as a connected group of black pixels in the document image. These connected components may be any script or character such as a dot, two dots, an individual letter or a group of connected letters (see figure 2.4). The diacritics such as dots and zig zags are called as minor components and the other bigger connected components, such as letters and connected group of characters, are called as major components. Instead of working with minor and major components separately, a major component C is combined to all its minor components and the resultant group of connected components is called as a sub-word. These sub-words may be individual words or they may form words by coming together with other sub-words. Note that a sub-word may consist of only a single major component or it may be formed by a major component and a group of minor components (see figure 2.4).

Figure 2.4: All the black pixel groups are individual connected components (CC). There are 11 CCs in this image, with 6 major CCs and 5 minor CCs, constructing 6 sub-words, which are separated by lines.

2.4.3 Feature Extraction

The choice of the features and the similarity measure that will be used in word matching algorithm is important. For word matching, Dynamic Time Warping (DTW) method is one of the mostly used approaches [69] and for computing the similarities, we also prefer to use DTW in this study. Features used in the proposed approach are adapted from [69, 34]. These features are: Upper/Lower word profiles, Vertical projection, Upper/Lower vertical projection, Background to ink transition, Second moment order, Center of gravity, Number of foreground pixels between the upper and the lower contours, Variance of ink pixels.

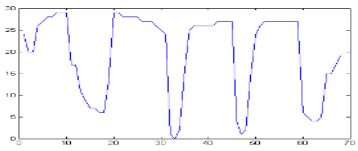
Upper and lower word profiles are good features to learn about the outline shape of a sub-word. But, word profiles provide a coarse way of representing word images. On the other hand, projection and transition features capture the ink distribution. Center of gravity and second moment order characterizes the column from the global point of view. They give information about how many pixels are in which region and how they are distributed.

These features are calculated for each column of a sub-word and as a result a 10-variant feature vector of length a (where a is the column size of the corresponding sub-word) is constructed. The range of each feature dimension is normalized to the unit interval. In figure 2.5, a word image and its corresponding 10 features are given.

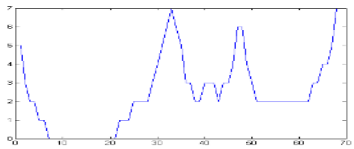
2.4.4 Sub-word matching

After feature extraction step, sub-word similarities are found based on DTW algorithm [69]. Before continuing with the word matching algorithm, firstly a short description for DTW algorithm is given.

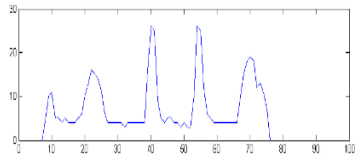
(a) An Ottoman word



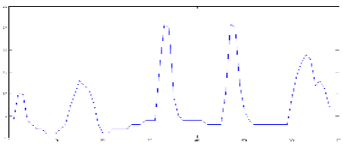
(b) Upper Word Profile



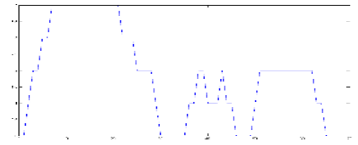
(c) Lower Word Profile



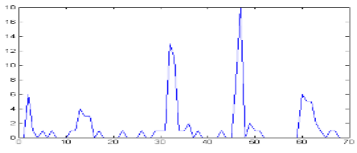
(d) Vertical Projection



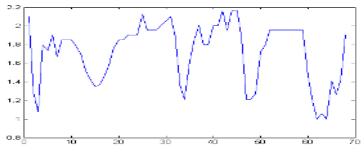
(e) Upper Vertical Projection Profile



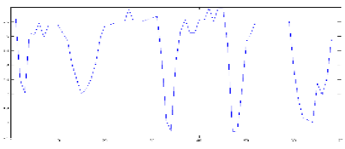
(f) Lower Vertical Projection Profile



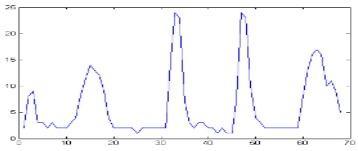
(g) Background To Ink Transition



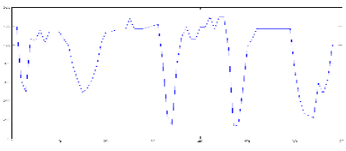
(h) Second Moment Order



(i) Center of Gravity



(j) Number of foreground pixels between upper and lower contours



(k) Variance

Figure 2.5: An Ottoman word (*"fate"*) and its 10 features.

Dynamic Time Warping Method

DTW was first introduced for putting series into correspondence and firstly used in speech recognition [49]. In recent years, DTW has been also used in the field of document analysis in order to match visual words [69, 83, 33].

As described in [69], the best characteristic of DTW algorithm is that the two samples do not have to be in the same size (see figure 2.6). In this way, the same words in different sizes can be easily matched and this is an important feature of DTW in cross document word matching. In cross documents, the same characters may be in the different sizes because of different writing styles and fonts.

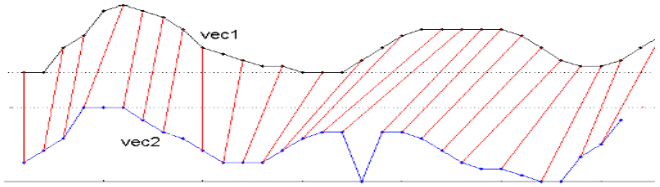


Figure 2.6: In DTW method, signals having different lengths can be easily matched.

In DTW, the distance between 2 time series, which are lists of samples taken from a signal, ordered by time, is calculated with dynamic programming as in the equation (2.1), where $d(x_i, y_i)$ is the distance between i^{th} samples.

$$D(i,j) = min \left\{ \begin{array}{l} D(i,j-1) \\ D(i-1,j) \\ D(i-1,j-1) \end{array} \right\} + d(x_i, y_i) \tag{2.1}$$

Without normalization, DTW algorithm may favor the shorter signals. In order to prevent this, a normalization is done based on the length of the warping

path K. Backtracking finds the warping path, which is along the minimum cost index pairs of matrix D. As a result, total distance $D(i, j)$ is divided by this path length K.

Sub-word similarity calculation

Back to the proposed method, DTW cost between two sub-words is calculated by equation (2.2), where k is the index of the feature, A and B are two character images, i and j are the columns of A and B and the cost is normalized by warping path length K.

$$d(x_i, y_j) = \sum_{k=1}^{10} (f_k(A_i) - f_k(B_j)) \tag{2.2}$$

In word matching approaches based on DTW, some pruning methods such as aspect ratio or size of the visual word image are used in order to reduce the number of candidates for a query word [69]. Since words are tried to be matched in different data sets, size variations of even the same characters are large between source and target data sets. In this study, column difference is used in order to eliminate weak candidates for a sub-word, but it is preferred to eliminate less number of candidates because of the large variations in size of sub-words across documents.

After sub-word similarities are calculated, query words are tried to be spotted in the target data set. Spotting of query words is done by a matching method which is based on the comparison of individual sub-words rather than entire words or characters.

2.4.5 Line Matching

When a query word is tried to be spotted in all target lines, number of possible candidate pseudo words is very high and size of search space increases. In order

to decrease the size, firstly source and target lines are tried to be matched. After line assignment is done, words of a source line are only searched in its assigned target line (see figure 2.7).

Before line matching step, weak target lines for each source line are pruned in two ways : (i) Gazel based pruning (ii) Line based pruning. Firstly, total number of sub-words in each source and target gazels and in each source and target lines are calculated separately. If difference of number of sub-words in any two source and target gazel is smaller than a threshold, these gazels are said to be possible matchings for each other, otherwise that target gazel is removed from possible set for that source gazel. And then for each source line, possible lines are found among these pruned set. If difference of the number of sub-words in a source and a target line is smaller than a predefined threshold, that target line is a possible matching for that source line. By pruning, size of the set of possible target lines for each source line is reduced.

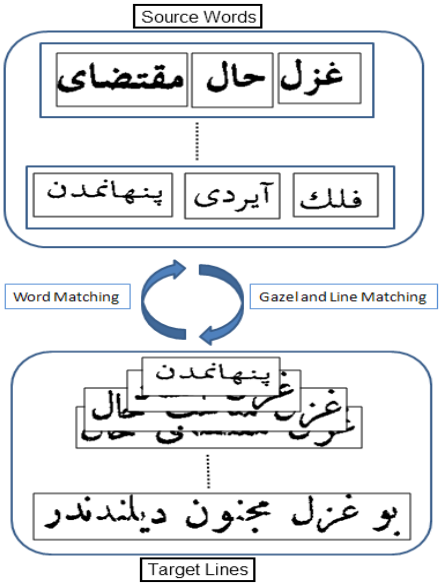


Figure 2.7: A two-way feedback mechanism is provided for line and word matching. After assigning a source line to a target line, words of a source line are spotted in its assigned target line.

After pruning steps, line assignment between source and target lines are done in the following way : Words of each source and target lines are tried to be matched by using a position constraint. If a query word starts at a_{th} sub-word and ends at b_{th} sub-word of a source line, that query word is only searched in a range of target sub-words occurring at positions $[a - x, b + y]$, where x and y are selected constants (they are taken as 3 in the experiments). In this range, a pseudo word is generated and it is matched with that query word as it will be described in section 2.4.5. Individual word matching scores are summed and the target line which has the minimum distance to a source line is assigned to that source line.

Algorithm 1 Line Matching

Input: sourceLines[1...m], targetLines[1...n], TH_1, TH_2

1: TH_1 is gazel based and TH_2 is line based filtering threshold.
2: U is a function that calculates the number sub-word differences between 2 given part/page or lines.
3: $lineMatchingScoreMatrix[1...m, 1...n] \leftarrow 0$
4: **for** $i = 1 \rightarrow m$ **do**
5: $u \leftarrow gazelIdOfLine_i$
6: **for** $j = 1 \rightarrow n$ **do**
7: $t \leftarrow gazelIdOfLine_j$
8: **if** $U_{i,j} \leq TH_1$ & $U_{u,t} \leq TH_2$ **then**
9: lineMatchingScoreMatrix(i,j) = sequenceMatching(i,j)
10: **else**
11: lineMatchingScoreMatrix(i,j) = intMax;
12: **end if**
13: **end for**
14: **end for**

Algorithm 2 Sequence Matching based on position constraint

Input: $sourceLine_j, targetLine_t$

1: $sourceLineWords \leftarrow [m...1]$
2: $targetLineSubwords \leftarrow [1...k]$
3: $lineMatchingScore \leftarrow 0$
4: $start \leftarrow k$
5: **for** $i = 1 \rightarrow m$ **do**
6: $\quad [a, b] \leftarrow positionRangeOfWord_i$
7: $\quad [wordMatchingScore \;\; start] \leftarrow nGramSearch(word_i, [a, b])$
8: $\quad lineMatchingScore \leftarrow lineMatchingScore + wordMatchingScore$
9: **end for**

Word Matching

Assume that A is a query word which has 4 sub-words and B is a target pseudo word which has 5 sub-words such as Then, all possible ordered combinations of target and source sub-word matchings are found and the combination with the minimum score is chosen as the matched target word. For example, for A and B, the all combinations are listed in figure 2.8.

