Pattern Recognition Letters 32 (2011) 1126–1138

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

A line-based representation for matching words in historical manuscripts

Ethem F. Can *, Pınar Duygulu

Department of Computer Engineering, Bilkent University, Bilkent 06800, Ankara, Turkey

ARTICLE INFO

Article history:
Received 2 December 2009
Available online 24 February 2011
Communicated by S. Sarkar

Keywords:
Historical manuscripts
Word image matching
Word retrieval
Word spotting
Line-based representation

ABSTRACT

In this study, we propose a new method for retrieving and recognizing words in historical documents. We represent word images with a set of line segments. Then we provide a criterion for word matching based on matching the lines. We carry out experiments on a benchmark dataset consisting of manuscripts by George Washington, as well as on Ottoman manuscripts.

1. Introduction

With the increase in the number of historical texts available in the digital environment, efficient access to these valuable documents has become crucial. Manual indexing of documents is costly, however, and can be carried out only in limited amounts; therefore automatic systems need to be built to make the ever-growing content available to users. There are various issues in the analysis of historical documents including enhancement of degraded documents, artifact removal, layout analysis, text line and word segmentation, recognition and retrieval (Antonacopoulos and Downton, 2007).

Following the long history of optical character recognition (OCR) (Suen et al., 1980; Impedovo et al., 1991; Amin, 1997; Plamondon and Srihari, 2000; Khorsheed, 2002; Cheriet et al., 2009) there are now plenty of OCR systems available for various languages (Kanungo et al., 1998; Chang et al., 2009). On the other hand, when historical documents are considered recognition of characters continues to be an active research area (Govindaraju et al., 2009).

Inspired by cognitive studies that have observed the human tendency to read whole words at a time (Madhvanath and Govindaraju, 2001), word-spotting techniques have been recently proposed to access historical documents as an alternative to character-based systems. In these studies, the words rather than the characters are considered as the basic units and the need for performing character segmentation and recognition is eliminated by considering the words as a whole. Word spotting has gained more interest with the work of Manmatha et al. applied on manuscripts by George Washington held in the Library of Congress (Manmatha et al., 1996).

The common approach in word spotting is to first segment documents into words, and then locate all the instances of a word image in the documents by means of word-matching techniques so that the results can be used for word-retrieval or word-recognition purposes.

The representation and matching of words continue to be challenging problems for word spotting. In this study we address the challenges and propose a simple but effective method to resolve them. Going beyond the George Washington dataset, which has become a benchmark in the word spotting literature, by applying our method on Ottoman documents provided in (Ataer and Duygulu, 2006), we also address the challenge of working on different alphabets and different writing styles (see Fig. 1 for sample lines from those documents).

Starting from the idea that words consist of lines and curves (the latter of which can also be approximated by lines) and inspired by the work in (Ferrari et al., 2008) where encouraging results are obtained by using line segments as descriptors for object recognition, we describe words by using line segments extracted from the contours of words images. Then, the distances between the line descriptors of the images determine the degree of similarity of the images.

The main contributions of this paper can be summarized as follows: (a) we propose an effective and efficient representation of word images based on line descriptors, (b) a new word-matching criterion using pairs of matched line descriptors, (c) we apply our method not only on English, but also on Ottoman documents without the need for complicated pre-processing or post-processing steps specific to the language or document type, and (d) we approach to word matching in a multi-scaled way by employing line approximations at different scales.

* Corresponding author. Tel.: +90 312 2903143; fax: +90 312 2664047.
E-mail addresses: efcan@cs.bilkent.edu.tr (E.F. Can), duygulu@cs.bilkent.edu.tr (P. Duygulu).

0167-8655/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2011.02.013

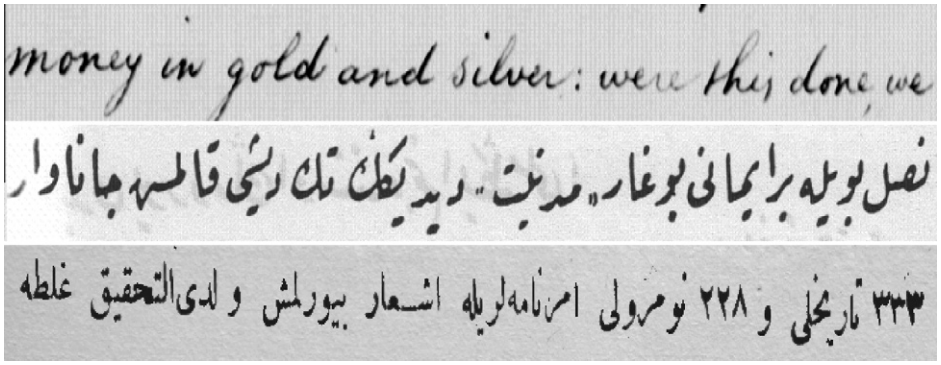

Fig. 1. Sample lines of words from the collections used in the study. The top row is a sample line from documents in English, the middle row from printed Ottoman documents, and the last row is from handwritten Ottoman documents.

In the following, we first review related studies in the literature, discussing the advantages and disadvantages of those methods. Then we present our approach and offer a detailed explanation of the proposed method. Finally, we provide extensive analysis of the proposed approach on different datasets, followed by a comparison of ours with the other studies in the literature and a discussion of our results.

2. Related work

In the studies of Manmatha et al. (1996), Rath and Manmatha (2003a,b), with the assumption that multiple instances of a word are written similarly by a single author, words are represented by simple image properties, such as projection profiles, word profiles, or background/ink transitions. Compared to other techniques such as sum of squared differences (SSD), and Euclidean distance mapping (EDM), dynamic time warping (DTW) is shown to be the best method for matching words. In (Balasubramanian et al., 2006), they similarly use DTW to match words in printed documents using profile-based and structural features. The DTW-based methods are successful in matching exact words with small variations in handwriting. The DTW-based partial matching method in (Meshesha and Jawahar, 2008) is proposed also for morphological variants of words. Three types of features are exploited: word profiles, moments and transform domain representations. The main issue with DTW-based studies is the complexity of running time. Kumar et al. (2007) makes use of the locality sensitive hashing (LSH) technique for increasing the speed, and focus on documents in Indian.

Although word matching is mostly used for retrieval, in a more recent study (Rath and Manmatha, 2007) it is used for clustering to recognize words. In (Rothfeder et al., 2006), an HMM-based method is proposed to align segmented words with transcriptions.

In (Adamek et al., 2007), in order to eliminate the limitations of profile-based or structural features that depend on slant angle and skew normalizations, Adamek et al. propose a contour-based approach to match the image words. They extract the contours of the image after several processes, including binarization with adaptive pixel-based thresholding, as well as removing artifacts (e.g. segmentation errors) and diacritical marks, and produce a single closed contour. Then they employ the multi-scale convexity concavity (MCC) representation, which stores the convexity/concavity information and utilizes DTW for matching.

In (Srihari et al., 2005), CEDARABIC system is presented for spotting Arabic words written by multiple writers. On manually segmented word images, the words are retrieved for a given query using the gradient based binary features described in (Zhang et al., 2004). Similar methods are also applied on English and Sanskrit documents in (Srihari et al., 2006; Srihari and Ball, 2008). In (Ball et al., 2006), in order to handle the problems of automatic word segmentation, which is especially prone to error on Arabic documents, a segmentation-free method is proposed as an alternative to the methods that require words to be segmented. The query words are searched over sliding windows on segmented text lines.

In (Leydier et al., 2007), they use gradient angles as features and variations of elastic distance. They search for a template word in the whole document without requiring segmentation; this prevents errors caused by segmentation; however, speed remains a problem for this study as well. In their following study, the method is generalized for word retrieval in order not to tackle with segmentation, and applied on different languages, specifically on Latin, Arabic and Chinese manuscripts (Leydier et al., 2009). For each character a model is selected from the documents. Supported with rules specific to the language, the characters in a word are searched over the unsegmented documents using zones of interest.

In (Rodriguez-Serrano and Perronnin, 2009), they propose a statistical framework on a multi-writer corpus. The authors make use of the continuous hidden Markov model (C-HMM) and semi-continuous hidden Markov model (SC-HMM) and demonstrate that their method outperforms DTW-based approaches for word-image distance computation.

Focusing on printed Greek documents Konidaris et al. (2007) propose an algorithm for word spotting that creates synthetic data and incorporates user feedback in retrieval. In (Bhardwaj et al., 2009), a script independent keyword spotting, based on image moments, is proposed and applied on Sanskrit documents. In (Rothfeder et al., 2003) word images are matched based on the corresponding interest points. The other studies on word spotting and retrieval include (Terasawa et al., 2006; Sankar and Jawahar, 2006; Llados et al., 2007).

In (Ataer and Duygulu, 2007), the words are treated as if they were objects in images. The authors extract interest points from word images by using the scale invariant feature transform (SIFT) operator (Lowe, 2004). A codebook obtained by the vector quantization of SIFT descriptors is then used to represent and match the words. The method is tested on Ottoman documents.

There are a few other recent studies focusing on Ottoman. In (Saykol et al., 2004), symbols are extracted and kept in a shape codebook, to be used for querying word images in Ottoman documents. An extended version is presented in (Yalniz et al., 2009b). A combined character segmentation and recognition system is proposed in (Yalniz et al., 2009a) to be used for retrieval of printed Ottoman documents.

3. Proposed method

Our proposed approach requires word images to be extracted from document images. Segmentation of a document image into words, which usually follows a text-line extraction step, is an

important and difficult task (Manmatha and Srimal, 1999; Marti and Bunke, 2001; Feldbach and Tonnies, 2003; Srihari et al., 2005; Likforman-Sulem et al., 2007; Zahour et al., 2007; Arivazhagan et al., 2007; Ouwayed and Belaid, 2009; Louloudis et al., 2009; Kurniawan et al., 2009; Kumar et al., 2010). In this study, we use the word images provided with the datasets experimented on, and do not address the segmentation. In the rest of the paper, the segmented word images will be considered.

The proposed method consists of four steps: extraction of lines from word images, description of lines, line matching, and word matching (see Fig. 2). In what follows, these steps are described in detail.

3.1. Line extraction

In the first step, lines comprising the words are extracted by means of binarization, contour extraction, and line approximation, as described below. In Fig. 3, the results of each step are presented for sample word images.

- *Binarization:* Most existing studies employ complex and costly preprocessing steps. In this study, we focus on the representation and matching of words and do not want the pre-processing steps to dominate the method. Therefore, we apply only binarization which is an essential part of most methods. Recall that binarization is performed on segmented word images, and therefore variations within an image are tolerable.
  Binarization is not a straightforward task, especially in the case of historical documents, which are usually degraded and heavily affected by noise. Although there are a variety of binarization methods available in the literature (see He et al., 2005; Gupta et al., 2007; Stathis et al., 2008; Moghaddam and Cheriet, 2010) for comparative evaluations on historical documents) it is difficult to have an objective evaluation criterion to choose the best one, and the performance of an algorithm may change from one document type to another.
  While local and adaptive methods are likely to perform better, our preference is not to fine-tune a specific binarization method for the datasets at hand, thus, we employ only a basic binarization method, one based on thresholding. The threshold value is computed as the mean intensity value of the gray-scale image. The Otsu method (Otsu, 1979), which is shown to be comparable to the complex methods on historical documents (Gupta et al., 2007), is also experimented with, and similar performances are obtained.
- *Extraction of contour segments:* As the next step, the connected components are found using eight-neighbors and the contour segments are extracted from these connected components using OpenCV library (Bradski, 2000). A single word is likely to consist of multiple contour segments because of noise factors resulting in wrong segmentation and due to diacritical marks as in Arabic and Ottoman or the holes inside the characters. Also, due to the variations in handwriting, the word images of a single word instance may have different number of contours extracted. We do not apply any postprocessing to obtain a single contour as in (Adamek et al., 2007), but make use of the list of contour segments extracted which are then approximated by lines as explained in the following.
- *Line approximation:*
  Polygonal approximation, for the description of the boundaries as a sequence of straight lines, is commonly used for shape representation (Marji and Siy, 2004; Carmona-Poyato et al., 2010; Parvez and Mahmoud, 2010). While most of the polygonal approximation methods are shown to perform well on

Fig. 3. Line extraction process on sample words. (a) Original gray-scale word images. (b) Binarized forms of the original gray-scale word images. (c) Contour segments extracted from the binarized forms of the images. (d) Approximated lines on the points of contour segments ($\tau = 3.0$). Note that, a single word may consist of multiple the contour segments, and the holes inside a character usually correspond to separate contour segments.

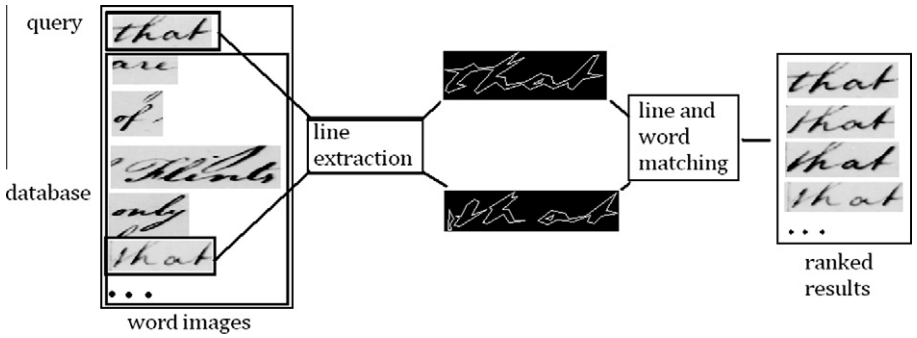

Fig. 2. Given the word images for the entire collection, the first step is to extract the lines from the contour segments on the word images. Considering a pair of word images, lines in one word image are compared with the lines in the other word image using the line-based descriptors. Then a similarity score for each pair of word images is computed based on the matching lines. Given a query word image, the most similar word images are ranked based on these scores.

simple shapes, when the handwritten characters/words are considered, the problem becomes more difficult (Parvez and Mahmoud, 2010). First of all, due to the high level of noise factors in historical documents, the contours are not smooth. The same character/word can be written in various ways, resulting in differences in the number and type of contour segments. Therefore, it is necessary to consider the shapes with variations in size and orientation and with different levels of details in different parts, and to allow partial matching (Marji and Siy, 2004).

In this study, we approximated the points on the contours into lines using the DouglasPeucker algorithm as a popular and standard method (Agarwal and Varadarajan, 2000) (see Algorithm 1).

Algorithm 1. Pseudo code of line approximation on contour segments.

input: points on contour segments and τ
output: ζ
Let $C = \{c_1, c_2, \ldots\}$ *is extracted contour segments;*
ζ *is the set of approximated lines on contour segments;*
τ *is the approximation accuracy;*
$\zeta = \emptyset$;
foreach *contour segment* $c_i \in C$ **do**
  ψ_i = points on c_i;
  ζ_i = Douglas–Peucker (ψ_i, τ);
  $\zeta = \zeta \cup \zeta_i$;
end

The Douglas–Peucker algorithm was first proposed in (Douglas and Peucker, 1973) and improved by Hershberger and Snoeyink (1992) in terms of the worst-case running time from the quadratic form in n to $n\log_2(n)$ where n is the number of points.

The Douglas–Peucker algorithm reduces the number of points in a curve by approximating it by a series of points. First, between a start and an end point, a sequence of points is approximated with a line segment. If the distance of the farthest point from the line is less than a threshold, the algorithm stops, otherwise it recursively divides the line into two from the farthest point (Heckbert and Garland, 1997).

The parameter τ used in the Douglas–Peucker algorithm can be defined as approximation accuracy, tolerance value, or compression factor. It serves for the determination of key points when fitting lines into points.

The greater values of τ result in a smaller number of lines and sharper segments, while smaller values of τ result in a greater number of lines and smoother segments. The effect of τ, which is in pixel units, is illustrated in Fig. 4.

We follow the studies proposed for analyzing a contour at different scales and for approximating it in a multiscale representation. For this purpose, we combine the results of different τ values,which allows us to capture the details at different levels, and also to perform partial matching. The errors due to noise factors at the finer levels can be compensated for at the coarser levels, while important details can still be preserved. In Section 4.6, the effect of different τ values on word retrieval will be explained in detail.

In the literature, there are non-parametric techniques available (Carmona-Poyato et al., 2010; Marji and Siy, 2004) to eliminate the need for parameter selection in line approximation process. The Douglas–Peucker algorithm was chosen since it is a standard and popular method in line approximation, and can be replaced with the others which are likely to produce better performances when single τ values are considered. The main contribution of our approach is to take the advantage of combining the results of different parameters, and therefore to be an alternative to the methods that optimize for a single best value.