S_4	S_3	S_2	S_1
T_5	T_4	T_3	T_2–T_1
T_5	T_4	T_3 – T_2	T_1
T_5	T_4-T_3	T_2	T_1
T_5 – T_4	T_3	T_2	T_1

Figure 2.8: Target sub-words are assigned to source sub-words and among these combinations, the sequence with minimum score is chosen.

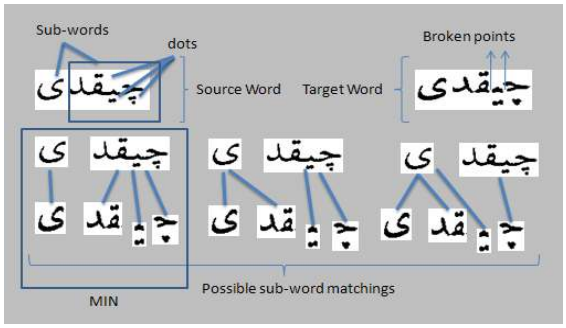


Figure 2.9: A source word is matched with a target pseudo-word based on n-gram approach. Possible ordered combinations of sub-words are found and the best sequence is found for the query word.

After, source and target sub-words are matched, total word distance is calculated as in equation (2.3), where $cost(a_i, b_i)$ is DTW distance of i^{th} query sub-word and assigned sub-word or sub-words to it. This approach also solves the broken sub-word problem, with the assumption that only target data set may have broken sub-words.

$$cost(A, B) = \sum_{i=1}^{n} (cost(a_i, b_i)) \tag{2.3}$$

2.4.6 Word Segmentation based on Word Matching

After the line assignment is done between source and target lines, words of a source line are tried to be spotted in its assigned target line by a sliding window approach. A query word is searched in the candidate line based on an n-gram approach by shifting a sliding window over the target sub-words. Assume that there are k sub-words in the query word, then size of sliding window equals to $k+n$, where n is a constant. At each sliding iteration, a source pseudo word is constructed which has $k+n$ sub-words. This query word and source pseudo-word

is matched as told in 2.4.5 and the pseudo-word which has the minimum match score is chosen as the spotted word.

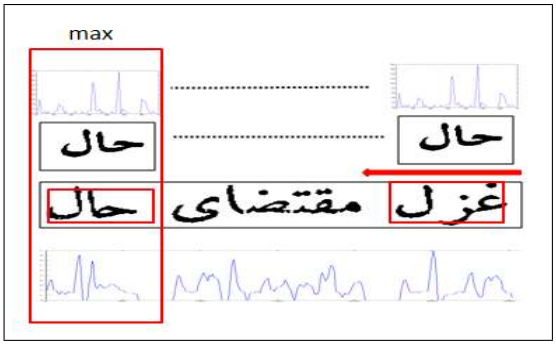


Figure 2.10: A query word is searched in a candidate line by a sliding window approach and at each iteration, matching score is calculated for query word and the generated source pseudo-word.

Chapter 3

Word Segmentation Experiments

3.1 Datasets used in the experiments

Two versions of *Leyla and Mecnun Divan* by *Fuzuli* in two different writing styles are used as the data set. The *Divans* are the collection of poems written by the same poet. A *gazel* is a specific form of a poem in Ottoman poetry and *Leyla and Mecnun Divan* is composed of *gazels* and source and target data sets consist of 26 gazels. Both data sets are obtained by scanning books with resolution 300x300.

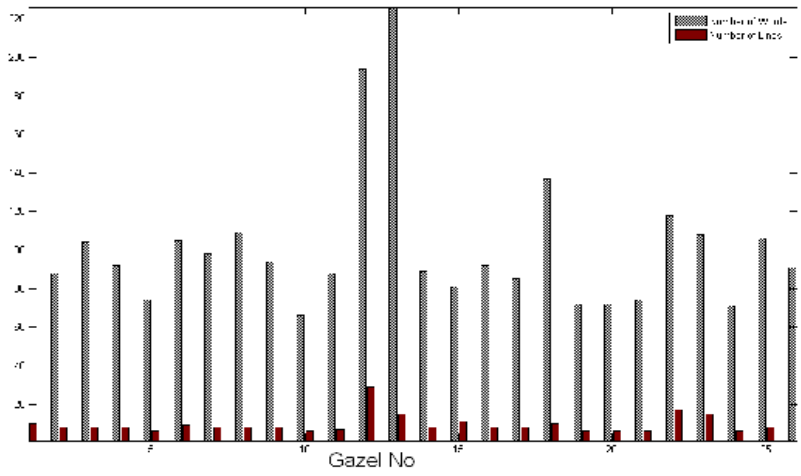
Source data set is a machine printed version of *Leyla and Mecnun* [29]. It was printed with laser printed technology in 1996. It's in good quality, not noisy and degraded and there is no deformation on pages. Also because of a relatively newer printer technology (machine printed), intra and inter word distances can be easily distinguished. This data set is used as source data set since it is relatively easier to segment into words with a simple word segmentation method. Target data set is nearly 100 years older than source data set and it's a *lithography* printed version of *Leyla and Mecnun*. It is not in a very good quality and it has some deformations and noise.

	Source	Target
Printed Technology	Machine	Lithography
Printed Year	1996	1897
Number of Total Lines	408	402
Number of Total Words	2688	2640
Number of Unique Words	1379	1357
Number of Sub-words	5964	6186

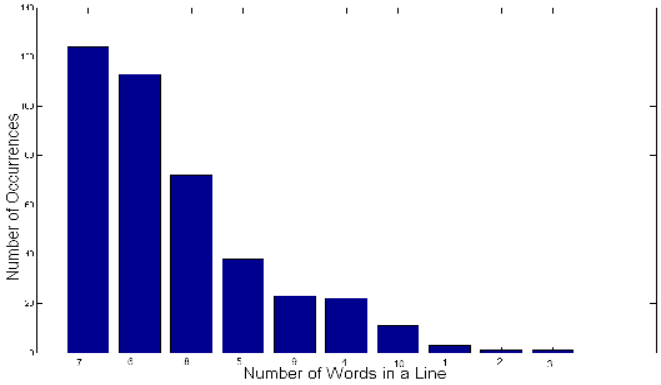
Table 3.1: Source and target data sets and their features.

Table 3.1 summarizes the features of source and target documents. As it can be seen, although these data sets are different writing versions of the same document and content is the same, they may have different number of words and lines. Because in different versions of divans, some words and lines may be omitted or added. As a total, 124 words of the source data set are not included in target data set, while 76 words of target set are not in source data set and 8 lines of source data set are not included in target data set and two new additional lines are added to target data set. For example, as it can be seen in Figure 2.2, two lines of a gazel in source data set are removed in target version.

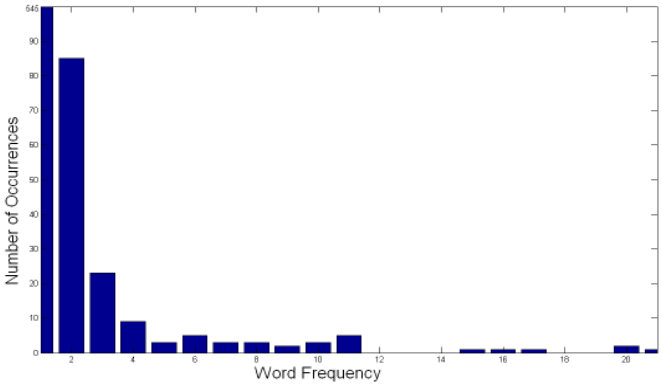
Also, in total there are 26 gazels in the data sets and in Figure 3.1, statistics are given for number of words in each gazel. as it can be seen in Figure 3.1-a, there can be up to 226 words and 29 lines in a gazel. On the average, there are 102 words and 10 lines in a gazel. Also, as it can be seen in Figure 3.2-b, a line may be as short as one words and at most there are 7 words in a line. Figure 3.2-c shows that, while there are 2688 words in source data set, most of them appear only a few times.



(a)

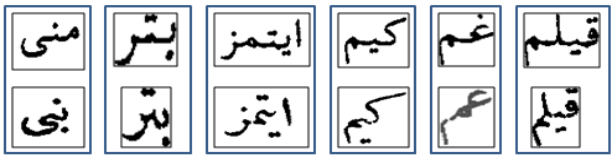


(b)

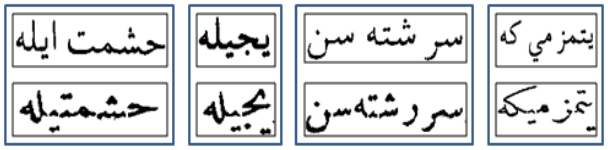


(c)

Figure 3.1: For the source data set (a) The statistics about the distribution of the number of words and lines in each gazel (b) The distribution of number of words in a line. (c) Word frequencies.



(a)



(b)

Figure 3.2: In (a) and (b), at the top, there are words in source data set, and at the bottom, there are corresponding words in target data set. (a) Same sub-words in different shapes. (b) Same words are written in different forms.

Figure 3.2-a shows same sub-words having different shapes across documents and matching them is hard. A word may be changed or written in a different form as it can be seen in Figure 3.2-b. For example, the two sub-words of the last word are connected in target data set.

Moreover, broken sub-words are also problem during the word matching. As a total of 2640 source data set words, 614 of them have broken sub-words. Broken sub-words are tried to be connected by a Manhattan distance approach as described in the preprocessing step. Also n-gram based word matching approach is proposed as told previously. Unfortunately, all of the broken sub-words can not be recovered and these unrecovered sub-words make word matching harder. Some sub-words are not extracted correctly in the sub-word extraction step and in the first 12 gazels, 254 sub-words are wrongly extracted out of 2937 sub-words. In Figure 3.3, there are some broken sub-word examples and in Figure 3.4, statistics about the number of broken sub-words are given.

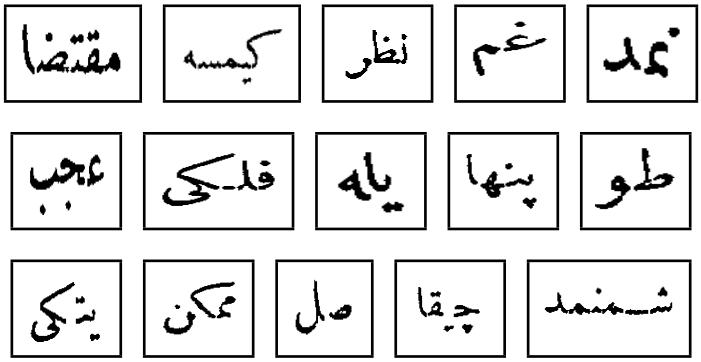


Figure 3.3: Sub-words at first row can be easily connected by Manhattan distance approach, ones at second row can be connected by n-gram approach and ones in third row can not be connected.