3.2. Line description

We describe a line ℓ using the position, orientation, and length information as in (Ferrari et al., 2008):

$$\ell = \{p_s, p_m, p_e, \theta, \rho\}. \tag{1}$$

As illustrated in Fig. 5, $p_s = (x_s, y_s)$ is the start point, $p_m = (x_m, y_m)$ is the mid-point, $p_e = (x_e, y_e)$ is the end point, θ is the orientation, and ρ is the length of the line ℓ.

Each word image I is then represented as a set of line descriptors, as $I = \{\ell_1, \ell_2, \ldots, \ell_N\}$, where N is the number of lines approximated for the word image. We normalize the line descriptors of each word image by rearranging the positions of the lines depending on the location of the center point of the word image (X, Y). Then, representative points of each line descriptor are re-arranged to translate the points to word frame coordinates.

We use $p'_m = (x_m - X, y_m - Y)$ to represent the position of a line in a word image and refer to it as r.

3.3. Line matching

In order to find a matching score, we first find the distances between the line descriptors of the images. The distance between the two line descriptors, ℓ_a and ℓ_b, are computed by following the dissimilarity function as in (Ferrari et al., 2008):

$$d(\ell_a, \ell_b) = 4d_r + 2d_\theta + d_l, \tag{2}$$

where $d_r = |r_a - r_b|$, $d_\theta = |\theta_a - \theta_b|$, and $d_l = |\log(\rho_a, \rho_b)|$.

The first term is the difference of the relative positions of the mid-points of the lines (r_a and r_b). The second term is the difference between the orientations of the lines, where θ_a, $\theta_b \in [0, \pi]$. The third term is the logarithmic difference between the lengths of the lines (ρ_a and ρ_b).

3.4. Word matching

Having a criterion for determining the similarity of a pair of line descriptors, we propose a new matching technique for finding the similarity of words using the line segments.

The dissimilarity of two word images, I^a and I^b, which are described as $I^a = \{\ell_1^a, \ell_2^a, \ldots, \ell_3^a\}$ and $I^b = \left\{\ell_1^b, \ell_2^b, \ldots, \ell_3^b\right\}$ are then computed based on the values $d\left(\ell_i^a, \ell_j^b\right)$, where $i = 1,2,\ldots,N_a$ and $j = 1,2,\ldots,N_b$. For each line ia in I^a, we search for the best matching line ℓ_j^b in I^b by finding the line with the minimum distance (i.e. maximum similarity). That is, $\left(\ell_i^a, \ell_j^b\right)$ is a matching pair if $d\left(\ell_i^a, \ell_j^b\right) < d\left(\ell_i^a, \ell_k^b\right) \forall_k,\ j \neq k,\ k = 1,2,\ldots,N_b$. If two or more lines in I^a match a single line in I^b then we choose the one with the minimum distance and eliminate the others. The final distance between two images is then computed as the sum of the dissimilarity score of some of the best matches. The dissimilarity score is defined below:

$$(D_{a,b}) = \sum_i d\left(\ell_i^a, \ell_j^b\right), \tag{3}$$

where $\ell_j^b = match(\ell_i^a)$.

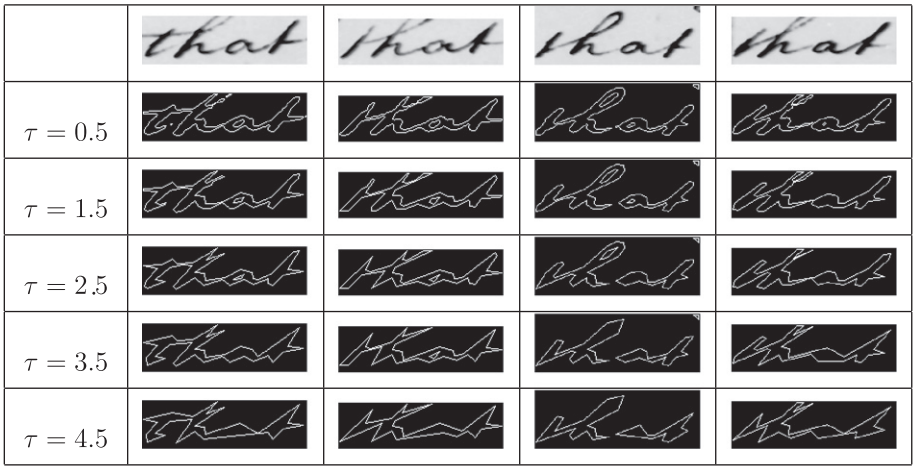

Fig. 4. Representation of the lines fitted into the points of the contour segments on our instances of the word that, for τ values 0.5, 1.5, 2.5, 3.5, and 4.5, respectively. Note that, different instances of a word may have different number of contour segments, and different number of extracted lines due to the writing style. There may be noise inside the segmented word images, or parts of the word may be cut due to wrong segmentation. Most of these problems are handled by extraction of lines in different levels of details.

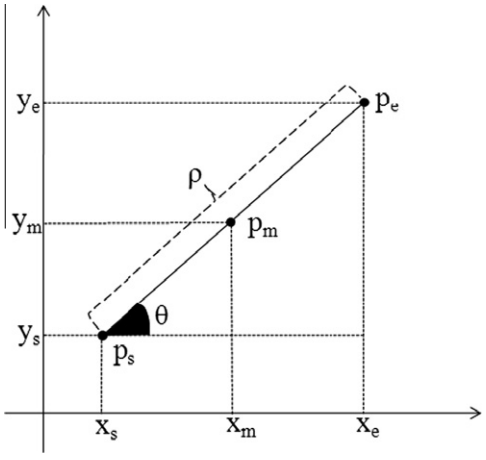

Fig. 5. Start point (p_s), mid-point ($p_m = r$), end point (p_e), orientation (θ), and length (ρ) of a line that is approximated on the points of a contour segment.

Considering the example given in Fig. 6; $I^a = \{\ell_1^a, \ell_2^a, \ell_3^a\}$ and $I^b = \{\ell_1^b, \ell_2^b, \ell_3^b, \ell_4^b\}$ and the minimum matches are $\{(\ell_1^a, \ell_3^b), (\ell_2^a, \ell_2^b), (\ell_3^a, \ell_2^b)\}$ in this case the total dissimilarity value of I^a and I^b is computed from the matches as $D_{a,b} = d(\ell_1^a, \ell_3^b) + min(d(\ell_2^a, \ell_2^b), d(\ell_3^a, \ell_2^a))$. Note that $D_{a,b} \neq D_{b,a}$.

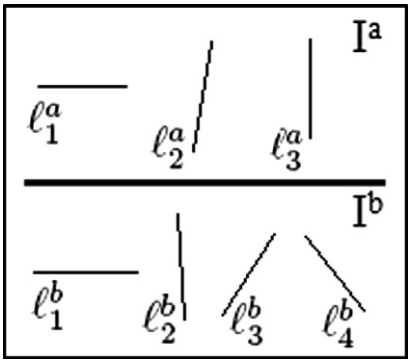

Fig. 6. Illustration of matching pairs of line descriptors of the images I^a and I^b to compute the dissimilarity score.

In order to compute the final score $\mathbf{f}(I^a, I^b)$ between the images I^a and I^b, instead of using only $D_{a,b}$, the sum of the total distances of the matched line descriptors, we consider other values as well: the number of hits $h_{a,b}$, as the number of matches between two images (in the example above, the number of hits is 2, $h_{a,b}$, = 2), and the number of lines in the images N_a and Nb. We normalize the dissimilarity value $D_{a,b}$, between two images I^a and I^b as defined in Eq. (4).

$$f(I^a, I^b) = (D_{a,b}) \left(\frac{(N_a - h_{a,b})^2 + (N_b - h_{a,b})^2}{\sqrt{[(N_a)^2 + (h_{a,b})][(N_b)^2 + (h_{a,b})^2]}} \right). \quad (4)$$

The equation above changes the value $D_{a,b}$, so that images with a small difference between the number of line descriptors and the number of hits have more chance of being matched than images in which the difference is greater.

Finally, we construct a global distance matrix **F** with the size of $Q \times Q$, where Q is the number of word images in the test bed, using f (I^a, I^b) values which are the dissimilarity values between the images, so that $F(a,b) = f(I^a, I^b)$. For instance, $F(1,3)$ is the dissimilarity value between the first and third images in the dataset.

The only parameter introduced in our approach, τ, has an important role in determining the lines in the approximation process. In other words, for different values of τ, the points of the contour segments are approximated into lines in different scales.

In order to combine two or more results at different tolerance values, for a multi-scale approach, we simply sum the matrices, such that $F' = F_{\tau=\tau_1} + F_{\tau=\tau_2}$, where the matrix F' is the distance matrix constructed by combining the distance matrices for $\tau = \tau_1$ and $\tau = \tau_2$.

4. Experimental results

4.1. Datasets

While there are several datasets used for evaluating handwriting recognition, in the case of historical documents there are only a few datasets available due to the difficulties in line and word segmentation and time-consuming ground truth generation which usually requires an expert (Fischer et al., 2010).

In this study, we focus on two types of datasets used in previous studies, for which segmented word images and annotations are

available: namely, George Washington (GW) datasets and Ottoman (OTM) datasets.

A benchmark in word-spotting literature, GW datasets are the subsets of the collection of George Washington's manuscripts held at the Library of Congress. The documents have been segmented into words by Manmatha and Srimal (1999) and labeled. The first set, hereafter referred to as GW10, consists often pages with 2381 words and was also used in (Rath and Manmatha, 2003a,b; Rothfeder et al., 2003), and the second set, hereafter referred to as GW20, consists of 20 pages with 4860 words and was also used in (Adamek et al., 2007; Rath and Manmatha, 2003b). The documents are of acceptable quality, however, some word images have artifacts or do not have any words at all due to segmentation errors (see Fig. 7).

In order to test the effectiveness of our approach on documents with different alphabets, especially on those with diacritical marks, we also use the Ottoman datasets provided by Ataer and Duygulu (2006, 2007). The first set consists of 257 words in three pages of text (hereafter referred to as OTM1) and the second one consists of 823 words in six pages of text (hereafter referred to as OTM2). In order to test the proposed method on documents with different styles, a third set is constructed: the combination of OTM1 and OTM2 (hereafter referred to as OTM1 + 2). While the documents in OTM2 are printed, those in OTM1 are handwritten. OTM1 is written with a commonly encountered calligraphy style called Riqqa, which is used in official documents. While simple projection profile based approaches were used for line and word segmentation on the printed documents, handwritten documents were manually segmented. Again, the documents are of acceptable quality; however, the segmented images have artifacts (see Fig. 8).

We should note that, our focus is on representation of words after segmentation, and therefore in this study we choose to use the available word images without applying any post-processing and do not consider any other segmentation method. Better methods are likely to result in better retrieval performances.

We should also mention that, word segmentation errors can be tolerated with the proposed approach. For example, for the cases where a single word is segmented into multiple parts, when two words are combined to form one word, or when words have artifacts inside due to touching lines or touching words, the subparts will be still matched with the original word with relatively high matching scores (see Figs. 11 and 13).

Here we focus on GW10 dataset and provide some statistics for discussing the variations in the datasets used. In this dataset there are 2381 word images corresponding to 933 distinct words. While there are 641 words appearing only once, there is a word which occurs 120 times (see Fig. 9). Although the height variations are small, the widths of the words vary, and more importantly the variations for different instances of a single word can be large (see Fig. 10(a)). The number of contour segments extracted from the word images vary from 1 to 25, and it is usually slightly correlated with the width of the word images (see Fig. 10(b)).

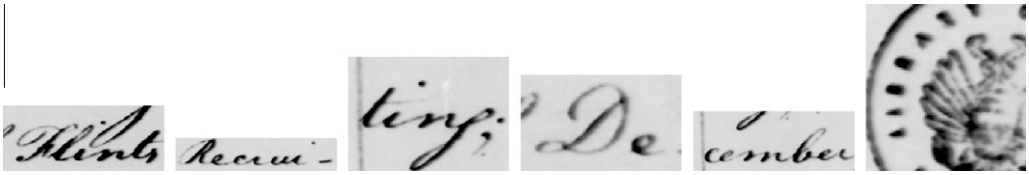

Fig. 7. Word images from the George Washington (GW) datasets.

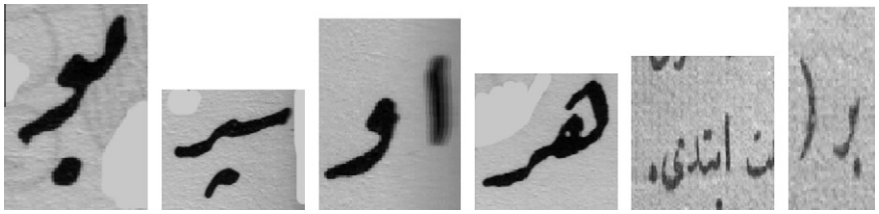

Fig. 8. Word images from Ottoman datasets.

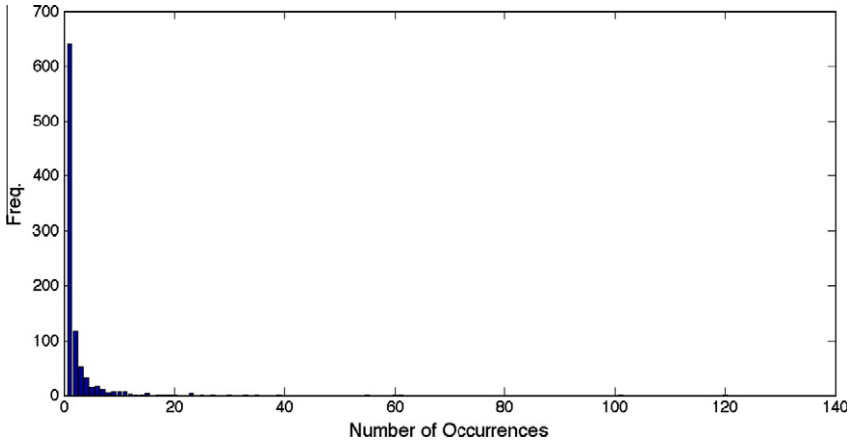

Fig. 9. Word frequency distribution for GW10 dataset.

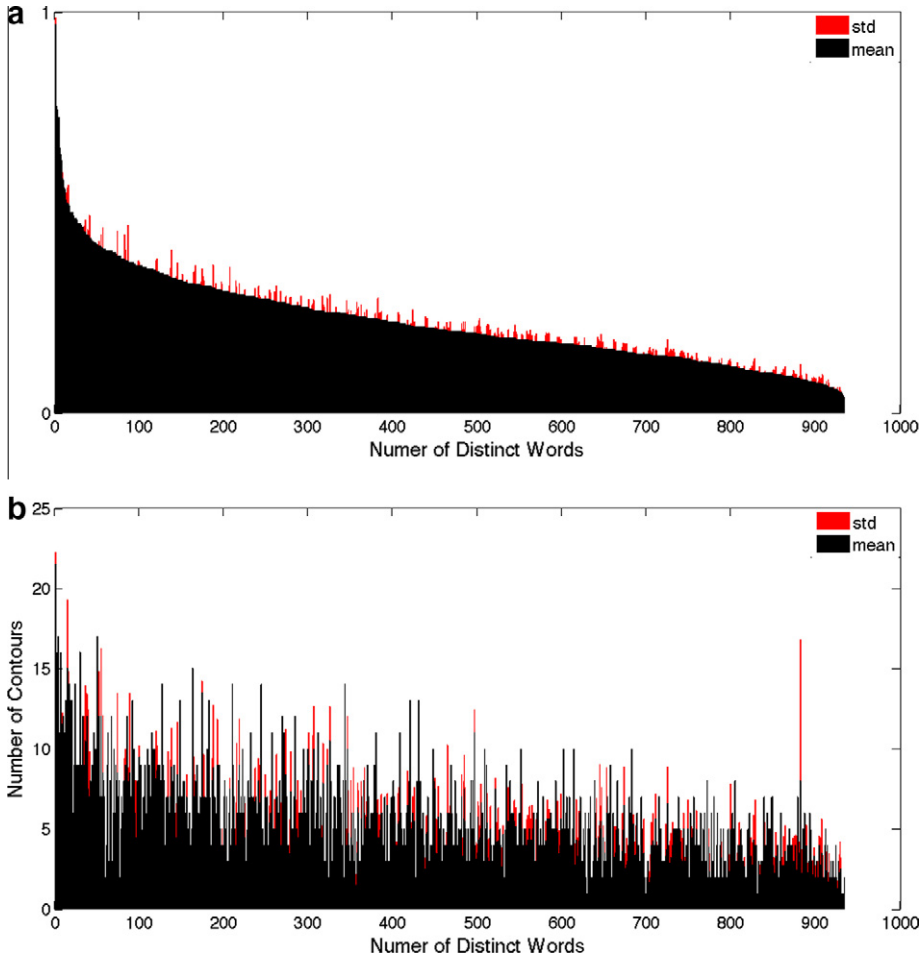

Fig. 10. (a) For distinct words the mean (shown in black) and standard deviation (shown in red) of the width of the word images corresponding to that word instance in sorted order. (b) For the same words, the mean and standard deviation of the number of extracted contour segments. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The number of contour segments for word images corresponding to a single word instance may vary, with standard deviation value up to around 9. On some selected words we observe that the orientations of the words are in the range of 30–40°, and the variations among different word images of the same word are in the range 6–8°. The proposed method tolerate these small variations in orientation and size. Our focus is on documents by single author, and we expect larger variations to rarely happen.