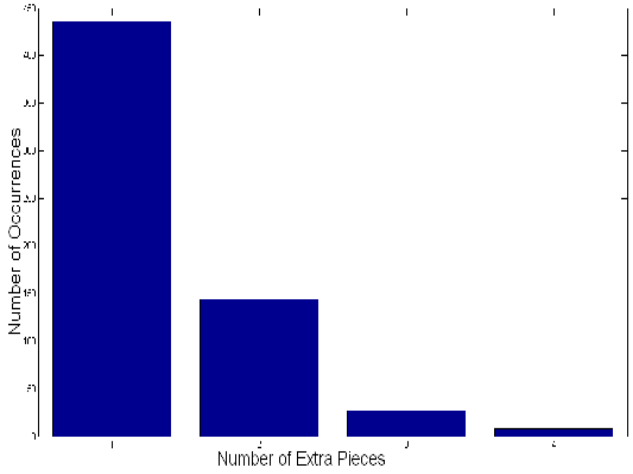


Figure 3.4: Distribution of number of extra pieces that a sub-word is broken into. For example, if a sub-word is broken into two extra pieces, it means that sub-word is broken at two points and as a result that sub-word is composed of three pieces instead of one.

3.2 Evaluation Criteria

In order to evaluate the proposed method, both source and target data set words are manually extracted in order to construct the ground truth data sets. Word segmentation success rates are defined with two different methods as described below.

3.2.1 Full Word Matching Method

In this method, no partial matchings between words are accepted. An extracted word is counted as correct only if it's totally the same as a word in the ground truth data set. In this method, precision and recall values are calculated in order to learn the word segmentation performance as in formulas (3.1) and (3.2).

$$Precision = \frac{\text{number of correctly extracted words}}{\text{number of extracted words}} \tag{3.1}$$

$$Recall = \frac{\text{number of correctly extracted words}}{\text{number of words in the ground truth}} \tag{3.2}$$

3.2.2 Partial Word Matching Method

Besides full word matching scores, IJDAR 2007 word segmentation contest evaluation criteria [37] is used for finding partial word matching performances. In this approach, less punishment is given if a word is not extracted correctly. Firstly, a matching score matrix M is constructed between extracted words and ground truth words of data set. If the number of intersection of ink pixels of an extracted word i and a ground truth word j is above a threshold, these words are said to be matched and *M(i,j)* is set to 1, otherwise it is set to 0. Assume that G is ground truth data set and R is word segmented data set, matching matrix M is

constructed as below, where T is a function to count the number of ink pixels, K is the number of extracted words and N is the number of ground truth elements.

$$M(i,j) = \begin{cases} 1, & \text{if } T(G \cap R)/T(G \cup R) < th \\ 0, & \text{otherwise.} \end{cases}$$

Also, detection rate (DR) and recognition accuracy (RA) are calculated as in equations 3.3 and 3.4.

$$DR = (o\_2\_o + g\_o2\_m + g\_m2\_o)/N \tag{3.3}$$

$$AR = (o\_2\_o + d\_o2\_m + d\_m2\_o)/K \tag{3.4}$$

where o_2_o (number of one ground truth word to one extracted word matching), g_o2_many (number of one ground truth word to many extracted words), g_m2_o (number of many ground truth word to one extracted word), d_o2_many (number of one detected word to many ground truth), d_m2_o (number of many ground truth word to one detected word) are calculated from the matching matrix M. A performance metric is FM is calculated as in equation 3.5.

$$FM = \frac{(2DR \times RA)}{(DR + RA)} \tag{3.5}$$

3.3 Word Segmentation using Vertical Projection based Method

Source data set is segmented into words in order to use segmented words as queries during the word spotting process. This segmentation is done with a

simple vertical projection method. Vertical projection histogram is calculated for each source line image and a predefined threshold *th* is used to decide on word boundaries. If length of a white pixel group between two ink pixels is larger than this threshold *th*, it is assumed that this group of white pixels is an inter word gap. After deciding inter word spaces in this way, words are extracted in the source data set.

Instead of using an empirically selected threshold, a training based approach is preferred to decide the best distance gap amount as an inter word gap threshold. For this purpose, two pages of source data set are used and words on them are manually segmented. Frequency of inter word distances are calculated and the pixel space threshold which occurs most frequently as inter word distance is learned. The applied training process gives this threshold as 7 pixels for source data set and based on this threshold, vertical projection based word segmentation is done in source data set.

Vertical projection based word segmentation success rates in source and target data sets, with different thresholds are given in tables 3.2 and 3.3 respectively. As seen, vertical projection is not successful on deciding word boundaries in target data set since intra and inter word gap distances vary from page to page in this data set.

	Source		Target	
Th	Precision	Recall	Precision	Recall
6	0.8145	0.8253	0.5533	0.4354
8	0.8664	0.8511	0.5918	0.4372
11	0.7995	0.7411	0.5959	0.4304
12	0.7622	0.6783	0.4685	0.3625

Table 3.2: Vertical projection word segmentation success rates in source and target data sets with full word matching word segmentation success criteria. Different thresholds values are used as pixel space distances.

Dataset	Th	AR	DR	FM
Source	1/2	0.94	0.92	0.93
	3/4	0.89	0.88	0.89
Target	1/2	0.64	0.47	0.54
	3/4	0.62	0.45	0.52

Table 3.3: Vertical projection word segmentation success rates in source and target data sets with partial word segmentation success criteria. Threshold 8 is used as gap distance and two different acceptance threshold are used in order to find the partial matchings.

3.4 Word Matching

3.4.1 Intra Document Word Matching

In this experiment, in order to learn how successful profile based features are in intra document word matching, manually segmented words of source data set are searched in source data set itself. 10 pages of source data set are used in this experiment. Firstly, query words are searched in a set of manually extracted words and then query words are searched on source data set lines by a sliding window approach, in which a window is shifted over sub-words and at each iteration, a candidate pseudo-word is generated and these candidate words are tried to be matched with query words. If their matching score is below a threshold, these query and candidate words are said to be matched. As it can be seen in table 3.4, precision and recall rates are high in segmented source data set, while recall rates are very low in unsegmented data set since number of possible candidate pseudo words increases with the sliding window approach and profile based features and DTW method are not good at eliminating false matchings while they can successfully retrieve different instances of a query word. Even intra document word matching is difficult when query words are searched on an unsegmented data set and this is even more difficult when query words are

matched in across documents having different writing styles.

	Segmented Data Set		Unsegmented Data Set	
Th	Recall	Precision	Recall	Precision
0.2	0.80	0.82	0.80	0.39
0.4	0.90	0.76	0.90	0.31
0.6	0.93	0.73	0.93	0.26
0.8	0.94	0.72	0.94	0.25

Table 3.4: Word retrieval success rates in source data set based on different matching score thresholds. Word are tried to be spotted on both segmented and unsegmented source data set.

3.4.2 Inter Document Word Matching and Word Segmentation in Target Data Set

After line assignment between source and target data sets is done, extracted words of source data set lines are spotted on their assigned target data set lines and these target lines are segmented into words as explained previously. Before line matching step, weak target lines for each source line are pruned in two ways (i) Gazel based pruning (ii) Line based pruning. Firstly, total number of sub-words in each source and target gazels and in each source and target lines are calculated separately. If difference of number of sub-words in any two source and target gazel is smaller than a threshold, these gazels are said to be possible matchings for each other, otherwise that target gazel is removed from set of possible matchings for that source gazel. And then for each source line, possible lines are found by using these pruned sets. If difference of the number of sub-words in a source and a target line is smaller than a predefined threshold, that target line is a possible matching for that source line. By pruning, size of the set of possible target lines for each source line is reduced. As it is illustrated in Figure 3.6-a, the maximum number of candidate gazels is 16 for the gazel 26 and Figure 3.6-b shows that there are maximum less than 140 candidate lines for all lines in a gazel after gazel

based pruning. Using these pruned sets, the number of candidate target lines for source lines is maximum 90 as it is shown in Figure 3.6-c.

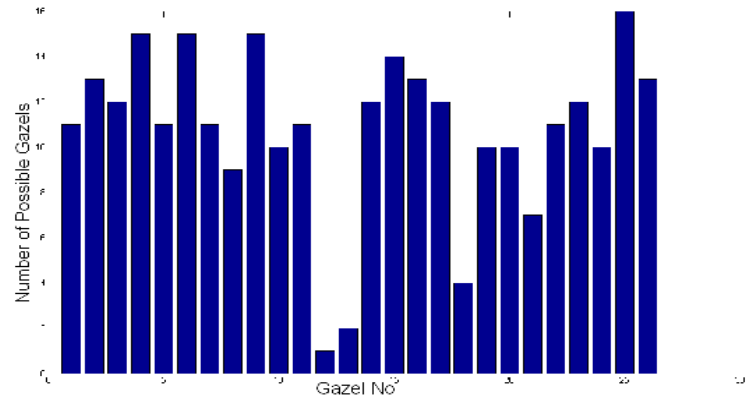
Figure 3.5: Word segmentation examples on target data set. In each box, first line is vertical projection based and second one is proposed word matching based word segmentation results. Lines between sub-words show the found segmentation points in both methods and in second image of each pair, false word boundaries are corrected by using rectangles.

After strong target candidates are found for each source line, among these sets, line assignment is done and experiments show that all source lines are assigned to their correct target lines. After line assignment, segmented words of a source line are spotted in its assigned target line and word segmentation is done in target lines. Word segmentation success rates are given in table 3.5. As seen, when

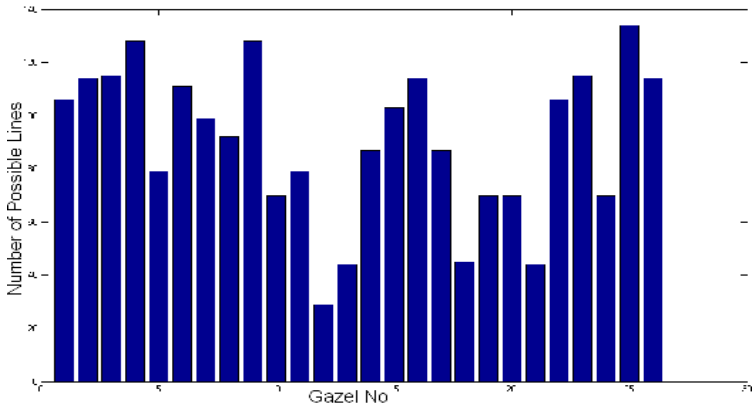
automatically extracted source data set words are used as query words, success rates decrease since some query words may be extracted wrongly and this cause some words to be segmented wrongly. Also, in Figure 3.5, word segmentation examples for some target lines are shown when both proposed word segmentation method and vertical projection based method are used for the word segmentation. In Figure 3.5, in first two line pairs, space distance 5 is used as a threshold to decide word boundaries in the vertical projection based method and in the last two line pairs, threshold 3 is used in order to decide word boundaries. As it can be seen most of the word boundaries decided by the word matching method are correct, while vertical projection based method can not detect word boundaries correctly.