The following subsections cover the results of the experiments provided separately for the GW and OTM datasets.

4.2. Results for GW datasets

In Fig. 11 we provide the retrieval results for the keywords "December", "Instructions", "should", and "1755." and show the first 10 matches. Note that the results retrieved by the algorithm for the keyword "should" display character mismatches and the queries "December", and "1755." also yield some partially matching results.

For the query of "December" five exact matches, two partially matches ("Vc.Decembe" and "Decembe"), and two false matches ("Recruits" and "Buckner") are retrieved. The two partially matched words are almost the same as the query word. As the line characteristics of the false matches are very close to the lines of the query word, our method retrieves these words in the initial ranks. Similarly, in the query of "1755." our method retrieves partially matched words as well as exact matches. Eight words out often exactly match, whereas one word "3,1755" partially matches the query word. The situation holds for other queries such as "particular-particularly", "he-the", "you-your", "recruit-recruits", and 'me-men".

In Fig. 12 the word-rank representation of the GW10 set is provided. The queries appearing in Fig. 12 are for words that have forty or more relevant images in the dataset. Our method manages to retrieve most of the relevant images in the initial ranks, with the result that few images remain to be retrieved in the following ranks – a situation depicted as a large white area occupying most of the image, beginning from the right side, and darkening to all black on the left side.

4.3. Results of Ottoman datasets

In Fig. 13 the retrieval results of the query for the keyword "bu" (meaning "this"), sought in the OTM1 + 2 set, is displayed. Note that the images have different sizes.

In Fig. 14, the word-rank representation of the OTM1 + 2 dataset is provided. Our method manages to retrieve most of the relevant images in the initial ranks, with the result that few images remain to be retrieved in the later ranks.

4.4. Evaluation criteria

In our study we mainly focus on the task of retrieval; therefore, the results are mostly provided in terms of precision scores and analyzed for the task of retrieval. Some studies test their methods

Fig. 11. The first 10 retrieval results for querying the keywords "December", "Instructions", "should", and "1755." in the GW10 set. The order is top to bottom and the images in the topmost position are the keywords.

Fig. 12. The word-rank representation from left to right for the words in GW10 that have forty or more relevant images. Each row represents a query for a different word. A black point means a correct match,and a blank means a wrong match. Note that most of the black points are close to left meaning that the relevant images are retrieved earlier.

in terms of recognition rate, and thus, in order to compare our results with those studies we also provide recognition rates.

In order to obtain the precision and recall values we use the trec_eval package provided by the National Institute of Standards and Technology (NIST), which is a common tool used in the literature. All the precision values given in this study are the average precision scores computed using `trec_eval`, as in (Rath and Manmatha, 2003b).

We also use the score of word error rate to compare our results with other studies that provide WER. In most of those studies, researchers use 20-fold cross validation by choosing the number of folds as the number of pages. In other words, the words on one page are tested against words on other pages to compute the recognition rates. The final recognition rate is provided as the average of the recognition rates of each iteration in the cross-validation process. For each page the recognition rate is computed by taking the ratio of the total number of correct recognitions and the total number of words on that page. Word error rate is computed for the words in a test page as follows:

$$WER = 1 - \left(\frac{\#correct\ mathches\ intest\ page}{\#words\ in\ test\ page} \right).$$

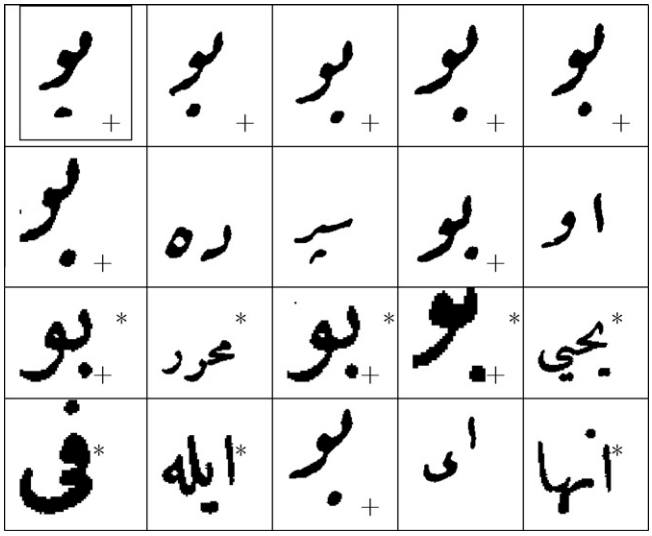

Fig. 13. The first 20 retrieval results for querying the keyword "bu (this)" in the OTM1 + 2 set. The order is top to bottom, left to right. The image on the top left position is the keyword. Images with a plus sign are correct matches. Images with a star sign are from the OTM2 set and the others are from OTM1.

In order to determine the number of correct matches on a test page, one-nearest neighbor approach is used. We provide two different types of WER; the first one considers the out-of-vocabulary (OOV) words, and the latter does not consider OOV words; a word is called an "Out of Vocabulary" word when the word appears on the test page but not on the other pages.

The best precision score obtained for the GW10 set is 0.688 and for the GW20 set is 0.566. The WER for GW20 – to compare with previous studies employing WER in testing their methods – is 0.303 when considering OOV words, and 0.189 when disregarding OOV words.

For the sets in the Ottoman language the best scores we obtain are 0.987 and 0.944 for OTM1 and OTM2, respectively. The highest precision score we obtain on OTM1 + 2 is 0.957.

4.5. Evaluation of the parameter τ

As mentioned in Section 3, we deploy the parameter τ in a multi-scale setting. First, we evaluate the effect of individual τ values by varying τ between 0.5 and 5.0, with an increment of 0.5. The precision scores for different r values of the GW10, GW20, and OTM1 + 2 collections are given in Fig. 15. Empirically, we find that the highest precision scores are obtained when $r = 2.5$: 0.638 and 0.523 for the GW10 and GW20 sets, respectively and 0.931 for the OTM1 + 2 set. However, the differences in the performances are minimal for different τ values, and therefore any τ value within the above mentioned range is acceptable. We observe that results of τ values greater than 5.0 display lower precision and recognition rates. For this reason, we do not consider the results of those τ values.

For testing the effectiveness of the multi-scale approach, we combine the results of different τ values by summing the dissimilarity scores (see Table 1). We empirically test different weighting schemes while adding these scores.

Then, since we observe no significant change, we decided not to use any weighting at all.

We observe that combining the results of individual τ values allows us a multi-scale approach and helps to obtain higher precision scores and recognition rates than using the distance matrices individually. While combining all scales is helpful in capturing all details and eliminating errors, we observe that a subset of levels, either by sampling over the scales or by considering a few consecutive ones only, also provides similar results. Although in the rest of the paper we report the best results, fine-tuning is not required for finding a specific value, rather a sample subset sufficient to capture different levels of details is all that is needed.

4.6. Analysis of the proposed method

Our matching technique considers not only the total dissimilarity value, but also the number of hits and number of lines in the images. The motivation behind considering parameters other than the dissimilarity value is that the number of lines varies between word images; this situation may alter the total dissimilarity value. Considering the other factors helps to obtain a better similarity criterion between the images. For example, a precision score of 0.415 is obtained on the GW10 test for $\tau = 2.5$ using only the dissimilarity value, whereas when other factors are considered the precision score turns out to be 0.638.