	Manually Extracted		Vertical Projection Extracted	
Method	Precision	Recall	Precision	Recall
Full Word	0.7534	0.7123	0.7034	0.6676
Partial Word	0.7687	0.7456	0.7342	0.6995

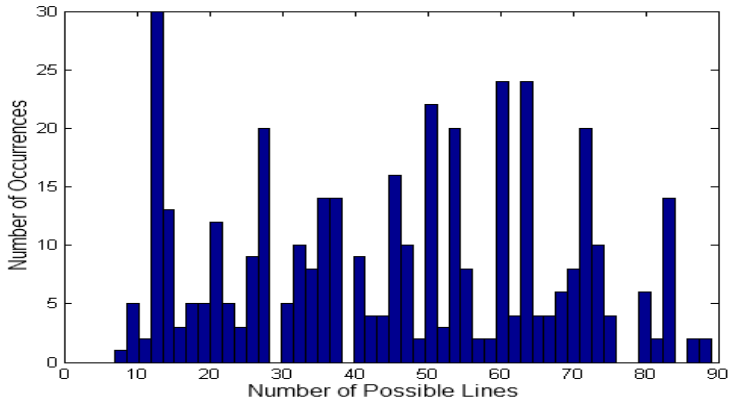
Table 3.5: Word segmentation success rates with full word and partial word matching word segmentation criteria are given. Both manually segmented words and vertical projection based segmented words are used as queries.



(a)



(b)



(c)

Figure 3.6: (a) Number of possible gazels for each 26 source gazel after gazel based filtering. (b) Number of possible lines for each gazel after gazel pruning.(c) Possible target lines for source lines after line based pruning.

3.5 Computational Analysis

When a user searches a query word in a collection that is not segmented into words, all data set lines are needed to be searched by a sliding window approach. But, when a word segmentation schema is provided, a query word is searched only in the set of segmented words, which will speed up the searching process. Word segmentation step requires time but it is done only once and when a user wants to search thousands of query words, the time spent on word segmentation is considered to be negligible. The time spent on word segmentation by the proposed approach is $O(w * l)$ where w is the average number of words in a source line and l is the number of target lines. After the words are segmented, the time spent to search query words is $O(q * w_2)$, where w_2 is the number of extracted words and q is the number of query words to be searched. But in an unsegmented case, when q query words are searched, the search time is $O(q * l)$ and $O(w * l) + O(q * w_2) \leq O(q * l)$, when q is very large.

3.6 Query retrieval

An experiment is performed to learn how much time is spent for word retrieval. Manually extracted source data set words are tried to be retrieved in order to compare the time spent to search a query word when target data set is segmented and unsegmented. Manually extracted words of target data set (assume that there are n words in this data set) are searched in the following sets:

1. in manually extracted words set (ME) : Target words are searched in manually extracted target words set and the time complexity for this search is $O(n^2)$.

2. in unsegmented target lines (UL) : Target words are searched on unsegmented target lines by a sliding window approach. In this experiment, the time required to search a query word and the number of false matchings increase since the number of candidate pseudo words increase at each sliding iteration. Time complexity is $O(n * l)$, where l is the number of lines in the target data set.

3. in vertical projection segmented words set (VPE) : This set is constructed by word segmentation with vertical projection with threshold 8. Complexity is $O(n * k)$, where k is the number of extracted words, since many of the words are extracted wrongly, word retrieval success scores are not very high as seen in table 3.6.

4. in word matching based segmented words set (WME) : This word set is constructed by the proposed word segmentation method. Search complexity is $O(n * t)$ (t is the number of extracted words) and word segmentation complexity which is found in the previous step is $O(w * l)$. It is more complex than vertical projection method, but retrieval success is higher.

As it is depicted in Figure 3.6, searching a query word in an unsegmented data set is very time consuming while searching words in segmented data set is faster. Also, precision and recall rates are promising when word segmentation is done with proposed word matching based method.

Set	Recall	Precision	Average Time (sc)
ME	0.80	0.73	253
UL	0.69	0.51	790
VPE	0.50	0.43	151
WME	0.75	0.70	233

Table 3.6: Query retrieval success scores with word matching threshold 0.4 and average query search time in sets described above.

3.7 Word Segmentation with Word Spotting based on constraints

In this experiment, query words are tried to be spotted without line assignment, which means queries are searched on all target lines by a sliding window approach. Firstly, query words which have a very small distance to their best match candidate pseudo words are extracted. And then by increasing the matching threshold in each iteration, other query words are spotted. Since query word search algorithm is based on a sub-word based sliding window approach, number of possible candidate pseudo words is huge and word segmentation success score is low (see table 3.7). Thus, some constraints are used in order to eliminate false matchings for the query words, which are gazel,line and position constraints. For example, when a gazel/line constraint is used, a query word is only searched in its correct gazel/line. Position constraint forces a query word to be searched in a specific region, which means a query word at the beginning of a line is only search in a range of sub-words at the beginning of target lines. In table 3.7, word segmentation success scores with full word matching criteria are given for different constraints. Success rates are high when all gazel, line and position constraints are used.

			Manually Extracted Words		Automatically Extracted Words	
G	L	P	Precision	Recall	Precision	Recall
			0.56	0.45	0.53	0.43
✓			0.65	0.62	0.60	0.56
✓	✓		0.76	0.68	0.71	0.63
✓	✓	✓	0.81	0.73	0.76	0.68
✓		✓	0.63	0.58	0.57	0.58
		✓	0.68	0.67	0.55	0.59
	✓	✓	0.62	0.56	0.54	0.51
		✓	0.56	0.52	0.51	0.49

Table 3.7: Cross document word matching based word segmentation success scores with full word matching success criteria. In this experiment, both manually and automatically extracted source data set words are used as queries. Also, gazel, line and position (G, L, P) constraints are used during the experiment.

3.8 Discussion of the Results

In this task, a cross document word matching based method is proposed to segment historical documents into words. The proposed method is not dependent on any pixel space between characters or words, thus it is even successful in documents which have little space between characters. The proposed method is based on the idea that word segmentation in some documents is easier and by using automatically extracted words of such a version, another version of the same document, in which word segmentation is harder can be segmented into words. By spotting extracted queries in another version, also an indexing schema can be provided if labels are available for source data set. Experiments on an Ottoman data set with a machine printed and *lithography* printed versions show that promising results are possible for word segmentation with a cross document word matching based method.

Firstly, intra document word matching experiments show that profile based features are good to detect different instances of a query word, but when query words are searched by sliding over the lines, success rate decreases since DTW is not successful to eliminate false matching for query words. Experiments show that DTW is not very successful to eliminate weak candidates in even intra document matching and it would be much challenging to eliminate them across documents because of variations in writing styles of these documents.

Secondly, considering query words as ordered consecutive chains gives higher success rates than searching query words individually. Experiments (table 3.7) show that, when query words are searched in all data set without any constraints, success rates are much lower, because DTW can not eliminate false matchings. When query words are considered in their context, which means query words are matched as a line, success rates increase.

Thirdly, complexity analysis experiments show that searching a query word in a segmented data set is much faster than searching it in an unsegmented data set and precision and recall rates are promising. Although segmenting a data set into words takes some time, it is done only once and when thousands of query words are searched in that data set, segmentation time becomes negligible.

In the future, we aim to segment words in handwritten Ottoman historical documents by using a printed version of them. It is harder to segment handwritten documents into words because of the overlapping and touching characters. Since the most importantly writing styles can be very different, the word matching method should be robust to changes in writing style. DTW based word matching is successful in matching words in the same documents and as we have observed, it can give promising results in similar writing styles. However, it is unsuccessful in matching printed words to handwritten ones as we observed with preliminary experiments. We therefore plan to work on better word matching methods in order to segment handwritten documents into words.

Chapter 4

Islamic Pattern Matching

4.1 Motivation

Figure 4.1: Some decorative kufic patterns on buildings. The first one is Hasankeyf Kizlar Camii minaret. The second one is Tomb Tower (Barda, Azerbaijan, CE 1323). The third one Gudi Khatun Mausoleum (Karabaghlar, Azerbaijan, 1335-1338). The last one is the Mausoleum of Sheikh Safi (Ardabil, Iran, 735 AH).

In kufic calligraphy images, there are some holy words from the Quran such

as *Allah* or *Mohammed* [11, 8]. Some examples[1] of decorative usage of kufic calligraphy can be seen in Figure 4.1. For example, on the last building, there are many *Allah* patterns.

In a kufic calligraphy design, scripts are designed in a non-deterministic process, in which letters and their relative arrangements are updated and the organization is redefined when a new word is added to the composition [62]. Even a single word or letter can be modified in many different ways to create different motifs. Thus, kufic calligraphy is a creative design and its free decorative character lets its motives be used as decorative elements.

Kufic script has 3 different classes: writing kufic, ornamental kufic and Ma'qeli kufic which is known as square kufic and also called as geometric, rectangular, quadrangular or rectilinear kufic. In this study, we prefer to study on square kufic images since it is one of the common kufic types in Islamic architectural decoration [6]. Its letters are in the form of a square or a rectangle. There are geometric shapes consisting of various straight lines and angles in square kufic designs [8, 6, 32]. These straight or angular strokes and angles are elongated with angle 45 or 90 in order to compose different motives [3, 6, 7, 1].

It is always difficult to determine the meaning of a calligraphy composition, even more difficult if this composition is formed with square kufic scripts. As far as we observed, distinguishing the words and understanding the text in a square kufic design is even hard for a person whose native language is Arabic since it is difficult to discriminate different letters having the same glyph shapes.

On the other hand, there are some benefits of studying documentation, analysis and preservation of square kufic images. Firstly, indexing square kufic designs will give a chance to automate the process of surveying and recording the designs on buildings and other artifacts as well as to prepare the data necessary for further studies and analysis [1]. Moreover, automatic analysis of such images and developing the rules to recognize them, may help better understanding of their characteristics and design rules, and also it may lead to the automatic generation

[1] All images in Figure 4.1 are taken from [6].

of new designs [1].

This preliminary study aims to provide a helpful tool to develop automatic methods for the analysis of square kufic calligraphy images. Recognition of words from the calligraphic compositions, specifically from square kufic images will serve as a start point for further analysis and indexing of the whole text. Reading those calligraphic compositions will shade lights on a relatively unknown era of the history.

In this study, automatic identification of square kufic patterns is done with the aid of existing kufic patterns, by following a query-by-example retrieval schema. In this way, automatic morphological segmentation can be provided by breaking down kufic words or sentences into their minimal grammatical constitutes. Segmented words can be used for the indexing and decipherment of the whole text in a kufic design, which will give a chance to catalogue and browse kufic patterns.

In next section, we firstly give a brief description of the challenges of studying on square kufic images.

4.1.1 Challenges of square kufic patterns

Square kufic designs are in Arabic and Arabic is a cursive language, in which it is hard to match characters since they are connected. Also, characters may take different shapes according to their position in the sentence; being at the beginning, middle, at the end and in isolated form [16, 25, 45]. Most of the characters which share the same shape with each other are only distinguished by adding dots or zigzags, called as diacritics. Moreover, because of the consonantal nature of Arabic, vowels are frequently omitted [17]. These challenges coming from the nature of Arabic, makes matching square kufic characters hard.

Moreover, studying on square kufic calligraphy images has other challenges due to the calligraphic nature of square kufic, which makes the automatic analysis of square kufic images [1] more difficult.

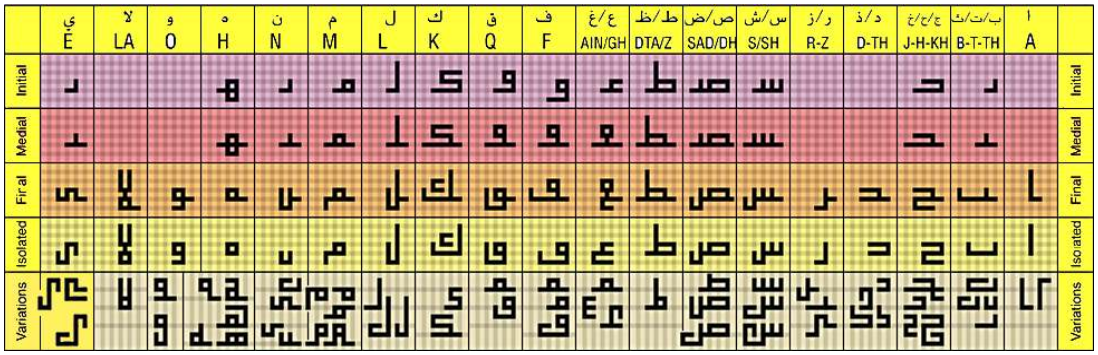


Figure 4.2: Square kufic script letters. Note that some characters are not in a single shape. Their shapes vary according to their position in the sentence because of the nature of Arabic language. Also, they may have different shapes in different designs because of the nature of kufic calligraphy. (Image courtesy of [1])

Firstly, the direction of the words in square kufic images can change, unlike the other calligraphy styles where the texts are written horizontally. The texts in square kufic calligraphy are written in a spiral way, starting from one corner of the edge and ending at the center of the image [6]. Since a single word can bend direction several times, it becomes challenging to recognize the same letters because bends may introduce new shapes. Although the design is generally like this, sometimes this rule is broken by zigzags crossing the design surface. In some square kufic designs, the shape of a square is maintained by designing repeated patterns around a square and at the center, these drawings form sided stars towards the middle [6]. Moreover, in square kufic designs, all the letters are made up of a few similar shapes so that there is very little distinction between the different letters. Even two letters may share the same shape in different designs. As it can be seen in square kufic alphabet (Figure 4.2) and in designs (Figure 4.3), there is a wide variety of appearance and shapes of same words in different designs, as well as different words may share the shape [6].

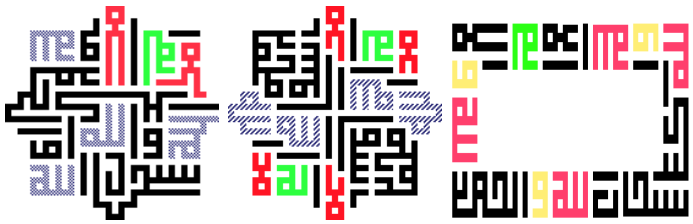


Figure 4.3: Same words in different shapes and different words in similar shapes. In the first two images, the gray, light gray and lighter gray sub-patters have different shapes in both designs. For example, the gray ones are different designs of the letter *La*. In the last image, the gray, light gray and lighter gray ones are different sub-patterns but they share similar shapes.

In square kufic, distinguishing two different words sharing the same shape becomes harder since these characters are generally written without diacritics, which help distinguishing the characters sharing the same shapes [1]. Also, since a calligrapher generally has to fill a specific space in a square kufic design, he/she is forced to modify the letters to fit the space whether by extending them or contracting them, or by changing their shapes [1]. This causes the distortion of these letters and makes the interpretation of the design more difficult [6]. Furthermore, instead of writing a letter more than once, it may be used by two different words in a design, as well as a word may be written just once and used by two different sentences in the same design.

All these challenges make the recognition and matching of Islamic patterns in square kufic images more interesting and difficult to deal with.

4.2 Related Work

In recent years, cultural heritage preservation has been a very important issue [39, 51, 91] and computer vision techniques are used for automatic indexing and

retrieval of 2-D imagery in cultural heritage [75]. In this sense, there are also studies in Chinese calligraphic images [84, 89, 90], as well as in Arabic calligraphy images.

Studies in Arabic calligraphy images generally focus on rendering, classification and generation of Islamic patterns in other calligraphy types [31, 27, 44, 62, 11, 14, 15, 28, 65, 81], and in a few studies automatic generation of square kufic images are tackled [61]. Based on our knowledge, matching of square kufic patterns have not been studied previously.

In [31], a method is described to generate a repeating pattern of the hyperbolic plane based on a tiling by any convex polygon. In this study, an algorithm is described to draw patterns based on tilings by a polygon which is not necessarily regular and that polygon is assumed to be convex. Moreover, a method based on radially-symmetric motives is proposed to generate Islamic Star Patterns [27, 44].

In [15], Aljamali and Banissi proposed a method to classify Islamic Geometric Patterns (IGPs) based on the minimum number of grids and lowest geometric shape methods for the construction of Star pattern, while IGP images are described using the discrete symmetry groups theory in the study [28]. Firstly, every pattern is classified into one of three major categories, in which classification is done based on the translation along one direction. Secondly, some symmetry features: the symmetry group and the fundamental region, are extracted. As a last step, the fundamental region is described by a color histogram.

In [14], comprehensive analysis and cataloguing of Islamic design patterns from digital images is done by the plane symmetry groups theory. By using image segmentation algorithms, these regular design patterns are then grouped by pixels to obtain the pieces forming tiles. After vector representation is done, the objects are compared according to their contours and then classified by their shape and color. In [62], a protype (Interactive Calligraphy Exploration) is described to compose calligraphic images by manipulating symmetries to produce unusual visual effects by dynamic calligraphy compositions. The approach introduces a novel method of interaction with compositional elements and demonstrates how controlled change propagation can be used to promote design exploration.

And lastly, in [61], cellular automata with extended Moor neighborhood is used to generate square kufic script patterns, specifically *Mohammed* patterns by defining some transition rules. Their approach focuses on three most famous *Mohammed* patterns, which makes the approach only specific to some models.

4.3 Our Approach

We observed that the straight lines in specific orientations are important for the identification of square kufic patterns. Our method is based on this observation and the relation between these horizontal and vertical lines are exploited in order to match patterns in square kufic designs. Proposed approach has the following steps: (i) Contours are approximated to lines. (ii) Sub-patterns in the images are automatically extracted. (iii-a) In the first method, sub-patterns are represented by chain codes and sub-pattern similarities are calculated by a string matching method. (iii-b) In the second method, sub-patterns are represented by shape context descriptors and sub-pattern similarities are calculated by matching shape context descriptors.

4.3.1 Preprocessing

Firstly, all images in the data set are blurred and then Otsu's binarization method [66] is used to binarize the images and noises are cleaned by removing connected components having an area size smaller than a threshold.

4.3.2 Line-Based Representation

We propose a shape representation based on lines to describe patterns in square kufic images. Firstly, contours are extracted from images and points on these contours are approximated to lines using Douglas-Peucker line approximation algorithm [30]. Slope angle and start-end coordinates of lines are used as line

features (see Figure 4.4).

In Arabic studies, also skeletons are used to represent the shape of an object [45]. In our approach, as an alternative to extracting the contours, we could also extract skeletons from binary images and then lines could be approximated from skeletons. We didn't prefer to study with skeleton images since skeleton representation would not be unique and be affected by noise.

4.3.3 Extraction of Sub-Patterns

Before pattern matching, all sub-patterns in the images are automatically detected. Connected lines which form a closed polygonal shape are extracted and these shapes are called as sub-patterns (see Figure 5.1). A sub-pattern is a connected group of letters and a sub-pattern may be an individual word or many sub-words may come together in order to form a word. As a result, shapes of sub-patterns are represented by connected lines, which have slope angle, start and end coordinates as features.

4.3.4 Pattern Matching

Chain Code Matching Method

In the first method, after sub-patterns are extracted, 8-connected chain code representation [35] is extracted for each sub-pattern by using line features. Length information of lines is not used in this step, thus proposed method is scale invariant. On the other hand, chain code representation is not rotation invariant. Because of this, a query pattern is searched in 8 different orientations of range [0-360], per 45 degrees, for matching it with patterns in candidate images.

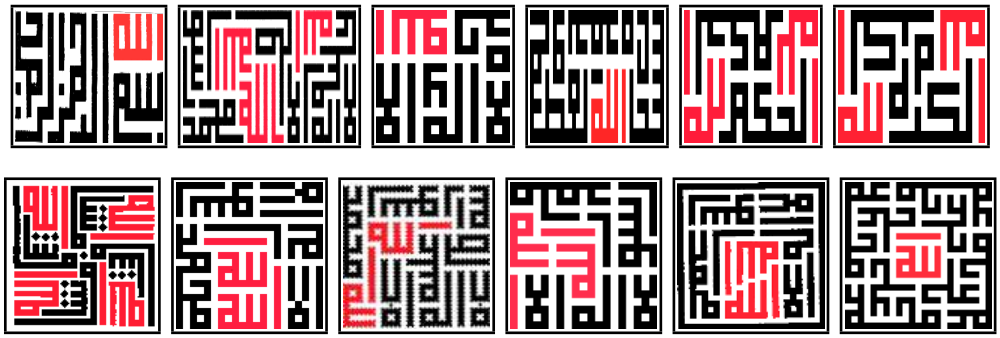
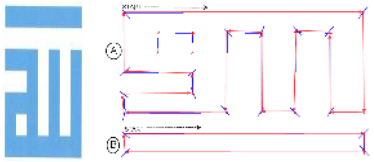


Figure 4.5: Some square kufic images with *Allah* patterns (in gray) are shown. Note that this word has different shapes in different designs.



(a) (b)

Figure 4.4: (a) Pattern *Allah* (b) Output of line simplification process. Start-end points of lines are shown with small lines. The chain code representation of sub-pattern A is 02161216121606616, while B's is: 0216.

Since chain code representation changes with the start point of the chain code extraction algorithm, we start to extract chain codes at the upper left corner of each sub-pattern in order to normalize the representation. In Figure 4.4, there are two sub-patterns[2] and their chain code representations. After each sub-pattern is represented by chain codes, sub-pattern similarities are calculated by using a string matching algorithm [63], in which chain codes are treated as strings.

[2]The image in Figure 4.4 is taken from [2].

Shape Context Method

In the second method, the well known shape context descriptor [20], which is used on a wide range of shape similarity problems in object recognition [20], is used in order to match sub-patterns. After all sub-patterns are extracted in images, sub-patterns are represented by their shape context descriptors. Generally, randomly selected points on the contours of shapes are used to extract shape context descriptors. But in this study, instead of selecting the points randomly, we use only the points on the intersections of connected lines (corners) to extract shape context descriptors for sub-patterns. In Figure 4.6, there is a sub-pattern and a shape context descriptor extracted on a point of this sub-pattern. A point is selected on the intersection of two lines and that point is connected to other corner points in order to obtain vectors. Distance and angle frequency histogram of these vectors is calculated in order to obtain the shape context descriptor at that point.