Our approach of line approximation runs in $m \cdot n\log_2(n)$, where m is the number of contour segments having more points than zero and n is the number of points on that contour segment. Matching the two word images requires the time $O(kN_aN_b)$, where N_a and N_b are the number of line descriptors for the images and k is the number of τ results combined. After the line descriptors obtained, the matching of lines takes time in the order of seconds. However, we should note that our consideration is not the efficiency and

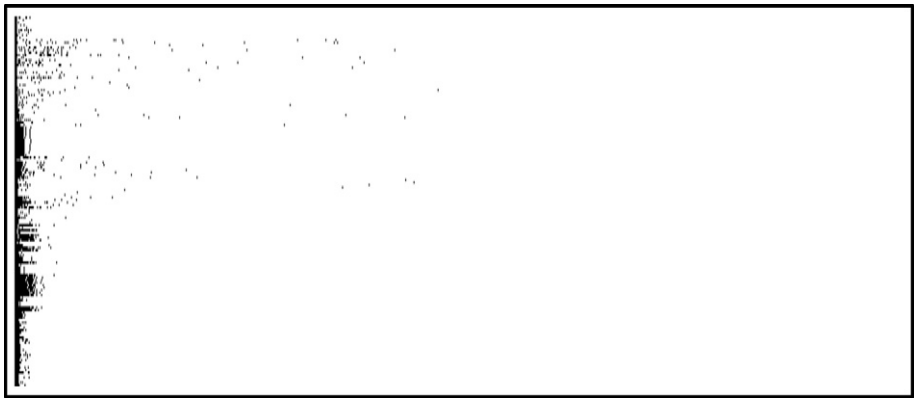

Fig. 14. The word-rank representation from left to right for the words in the OTM1 + 2 that have five or more relevant images. Each row represents a query for a different word. A black point means a correct match,and a blank means a wrong match. Note that most of the black points are close to left meaning that the relevant images are retrieved earlier.

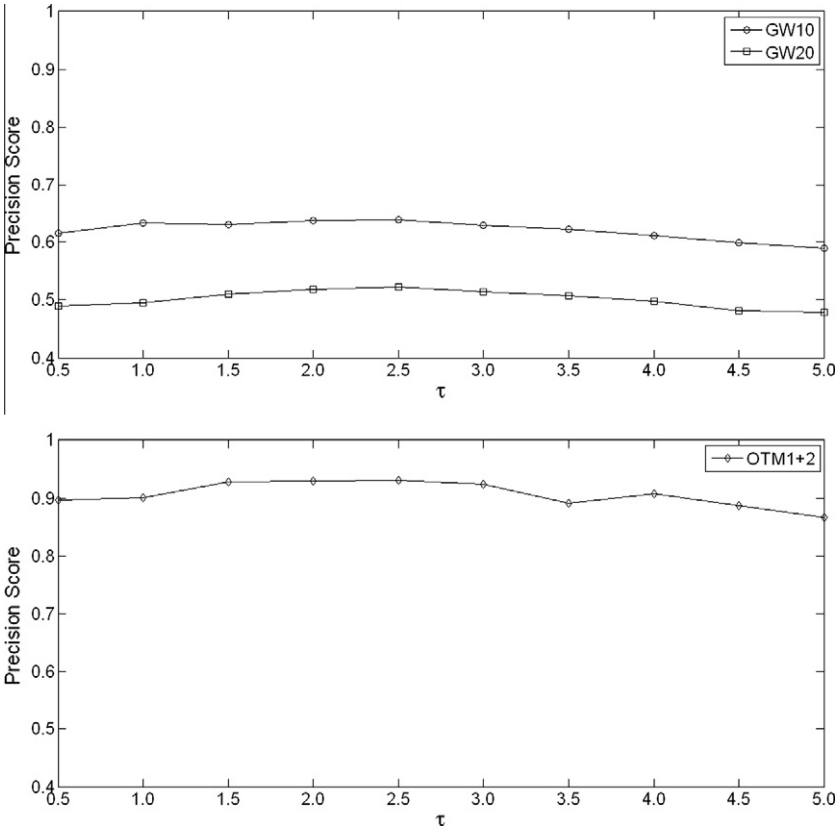

Fig. 15. The precision scores for different τ values. *Top*: the results on the GW10 and GW20 collections, and *bottom*: the results on the OTM1 + 2 collection.

Table 1
Precision scores (P) of some of the experiments that combine various τ values for the GW10, GW20, and OTM1 + 2 sets. Recall scores are always 1.0. Highest precision scores are obtained in the row with a †. Even though the combination of more τ values provides higher precision scores, some sample combinations (as in the rows with ♢) yield closer results.

τ										GW10	GW20	OTM1 + 2
0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	P	P	P
✓	✓									0.652	0.534	0.940
✓	✓	✓								0.662	0.549	0.937
✓	✓	✓	✓							0.673	0.535	0.939
✓	✓	✓	✓	✓						0.679	0.543	0.947
✓	✓	✓	✓	✓	✓					0.683	0.547	0.950
✓	✓	✓	✓	✓	✓	✓				0.686	0.551	0.952
† ✓	✓	✓	✓	✓	✓	✓	✓			0.688	0.566	0.957
✓	✓	✓	✓	✓	✓	✓	✓	✓		0.688	0.564	0.957
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	0.687	0.565	0.956
♢✓	✓	✓				✓				0.675	0.542	0.955
♢✓	✓			✓				✓		0.675	0.541	0.948
♢✓		✓		✓			✓			0.684	0.549	0.950
♢	✓	✓			✓	✓			✓	0.680	0.545	0.948
♢✓	✓	✓		✓		✓			✓	0.686	0.552	0.953

therefore we did not apply any method for improving the speed of the process.

The proposed method does not handle rotation invariance; however, we empirically test that our method can handle the rotation invariance of [−19, 24] degrees for GW sets, and [−14, 18] degrees for OTM sets. In order to find these numbers, we manually rotate the words and compute the distance between the original image and the rotated images, and then we check the distance between the rotated images and first image (not rotated) from querying the original word. The limit degrees provided above are the average values of each rotation test.

Next, we provide comparisons with other studies for the retrieval and recognition tasks.

4.7. Comparisons with other studies for the task of retrieval

In Table 2 we provide our results as well as the results of the existing studies in terms of precision-recall scores. We carry out experiments using all words in the collections; therefore, we provide precision scores in which the recall scores are 1.0. The studies providing a recall value lower than 1.0 include a pruning step that eliminates a set of likely wrong matches by analyzing different

Table 2
Precision scores of our and the other approaches. OTM1 + 2: the combination of OTM1 and OTM2 datasets.

Method	Dataset	Precision	Recall
Our approach	GW10	0.688	1.000
Our approach	GW10	0.774	0.770
DTW (Rath and Manmatha, 2003b)	GW10	0.653	0.711
DTW (Rath and Manmatha, 2003a)	GW10	0.726	0.652
Our approach	GW20	0.566	1.000
Our approach	GW20	0.667	0.673
DTW (Rath and Manmatha, 2003a)	GW20	0.518	0.550
Our approach	OTM1	0.987	1.000
Bag-of-words (Ataer and Duygulu, 2007)	OTM1	0.910	1.000
DTW (Ataer and Duygulu, 2007[a])	OTM1	0.940	1.000
Our approach	OTM2	0.944	1.000
Bag-of-words (Ataer and Duygulu, 2007)	OTM2	0.840	1.000
Our approach	OTM1 + 2	0.957	1.000
Bag-of-words (Ataer and Duygulu, 2007)	OTM1 + 2	0.810	1.000

[a] Ataer and Duygulu (2007) provide their own implementation of DTW for the OTM1 set.

criteria such as aspect ratio – a process that requires extra effort and runs tests on smaller sets, these studies therefore, obtain low recall values (the scores of our approach are the results of the experiments when considering the τ values stated in the row with a *j* sign in Table 1). Even though we do not include the pruning step, we provide our precision scores where the recall scores are similar to previous studies to better compare our study with such studies. For this purpose, we only take into account the first *x* percent of the retrievals. Precision and recall scores for different *x* values in the GW10 and GW20 collections are shown in Fig. 16. In Table 2, precision and recall scores in the second row are computed while considering the first 5% of the retrievals. This provides a recall score that is close to the recall score in (Rath and Manmatha, 2003b) for GW10 and therefore allows a better comparison. We keep the percentage of data same for GW20.

Table 3
Results of our and other methods in terms of WER for GW20 set.

Method	WER	WER w/o OOV words	Language model post-processing
Our approach	0.303	0.189	–
Adamek et al. (2007)	0.306	0.174	–
Lavrenko et al. (2004)	0.449	0.349	+

The precision score of our approach for the GW10 set is 0.688, with a recall score of 1.0. Rath and Manmatha (2003b) obtain 0.653 as the precision score. Our approach is better than theirs in terms of the precision score. However, the same authors obtain higher precision scores with lower recall scores in another study (Rath and Manmatha, 2003a). Regarding the precision scores, that study has better results than our method, in which the recall score is 1.0; however, when we consider the precision score of our study, with a recall score of 0.770, it is better than that study as well.

In the GW20 set, we obtain a precision score of 0.566 when the recall score is 1.0, and 0.667 when the recall score is 0.673. In both cases, our results turn out to be better than the results of the other studies (Rath and Manmatha, 2003b,a).