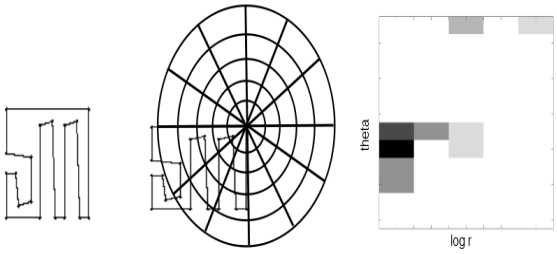


Figure 4.6: Shape context of a sub-pattern.

Chapter 5

Kufic Experimental Results

5.1 Dataset Description

Figure 5.1: Some examples from our square kufic data set. For example, the first image in second row has 4 *Allah* and *Mohammed* scripts, while the second image in the same row, has 4 *Masaallah*.

In this study, we prefer to study on square kufic images, because square kufic is one of the most common types in kufic calligraphy, so that a data set in a relatively larger size can be built. Since there is no existing square kufic data set, we constructed our data set by collecting images from Internet (mostly from [6]) and from a book on calligraphy [67]. In Figure 5.1, some square kufic image

examples[1] in our data set can be seen. In total there are 95 square kufic images in the constructed data set.

5.2 Matching a Query Pattern

After sub-pattern similarities are calculated, a query pattern is matched with other patterns as follows: Firstly, sub-patterns in query pattern are automatically detected. Then, a best match is found for each query sub-pattern among sub-patterns in a candidate image by searching query sub-pattern in 8 different orientations. Individual sub-pattern matching scores are summed in order to calculate the total pattern matching score. In the chain code matching based method, if total score is bigger than a threshold and in the shape context based method, if total score is smaller than a threshold, that candidate pattern is accepted as a match.

Note that we don't know how many sub-patterns a query pattern has. For example, as seen in Figure 4.5, the script *Allah* has 2 sub-patterns; first one is the letter *Alif* which resides individually and the other part is the combination of the remaining letters. Also, the script *Resul* is formed by 3 sub-patterns as it is shown in Figure 5.3, while *La ilahe illa allah* pattern has more sub-patterns as depicted in Figure 5.4. The sub-patterns in a query pattern are automatically extracted and this makes the proposed approach applicable to any query script. Below, there are experiments in order to find different instances of query patterns in candidate images.

5.2.1 Pattern *Allah*

Chain Code Matching based Method : We firstly perform experiments with *Allah* pattern queries, since this pattern is the most common word in our dataset. 68 candidate images, in which there is at least one instance of this pattern

[1] In Figure 5.1, the images (1,4-6) are taken from [67] and the second image is from [5], third is from [4].

	Category												
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
# of scripts	1	2	3	4	7	8	10	12	16	18	25	30	58
# of images	16	15	8	17	1	2	1	1	1	1	1	1	1
th_1	17	33	22	37	50	91	41	26	100	49	21	51	100
	77	95	71	78	1	92	99	98	97	89	57	98	100
th_2	24	38	31	47	89	40	0	32	100	61	46	47	100
	48	57	47	45	42	74	0	78	47	27	31	45	100
th_3	22	34	30	51	85	39	0	32	100	21	64	43	100
	31	38	35	33	23	69	0	64	26	2	25	25	100

Table 5.1: Average success rates (out of 100) for all *Allah* queries based on chain code matching method. For example, category 1 has 16 images each having only one *Allah* script and category 2 has 15 images each having 2 *Allah* scripts.

(see Figure 4.5 for examples[2] of this pattern.) are used in this experiment. In total, there are 334 *Allah* patterns in these images with varying number of *Allah* patterns in each image. As it is depicted in Table 5.1, there are at most 58 instances of this pattern in an image. There are 13 different categories based on the number of *Allah* patterns in the images. For example, category 1 has 16 images each having only one *Allah* pattern and category 2 has 15 images each having 2 *Allah* patterns.

In this experiment, given a query pattern and a matching score threshold, candidate patterns which have a matching score greater than this threshold are accepted as a match. In the experiments, three thresholds th_1=*a*(1/2)*, th_2=*a*(3/4)* and th_2=*a*(5/6)* are used in order to retrieve different instances of query patterns. These thresholds are set based on the perfect match score a, where a = *numOfLines*matchScore* and *numOfLines* is the number lines in a query pattern and *matchScore* is the score used in the string matching algorithm when two lines has the same chain code. Thus, we assume that the perfect match score is the ideal score when all lines in query and candidate patterns are matched. Precision and recall percentage rates are given in Table 5.1 for each category and with different threshold values. In category 7, since there is only

[2]In Figure 4.5, images 1,2,3 are taken from [1], 4-10 and 12 from [3].

one image having 10 instances of script *Allah*, and shapes of these patterns are very different from most of the queries, as the matching score threshold increases, retrieving patterns in this image gets harder. Also, average precision and recall rates for all of the *Allah* queries and for all categories are given in Table 5.2. Note that as the matching threshold increases, recall rate decreases since some of the *Allah* patterns can not be detected with bigger thresholds.

	Precision	Recall
th_1	0.50	0.81
th_2	0.49	0.50
th_3	0.47	0.36

Table 5.2: Average precision and recall rates in all categories for all 24 different *Allah* pattern queries.

Moreover, in Figure 5.2, average precision and recall rates are given for each 24 different *Allah* pattern queries. One of the reason in low success rate is the less common *Allah* pattern models. Since their shape is very distinct from the most of the instances of this pattern, false matchings are retrieved, while correct matchings are missed. Note that in Figure 5.2, the first four patterns give the highest scores since they are the most common *Allah* pattern models in the data set, while the last four pattern models are in fewer number of images.

The pattern *Allah* is confused with the pattern *lillah*, which is nearly the same as *Allah* word, except that *lillah* does not have the letter *Alif*. The third image in Figure 4.3, has a *lillah* pattern which is in gray.

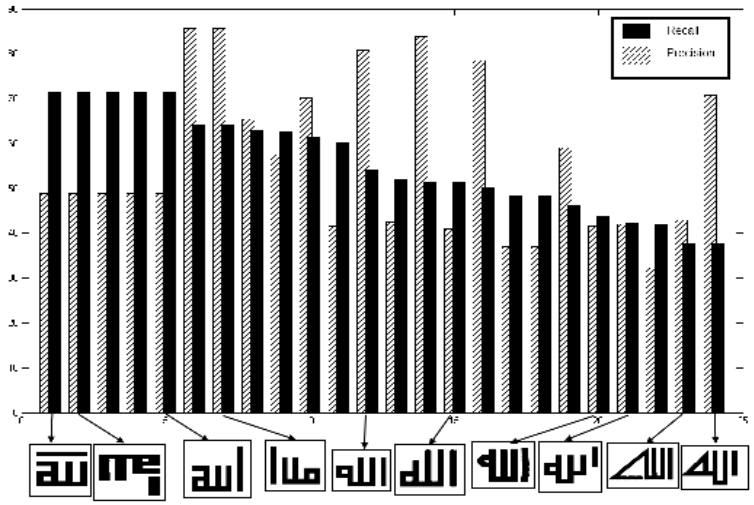


Figure 5.2: Average precision and recall percentage rates for each 24 different *Allah* query patterns with th_2 and some query patterns are shown.

Shape Context based Method : In order to test shape context based method with *Allah* pattern queries, we performed experiments with two different matching thresholds: i) th_1=0.3 ii) th_2=0.4. Average precision and recall rates for all queries are given in Table 5.3. As it can be compared with Table 5.2, chain code based method outperforms shape context based method. Although shape context may tolerate noise, it doesn't preserve the information of line connections, which chain code representation can easily do. Since first method is more successful, experiments will be done based on chain code matching method in next sections.

	Precision	Recall
th_1	0.41	0.32
th_2	0.55	0.55

Table 5.3: Average precision and recall rates for *Allah* pattern queries based on shape context method.

5.2.2 Pattern *Resul*

In order to show the performance with a different pattern query rather than *Allah* script, we evaluated our method with pattern *Resul*. Since our data set is not large, there are a few examples[3] of the script *Resul* in the data set (see Figures 5.3-5.4). Our method can successfully retrieve 24 instances of this pattern out of 38 with chain code matching method and threshold $th_2 = a*(3/4)$ is used in the experiment.

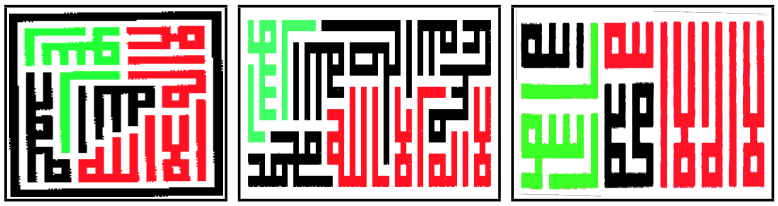


Figure 5.3: The patterns in light gray are *Resul* patterns, which are formed by 3 sub-patterns. The gray sub-patterns form the pattern *La ilahe illa allah*.

5.2.3 Pattern *La ilah illa allah*

Moreover, we performed experiments with another pattern *La ilah illa allah*, which is longer and composed of 7 sub-patterns. There are 4 different words in it (see Figures 5.3-5.4) and this pattern exists in 19 images[4] in the data set and in total there are 44 instances of it in these images. 28 instances of this pattern are detected out of these 44 patterns with chain code matching based method and threshold $th_2 = a*(3/4)$ is used in the experiment.

[3]First image in Figure 5.3 is taken from [5] and second one is from [6] and third one is from [67].

[4]The first image in Figure 5.4 is taken from [67], the second from [67] and third from [6].

Figure 5.4: The patterns in gray are *La ilahe illa allah* patterns. We can not detect the first one, since two sub-patterns are connected. Also the light gray patterns are *Resul* patterns and we can't detect the last one, since two sub-patterns are connected.

5.3 Categorization of images

Another experiment is to decide whether a given query pattern exist in a candidate image or not and all candidate images which have at least one instance of this query pattern, are retrieved. In this way categorization of images may be done based on the patterns inside them.

	TP	FP
th_1	0.90	0.15
th_2	0.76	0.12
th_3	0.66	0.09

Table 5.4: Success rates for finding the candidate images that have a given query pattern in itself.

This experiment is performed with *Allah* pattern queries and all images in the dataset are used, no matter these images have an instance of this pattern or not. If a query pattern is matched with any candidate pattern in an image, it is said that there exists an instance of that query pattern in that image. The success rates with three aforementioned thresholds in chain code matching based

method are given in Table 5.4. False positive rates are low since there exists a *Allah* pattern in most of the images.

5.4 Detecting Repeating Sub-patterns

In this experiment, without the usage of a query pattern, we automatically detect the repeating sub-patterns in a given image. Any sub-pattern that exists at least twice in an square kufic image, is accepted as a repeating sub-pattern. For example, in the first row[5] of Figure 5.5, the first image has 2 repeating patterns and others have more, since they are symmetric.

This experiment is performed in whole data set, based on chain code matching method and two threshold (th_1 and th_1) of chain code method are used to extract the repeating sub-patterns. After calculating the similarities of each sub-pattern with each other, the sub-patterns which have a matching score bigger than a predefined threshold, are accepted as repeating sub-patterns. The proposed method has true positive (TP) rate 72% and the FP error 25% with th_1 and TP rate 60% and FP rate 10% with th_2. The repeating sub-patterns which have different shapes in the same image can not be retrieved. For example, in Figure 4.5, in the second image of first row, there are three *Allah* pattern examples in gray. But since their shapes are different from each other, they can not be detected as repeating patterns.

Figure 5.5: Repeating pattern examples. All of the repeating patterns in these images are found.

[5]The images in Figure 5.5 are taken from [6].

The advantage of detecting repeating sub-patterns is that without the usage of a query pattern, we can automatically find the possible words in a given square kufic image. In this way, detecting the meaningful patterns will be deciphered in these calligraphic images and a fully automatic indexing schema for these images will be provided.

5.5 Discussion

In this task, we present a shape-based analysis of square kufic calligraphy images for search and retrieval tasks in these image collections. Proposed method is based on a line representation and patterns are matched with a chain code representation as well as a shape context matching method. We show that our line representation gives promising results in matching Islamic patterns in square kufic images.

Although our results are promising with most of the queries, some less common queries lead to lower success, since their shapes are different from the most of the patterns in the images. Also, since two different letters may share the same shape in a square kufic design, precision rates are small in the experiments. Our method can not retrieve instances of a query pattern when these instances are in very different models as it can be seen in Figure 5.6. Our method can not detect instances of *Mohammed* patterns, since there are in many different shapes in different designs.

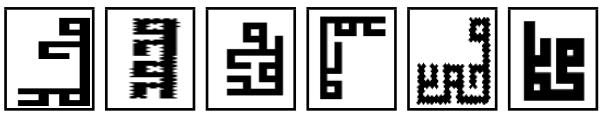


Figure 5.6: *Mohammed* patterns in different shapes and our method can not match them.

Other problem is the connected patterns, which have sub-pattern's within the other. There are some connected pattern examples[6] in Figure 5.7. In the first row, the first image has 2 *Allah* scripts, which are connected by their upsides with the purpose of decoration. The second image has 4 *Ali* scripts at each corner and these patterns are connected with each other. The last image has 4 *Allah* scripts, rotated in 4 directions. Their letter *Alif* is connected and forms a boundary. Our algorithm can't detect the patterns in these images.

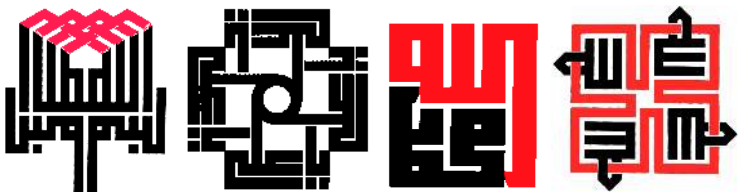


Figure 5.7: Connected pattern examples. Our method is not successful to detect them.

As a future work, we are planning to solve the connected sub-patterns problem by a sliding window approach which cuts patterns in each window and separate them from each other such that matching them becomes easier. Also, we are planning to extend our data set in order to study on different and longer words.

[6]The images in Figure 5.7 are taken from [67].

Chapter 6

Conclusion

In this thesis, word matching based methods are presented to tackle two historical document analysis problems, which are word segmentation in historical documents and Islamic pattern matching in square kufic images.

In the first task, a cross document word matching based method is proposed to segment historical documents into words. The method is based on the idea that word segmentation in some versions of documents is easier than their some other versions which are in different writing styles. By using automatically extracted words of a version, which is easier to segment into words, another version of the same document, in which word segmentation is harder, can be segmented into words. Experiments on an Ottoman data set consisting of a machine printed and a *lithography* printed versions show that a cross document word matching based method gives promising results to decide word boundaries in historical documents.

In the first task, rather than profile based features and dynamic time warping method, some other features would be used in order to match the words in cross documents. In this preliminary study, we firstly aimed to see how successful cross document word matching is with simple profile based features and in future, by using some other matching methods such as shape context, word matching success rates may be increased in order to improve the word segmentation success rates.

Also, some knowledge that comes from the rules of the document's language can be used in order to decide word boundaries in historical documents. For example, in Ottoman, the letter *Alif* is written in a different form when it is at the end of a word and by using this knowledge it is said that there is a word boundary after this letter whenever this letter takes a horizontal line shape. Since we wanted the proposed method to be language independent, we didn't prefer to use some language dependent knowledge, but language dependent knowledge may be also used in the proposed method in order to improve the word segmentation results.

Moreover, some other contextual information may be also used during word matching process. For example, in Ottoman divans, some words are likely to occur together in the whole text. A good example for this is the words *gül* and *bülbül* in Ottoman Divans. They may be matched as a pair in the document and also it may be said that there is a high possibility of an occurrence of the word *bülbül* when there is the word *gül* somewhere in the document.

In the second task, a line representation is proposed to match Islamic patterns in square kufic images using two methods: a chain code representation matching method and a shape context descriptor matching method. Experiments show that a line representation based method gives promising results in matching Islamic patterns in square kufic images.

Although, two proposed methods seems to be different, a query by example paradigm is employed in both tasks. In the cross document word matching based word segmentation task, query words are used to segment historical documents into words. In the second task, query patterns are used to find the instances of them in the images. In this way, kufic patterns are segmented into smaller units and words in calligraphy images are deciphered.

As a result, in these tasks, we tried to provide fast and retrieval schemas to access historical documents and in the future, these preliminary studies may be improved in order to provide automatic transcription of these documents, which will be helpful for many researches studying on these documents.

Bibliography

[1] The art of arabic calligraphy, 1993. www.sakkal.com/ArtArabicCalligraphy.html.

[2] Kufi kufi, 2009. www.kufikufi.blogspot.com/2009/04/kufi-kufi-squares.html.

[3] Kufic, 2009. en.wikipedia.org/wiki/Kufic.

[4] Kufic example 1, 2009. www.farm4.static.flickr.com/3377/3318123762_ea07344f17.jpg?v=0.

[5] Kufic example 2, 2009. www.waterholes.com/~dennette/1995/islam/shahada.htm.

[6] Kufic info, 2009. www.kufic.info/.

[7] Kuficpedia, 2009. www.kuficpedia.com.

[8] Maghribi kufic, 2009. calligraphyqalam.com/styles/kufic-maghribi.html.

[9] An Arabic document, 2011. http://www.yomiuri.co.jp/adv/chuo/dy/research/20100513.htm.

[10] Ottoman text archive project (OTAP), 2011. http://courses.washington.edu/otap/.

[11] S. J. Abas. Islamic geometrical patterns for the teaching of mathematics of symmetry. *Symmetry in ethnomathematics*, 12(1-2):53–65, 2001.

[12] T. Adamek, N. E. O'Connor, and A. F. Smeaton. Word matching using single closed contours for indexing handwritten historical documents. *International Journal on Document Analysis and Recognition*, 9:153–165, 2007.

[13] B. H. Al-Badr. A segmentation-free approach to text recognition with application to Arabic text, Ph.D. thesis. Seattle, WA, USA, 1995. University of Washington.

[14] F. Albert, J. M. Gomis, and M. Valor. Analysis and reconstruction of the tiling of Alcazar in Seville using computer vision tools. In *Proceedings of the 3rd International conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 127–130, New York, NY, USA, 2005.

[15] A. M. Aljamali and E. Banissi. Grid method classification of Islamic geometric patterns. *Geometric modeling: techniques, applications, systems and tools*, pages 234–254, 2004.

[16] A. Amin. Off line Arabic character recognition - A Survey. In *Proceedings of the 4th International Conference on Document Analysis and Recognition*, pages 596–599, Washington, DC, USA, 1997. IEEE Computer Society.

[17] A. Amin. Segmentation of printed Arabic text. In *ICAPR '01: Proceedings of the Second International Conference on Advances in Pattern Recognition*, pages 115–126, London, UK, 2001. Springer-Verlag.

[18] E. Ataer and P. Duygulu. Retrieval of Ottoman documents. In *Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 155–162, New York, NY, USA, 2006.

[19] E. Ataer and P. Duygulu. Matching Ottoman words: An image retrieval approach to historical document indexing. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 341–347, New York, NY, USA, 2007. ACM.

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

[21] C. D. Brina, R. Niels, A. Overvelde, G. Levi, and W. Hulstijn. Dynamic time warping: A new method in the study of poor handwriting. *Human Movement Science*, 27(2):242 – 255, 2008.

[22] M. Bulacu and L. Schomaker. Text-independent writer identification and verification using textural and allographic features. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29:701–717, April 2007.

[23] E. F. Can and P. Duygulu. A line-based representation for matching words in historical manuscripts. *Pattern Recognition Letters*, 32(8):1126 – 1138, 2011.

[24] E. F. Can, P. Duygulu, F. Can, and M. Kalpakli. Redif extraction in handwritten Ottoman literary texts. In *Proceedings of the 20th International Conference on Pattern Recognition*, ICPR '10, pages 1941–1944, Washington, DC, USA, 2010. IEEE Computer Society.

[25] J. Chan, C. Ziftci, and D. Forsyth. Searching off-line Arabic documents. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1455–1462, Washington, DC, USA, 2006. IEEE Computer Society.

[26] A. Cheung, M. Bennamoun, and N. W. Bergmann. An Arabic optical character recognition system using recognition-based segmentation. *Pattern Recognition*, 34(2):215 – 233, 2001.

[27] C. K. Department and C. S. Kaplan. Computer generated islamic star patterns. In *Proc. Bridges 2000: Mathematical Connections in Art, Music and Science*, page 4, 2000.

[28] M. Djibril and R. Thami. Islamic geometrical patterns indexing and classification using discrete symmetry groups. *Computing and Cultural Heritage*, 2008.

[29] M. N. Dogan. *Mecnun ve Leyla Dilinden Siirler*. Enderun Kitabevi, 1997.

[30] D. Douglas and T. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer 10(2)*, pages 112–122, 1973.

[31] D. Dunham. An algorithm to generate repeating hyperbolic patterns. In *Proceedings of ISAMA 2007*, pages 111–118, 2007.

[32] S. Etikan. The Use of the Kufic script, an Element of Islamic Ornament in Turkish Rug Art. *The Social Sciences 3(2)*, pages 104–112, 2008.

[33] A. Fornés, J. Lladós, and G. Sánchez. Old Handwritten Musical Symbol Classification by a Dynamic Time Warping Based Method. 5046:51–60, 2008.

[34] A. Fornes, J. Llados, G. Sanchez, and D. Karatzas. Rotation invariant hand-drawn symbol recognition based on a dynamic time warping model. *Int. J. Doc. Anal. Recognit.*, 13(3):229–241, 2010.

[35] H. Freeman. Computer processing of line-drawing images. *Computing Surveys*, pages 6(1):57–97, March 1974.