Ataer and Duygulu (2007) run their method on the OTM1 and OTM2 sets. They also compare their algorithm with the DTW method. Our method performs better than theirs as well as better than

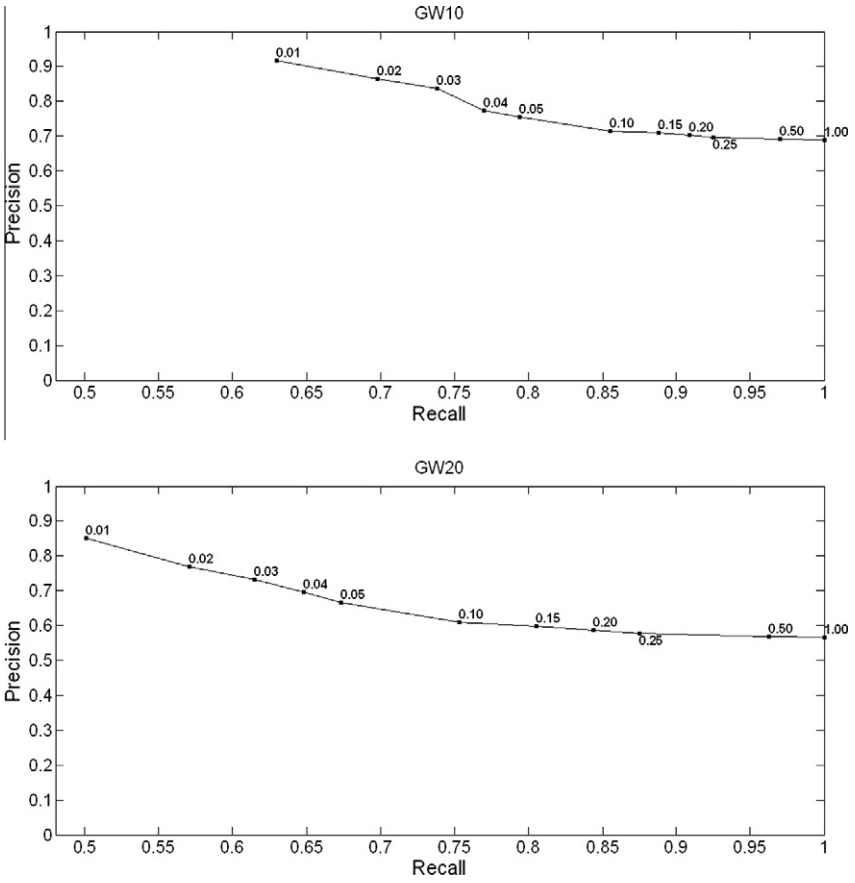

Fig. 16. Precision-recall scores for different *x* values in the GW10 and GW20 sets.

their implementation of DTW method on the OTM1 and OTM2 sets and also on the OTM1 + 2 set which is created to test the script independence.

4.8. Comparisons with other studies for the task of recognition

In Table 3, the WER with and without OOV words yielded by our method as well by other studies are given for the GW20 set.

Our results are better than the work in (Lavrenko et al., 2004) in terms of WER with and without OOV. Note that, in (Lavrenko et al., 2004) they use a language model post-processing, in (Adamek et al., 2007) it is stated that removing the language model post-processing causes a dramatic decrease in the recognition rate.

Adamek et al. (2007) provides the results in the form of WER as 0.306 and 0.174. Their score excluding OOV words is better than the score of our method, whereas our rate is better than their score in the experiments including OOV words. However, since they require a single closed contour, the work in (Adamek et al., 2007) does not work on scripts in which diacritical marks are important, as is the case with Ottoman. Moreover, their method depends on complex preprocessing steps that require additional time and effort, before matching the word images. Our implementation of the MCC–DCT algorithm without the preprocessing steps provides lower rates.

5. Summary and discussion

In this study, we propose an efficient and effective line-based word spotting method that provides high precision scores without requiring complicated pre-processing or post-processing efforts.

We make use of line descriptors to represent the word images. Further, we incorporate the use of the number of hits and the total number of line descriptors in the images, together with the similarity values of matching line descriptors in order to compute matching scores between words. We also take the advantage of combining the results of different parameters in the line approximation process to deal with slight variations.

We test our method on documents in English and also on two different scripts in Ottoman. The partial matching capability of our method is promising for capturing morphological variants of words encountered in Ottoman.

The current study requires the word images to be provided. However, line and word segmentation is prone to error on historical manuscripts especially for the documents on Arabic and Ottoman. While our method allows words to be matched even in the case of incomplete data or data containing error, in the future, we plan to search for a query image over the entire document in order to eliminate the need for word segmentation and use visual word codebooks as an initial step in order to speed up the matching process, encouraged by our preliminary study on small number of Ottoman divans (Can et al., 2010).

Acknowledgements

The authors thank Rana Nelson for proof reading. This work is supported by TUBITAK with the project number 109E006.

References

Adamek, T., O'Connor, N.E., Smeaton, A.F., 2007. Word matching using single closed contours for indexing handwritten historical documents. Internat. J. Document Anal. Recognition 9, 153–165.

Agarwal, P.K., Varadarajan, K.R., 2000. Efficient algorithms for approximating polygonal chains. Discrete Comput. Geom. 23, 273–291.

Amin, A., 1997. Off line arabic character recognition – A survey. In: Proc. 4th Internat. Conf. Document Analysis and Recognition, pp. 596–599.

Antonacopoulos, A., Downton, A., 2007. Special issue on the analysis of historical documents. Internat. J. Document Anal. Recognition 9 (2), 75–77.

Arivazhagan, M., Srinivasan, H., Srihari, S., 2007. A statistical approach to line segmentation in handwritten documents. In: Proc. Document Recognition and Retrieval XIV SPIE, vol. 6500, pp. 135–142.

Ataer, E., Duygulu, P., 2006. Retrieval of ottoman documents. In: Proc. 8th ACM Internat. Workshop on Multimedia Information Retrieval, pp. 155–162.

Ataer, E., Duygulu, P., 2007. Matching ottoman words: An image retrieval approach to historical document indexing. In: Proc. 6th ACM Internat. Conf. on Image and Video Retrieval, pp. 341–347.

Balasubramanian, A., Meshesha, M., Jawahar, C., 2006. Retrieval from document image collections. In: Proc. Conf. on Document Analysis Systems, pp. 1–12.

Ball, G.R., Srihari, S.N., Srinivasan, H., 2006. Segmentation-based and segmentation-free methods for spotting handwritten arabic words. In: Internat. Workshop on Frontiers in Handwriting Recognition (IWFHR-10), La Baule, France, pp. 53–58.

Bhardwaj, A., Setlur, S., Govindaraju, V., 2009. Word spotting for indic documents to facilitate retrieval. In: Guide to OCR for Indic Scripts, pp. 285–299.

Bradski, G., 2000. The OpenCV library. Dr. Dobb's J. Software Tools.

Can, E.F., Duygulu, P., Can, F., Kalpakli, M., 2010. Redif extraction in handwritten ottoman literary texts. In: 20th Internat. Conf. on Pattern Recognition (ICPR 2010).

Carmona-Poyato, A., Madrid-Cuevas, F.J., Medina-Carnicer, R., Munoz-Salinas, R., 2010. Polygonal approximation of digital planar curves through break point suppression. Pattern Recognition 43 (1), 14–25.

Chang, Y., Chen, D., Zhang, Y., Yang, J., 2009. An image-based automatic arabic translation system. Pattern Recognition 42 (9), 2127–2134.

Cheriet, M., Yocoubi, M.E., Fujisawa, H., Lopresti, D., Lorette, G., 2009. Handwriting recognition research: Twenty years of achievement and beyond. Pattern Recognition 42 (12), 3131–3135.

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

Feldbach, M., Tonnies, K.D., 2003. Word segmentation of handwritten dates in historical documents by combining semantic a priori-knowledge with local features. In: Internat. Conf. Document Analysis and Recognition, vol. 1, p. 333.

Ferrari, V., Fevrier, L., Jurie, F., Schmid, C., 2008. Groups of adjacent contour segments for object detection. IEEE Trans. Pattern Anal. Machine Intell. 30 (1), 36–51.

Fischer, A., Indermuhle, E., Bunke, H., Viehhauser, G., Stolz, M., 2010. Ground truth creation for handwriting recognition in historical documents. In: DAS '10: Proc. 9th IAPR Internat. Workshop on Document Analysis Systems, pp. 3–10.