[36] P. Fung and K. Mckeown. Aligning noisy parallel corpora across language groups: Word pair feature matching by dynamic time warping. In *Proceedings of the First Conference of the Association for Machine Translation in the Americas*, pages 81–88, 1994.

[37] B. Gatos, A. Antonacopoulos, and N. Stamatopoulos. ICDAR 2007 handwriting segmentation contest. *9th International Conference on Document Analysis and Recognition (ICDAR'07)*, pages 1284–1288, 2007.

[38] B. Gatos and I. Pratikakis. Segmentation-free word spotting in historical printed documents. In *Proceedings of the 2009 10th International Conference on Document Analysis and Recognition*, ICDAR '09, pages 271–275, Washington, DC, USA, 2009. IEEE Computer Society.

[39] C. Grana, D. Borghesani, and R. Cucchiara. Picture extraction from digitized historical manuscripts. In *Proceeding of the ACM International Conference on Image and Video Retrieval*, pages 1–8, New York, NY, USA, 2009. ACM.

[40] N. R. Howe, T. M. Rath, and R. Manmatha. Boosted decision trees for word recognition in handwritten document retrieval. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 377–383, New York, NY, USA, 2005. ACM.

[41] C. Huang and S. N. Srihari. Word segmentation of off-line handwritten documents. *Document Recognition and Retrieval (DRR) XV*, page 68150, 2008.

[42] A. Jain. *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ:Prentice-Hall, 1986.

[43] A. Kabacali. *Cumhuriyet oncesi ve sonrasi matbaa ve basin sanayii*. Cem Ofset, 1998.

[44] C. S. Kaplan. *Computer graphics and geometric ornamental design*. PhD thesis, 2002.

[45] M. S. Khorsheed. Off-line Arabic character recognition a review. *Pattern Analysis and Applications*, 5(1):31–45, 2002.

[46] N. Kilic, P. Gorgel, O. N. Ucan, and A. Kala. Multifont Ottoman character recognition using support vector machine. In *EEE International Symposium on Control Communication and Signal Processing (ISCCSP'08)*, pages 328–333, 2008.

[47] S. Kim, S. Jeong, G.-S. Lee, and C. Suen. Word segmentation in handwritten Korean text lines based on gap clustering techniques. In *Sixth International Conference on Document Analysis and Recognition*, pages 189 –193, 2001.

[48] T. Konidaris, B. Gatos, K. Ntzios, I. Pratikakis, S. Theodoridis, and S. J. Perantonis. Keyword-guided word spotting in historical printed documents using synthetic data and user feedback. *Int. J. Doc. Analysis Recognition*, 9(2):167–177, 2007.

[49] L. M. Kruskal J.B. *The symmetric time-warping problem: from continuous to discrete*. Addison - Wesley Publishing Co, Reading, MA, 1983.

[50] T. Kut. *Yazmadan basmaya: Muteferrika, Muhendishane, Uskudar*. Yapi Kredi Kultur Merkezi, 1996.

[51] J. Landre, F. Morain-Nicolier, and S. Ruan. Ornamental letters image classification using local dissimilarity maps. In *Proceedings of the 2009 10th International Conference on Document Analysis and Recognition*, pages 186–190, Washington, DC, USA, 2009. IEEE Computer Society.

[52] Y. Leydier, A. Ouji, F. LeBourgeois, and H. Emptoz. Towards an omnilingual word retrieval system for ancient manuscripts. *Pattern Recognition*, 42(9):2089 – 2105, 2009.

[53] J. Lladós, P. Pratim-Roy, J. A. Rodríguez, and G. Sánchez. Word spotting in archive documents using shape contexts. In *Proceedings of the 3rd Iberian conference on Pattern Recognition and Image Analysis, Part II*, pages 290–297, Berlin, Heidelberg, 2007. Springer-Verlag.

[54] G. Louloudis, B. Gatos, I. Pratikakis, and C. Halatsis. Text line and word segmentation of handwritten documents. *Pattern Recognition*, 42(12):3169 – 3183, 2009.

[55] S. Madhvanath and V. Govindaraju. The role of holistic paradigms in handwritten word recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):149 –164, feb 2001.

[56] R. Manmatha, H. Chengfeng, and E. Riseman. Word spotting: A new approach to indexing handwriting. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 631 –637, jun 1996.

[57] R. Manmatha, C. Han, E. M. Riseman, and W. B. Croft. Indexing handwriting using word matching. In *Proceedings of the first ACM international conference on Digital libraries*, pages 151–159, New York, NY, USA, 1996. ACM.

[58] R. Manmatha and N. Srimal. Scale space technique for word segmentation in handwritten documents. In *Scale-Space Theories in Computer Vision*,

volume 1682 of *Lecture Notes in Computer Science*, pages 22–33. Springer Berlin / Heidelberg, 1999.

[59] A. Marcolino, V. Ramos, M. Ramalho, and J. C. Pinto. Line and word matching in old documents. In *Proceedings of the 5th IberoAmerican Symposium on Pattern Recognition (SIARP00)*, pages 123–125, 2000.

[60] U.-V. Marti and H. Bunke. Using a statistical language model to improve the performance of an HMM-Based cursive handwriting recognition system. *IJPRAI*, 15(1):65–90, 2001.

[61] S. A. H. Minoofam and A. Bastanfard. A novel algorithm for generating Mohammad pattern based on cellular automata. In *Proceedings of the 13th WSEAS international conference on Applied mathematics*, pages 339–344, 2008.

[62] H. Moustapha and R. Krishnamurti. Arabic calligraphy: A computational exploration. *Mathematics and Design*, pages 294–306, 2001.

[63] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, March 1970.

[64] N. Nikolaou, M. Makridis, B. Gatos, N. Stamatopoulos, and N. Papamarkos. Segmentation of historical machine-printed documents using adaptive run length smoothing and skeleton segmentation paths. *Image Vision Comput.*, 28(4):590–604, 2010.

[65] V. Ostromoukhov. Mathematical tools for computer-generated ornamental patterns. In *In Electronic Publishing, Artistic Imaging and Digital Typography. In Lecture Notes in Computer Science*, pages 193–223. Springer-Verlag, 1998.

[66] N. Otsu. A threshold grey scale histogram. *IEEE Trans. on Syst. Man and Cyber.*, pages 62–66, 1979.

[67] S. Ozpalabiyiklar. *Bir Yazi Sevdalisi: Emin Barin.* Yapi Kredi, 2002.

[68] A. Ozturk, S. Gunes, and Y. Ozbay. Multifont Ottoman character recognition. In *7th IEEE International Conference on Electronics Circuits and Systems (ICECS)*, pages 945–949, 2000.

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

[70] T. M. Rath, S. Kane, A. Lehman, E. Partridge, and R. Manmatha. Indexing for a Digital Library of George Washington's Manuscripts: A Study of Word Matching Techniques. Technical report, 2002.

[71] T. M. Rath, V. Lavrenko, and R. Manmatha. A statistical approach to retrieving historical manuscript images without recognition. Technical report, 2003.

[72] T. M. Rath, R. Manmatha, and V. Lavrenko. A search engine for historical manuscript images. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 369–376, New York, NY, USA, 2004. ACM.

[73] T. M. Rath, R. Manmatha, and V. Lavrenko. A search engine for historical manuscript images. In *Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval*, pages 369–376, New York, NY, USA, 2004. ACM.

[74] J. A. Rodríguez-Serrano and F. Perronnin. Handwritten word-spotting using hidden Markov models and universal vocabularies. *Pattern Recognition*, 42(9):2106–2116, February 2009.

[75] E. Roman-Rangel, C. Pallan, J.-M. Odobez, and D. Gatica-Perez. Analyzing ancient Maya glyph collections with contextual shape descriptors. *International Journal of Computer Vision*, 10 2010.

[76] J. L. Rothfeder, S. Feng, and T. M. Rath. Using corner feature correspondences to rank word images by similarity. *Computer Vision and Pattern Recognition Workshop*, 3:30 –36, June 2003.

[77] E. Saykol, A. Sinop, U. Gudukbay, O. Ulusoy, and A. Cetin. Content-based retrieval of historical Ottoman documents stored as textual images. *IEEE Transactions on Image Processing,*, 13(3):314 –325, 2004.

[78] G. Seni and E. Cohen. External word segmentation of off-line handwritten text lines. *Pattern Recognition*, 27:41–52, 1994.

[79] S. N. Srihari and G. R. Ball. Language independent word spotting in scanned documents. In *Proceedings of the 11th International Conference on Asian Digital Libraries*, pages 134–143, Berlin, Heidelberg, 2008. Springer-Verlag.

[80] Y.-H. Tseng and H.-J. Lee. Recognition-based handwritten Chinese character segmentation using a probabilistic viterbi algorithm. *Pattern Recognition Letters*, 20(8):791–806, 1999.

[81] M. Valor, F. Albert, J. M. Gomis, and M. Contero. Textile and tile pattern design automatic cataloguing using detection of the plane symmetry group. *Computer Graphics International Conference*, 0:112, 2003.

[82] T. Varga and H. Bunke. Tree structure for word extraction from handwritten text lines. In *8th International Conference on Document Analysis and Recognition*, volume 1, pages 352 – 356, 2005.

[83] L. G. Vuurpijl and R. Niels. Dynamic time warping: An intuitive way of handwriting recognition. Master's thesis, 2004.

[84] S. T. S. Wong, H. Leung, and H. H. S. Ip. Model-based analysis of Chinese calligraphy images. *Comput. Vis. Image Underst.*, 109(1):69–85, 2008.

[85] I. Yalniz, I. Altingovde, U. Gudukbay, and O. Ulusoy. Integrated segmentation and recognition of connected Ottoman script. *Optical Engineering*, 48(11):1–12, 2009.

[86] I. Yalniz, I. Altingovde, U. Gudukbay, and O. Ulusoy. Ottoman archives explorer: A retrieval system for digital Ottoman archives. *J. Computing on Culture Heritage*, 2(3):1–20, 2009.

[87] M. Zand, A. Naghsh, and A. Monadjemi. Recognition-based segmentation in Persian character recognition. In *Proceedings of the Second International Conference on Advances in Pattern Recognition*. World Academy of Science, Engineering and Technology 38, 2008.

[88] B. Zhang, S. N. Srihari, and C. Huang. Word image retrieval using binary features. *Document Recognition and Retrieval XI*, (1):45–53, 2003.

[89] X.-F. Zhang, Y.-T. Zhuang, J.-Q. Wu, and F. Wu. Hierarchical approximate matching for retrieval of Chinese historical calligraphy character. *J. Comput. Sci. Technol.*, 22(4):633–640, 2007.

[90] Y. Zhuang, N. Jiang, and H. Hu. A web-based search engine for Chinese calligraphic manuscript images. In *Proceedings of the 8th International Conference on Advances in Web Based Learning*, pages 464–476, Berlin, Heidelberg, 2009. Springer-Verlag.

[91] B. Zitova, J. Flusser, and F. Sroubek. An application of image processing in the medieval mosaic conservation. *Pattern Anal. Appl.*, 7(1):18–25, 2004.