Govindaraju, V., Cao, H., Bhardwaj, A., 2009. Handwritten document retrieval strategies. In: AND '09: Proc. Third Workshop on Analytics for Noisy Unstructured Text Data, Barcelona, Spain, pp. 3–7.

Gupta, M.R., Jacobson, N.P., Garcia, E.K., 2007. OCR binarization and image preprocessing for searching historical documents. Pattern Recognition 40 (2), 389–397.

He, J., Do, Q.D.M., Downton, A.C., Kim, J.H., 2005. A comparison of binarization methods for historical archive documents. In: ICDAR '05: Proc. Eighth Internat. Conf. Document Analysis and Recognition, Washington, DC, USA, pp. 538–542.

Heckbert, P.S., Garland, M., 1997. Survey of Polygonal Surface Simplification Algorithms. Tech. Rep., School of Computer Science, Carnegie Mellon University, Pittsburgh, USA.

Hershberger, J., Snoeyink, J., 1992. Speeding Up the Douglas–Peucker Line-simplification Algorithm. Tech. Rep., University of British Columbia, Vancouver, BC, Canada, Canada.

Impedovo, S., Ottaviano, L., Occhinegro, S., 1991. Optical character recognition – A survey. Internat. J. Pattern Recognition Artif. Intell. 5, 1–24.

Kanungo, T., Marton, G.E., Bulbul, O., 1998. Performance evaluation of two arabic OCR products. In: Proc. AIPR Workshop on Advances in Computer Assisted Recognition, SPIE, vol. 3584.

Khorsheed, M.S., 2002. Off-line arabic character recognition – A review. Pattern Anal. Appl. 5, 31–45.

Konidaris, T., Gatos, B., Ntzios, K., Pratikakis, I., Theodoridis, S., Perantonis, S.J., 2007. Keyword-guided word spotting in historical printed documents using synthetic data and user feedback. Internat. J. Document Anal. Recognition 9 (2), 167–177.

Kumar, A., Jawahar, C.V., Manmatha, R., 2007. Efficient search in document image collections. In: Proc. 8th Asian Conf. on Computer Vision, pp. 586–595.

Kumar, J., Abd-Almageed, W., Kang, L., Doermann, D., 2010. Handwritten arabic text line segmentation using affinity propagation. In: DAS '10: Proc. 9th IAPR Internat. Workshop on Document Analysis Systems, pp. 135–142.

Kurniawan, F., Khan, A.R., Mohamad, D., 2009. Contour vs non-contour based word segmentation from handwritten text lines: An experimental analysis. Internat. J. Digital Content Technol. Appl. 3 (2).

Lavrenko, V., Rath, T., Manmatha, R., 2004. Holistic word recognition for handwritten historical documents. In: Proc. Document Image Analysis for Libraries, pp. 278–287.

Leydier, Y., Lebourgeois, F., Emptoz, H., 2007. Text search for medieval manuscript images. Pattern Recognition 40, 3552–3567.

Leydier, Y., Ouji, A., LeBourgeois, F., Emptoz, H., 2009. Towards an om-nilingual word retrieval system for ancient manuscripts. Pattern Recognition 42 (9), 2089–2105.

Likforman-Sulem, L., Zahour, A., Taconet, B., 2007. Text line segmentation of historical documents: A survey. Internat. J. Document Anal. Recognition 9 (2), 123–138.

Llados, J., Pratim-Roy, P., Rodrguez, J., Snchez, G., 2007. Word spotting in archive documents using shape contexts. In: IbPRIA '07: Proc. 3rd Iberian Conf. on Pattern Recognition and Image Analysis, Part II, pp. 290–297.

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

Lowe, D.G., 2004. Distinctive image features from scale-invariant key points. Internat. J. Computer Vision 60 (2), 91–110.

Madhvanath, S., Govindaraju, V., 2001. The role of holistic paradigms in handwritten word recognition. IEEE Trans. Pattern Anal. Machine Intell. 23 (2), 149–164.

Manmatha, R., Han, C., Riseman, E.M., 1996. Word spotting: A new approach to indexing handwriting. In: Proc. Conf. on Computer Vision and Pattern Recognition, pp. 631–637.

Manmatha, R., Srimal, N., 1999. Scale space technique for word segmentation in handwritten manuscripts. In: Proc. Second Internat. Conf. on Scale-Space Theories Computer Vision, pp. 22–33.

Marji, M., Siy, P., 2004. Polygonal representation of digital planar curves through dominant point detection a nonparametric algorithm. Pattern Recognition 37, 2113–2130.

Marti, U.V., Bunke, H., 2001. Text line segmentation and word recognition in a system for general writer independent handwriting recognition. In: Sixth Internat. Conf. on Document Analysis and Recognition – ICDAR 2001, Washington, DC, USA, p. 159.

Meshesha, M., Jawahar, C., 2008. Matching word images for content-based retrieval from printed document images. Internat. J. Document Anal. Recognition 11, 29–38.

Moghaddam, R.F., Cheriet, M., 2010. A multi-scale framework for adaptive binarization of degraded document images. Pattern Recognition 43 (6), 2186–2198.

Otsu, N., 1979. A threshold selection method from gray level histograms. IEEE Trans. Systems Man Cybernet. 9, 62–66.

Ouwayed, N., Belaid, A., 2009. Separation of overlapping and touching lines within handwritten arabic documents. In: CAIP '09: Proc. 13th Internat. Conf. on Computer Analysis of Images and Patterns. Berlin, Heidelberg, pp. 237–244.

Parvez, M.T., Mahmoud, S.A., 2010. Polygonal approximation of digital planar curves through adaptive optimizations. Pattern Recognition Lett. 31 (13), 1997–2005.

Plamondon, R., Srihari, S.N., 2000. On-line and off-line handwriting recognition: A comprehensive survey. IEEE Trans. Pattern Anal. Machine Intell., 63–84.

Rath, T.M., Manmatha, R., 2003a. Features for word spotting in historical manuscripts. In: Proc. 7th Internat. Conf. on Document Analysis and Recognition, pp. 218–223.

Rath, T.M., Manmatha, R., 2003. Word image matching using dynamic time warping. In: Proc. Conf. on Computer Vision and Pattern Recognition, vol. 2, 2003, p. 521.

Rath, T.M., Manmatha, R., 2007. Word spotting for historical documents. Internat. J. Document Anal. Recognition 9, 139–152.

Rodriguez-Serrano, J.A., Perronnin, F., 2009. Handwritten word-spotting using hidden Markov models and universal vocabularies. Pattern Recognition 42 (9), 2106–2116.

Rothfeder, J., Manmatha, R., Rath, T., 2006. Aligning transcripts to automatically segmented handwritten manuscripts. In: Proc. Conf. on Document Analysis Systems, vol. 3872, pp. 84–95.

Rothfeder, J.L., Feng, S., Rath, T.M., 2003. Using corner feature correspondences to rank word images by similarity. In: Computer Vision and Pattern Recognition Workshop, vol. 3, p. 30.

Sankar, P., Jawahar, C., 2006. Enabling search over large collections of telugu document images an automatic annotation based approach. In: 5th Indian Conf. on Computer Vision, Graphics and Image Processing, pp. 837–848.

Saykol, E., Sinop, A.K., Gudukbay, U., Ulusoy, Cetin, E., 2004. Content-based retrieval of historical ottoman documents stored as textual images. IEEE Trans. Image Process. 13 (3).

Srihari, S., Ball, G., 2008. Language independent word spotting in scanned documents. In: ICADL 08: Proc. 11th Internat. Conf. on Asian Digital Libraries, pp. 134–143.

Srihari, S., Srinivasan, H., Babu, P., Bhole, C., 2005. Handwritten arabic word spotting using the cedarabic document analysis system. In: Proc. Symposium on Document Image Understanding Technology (SDIUT-05), College Park, MD, pp. 123–132.

Srihari, S.N., Srinivasan, H., Huang, C., Shetty, S., 2006. Spotting words in latin, devanagari and arabic scripts. Artif. Intell. 16, 2–9.

Stathis, P., Kavallieratou, E., Papamarkos, N., 2008. An evaluation survey of binarization algorithms on historical documents. In: 19th Internat. Conf. on Pattern Recognition (ICPR 2008), Tampa, FL.

Suen, C.Y., Berthod, M., Mori, S., 1980. Automatic recognition of handprinted characters the state of the art. Proc. IEEE 68 (4), 469–487.

Terasawa, K., Nagasaki, T., Kawashima, T., 2006. Automatic keyword extraction from historical document images. In: Proc. Conf. Document Analysis Systems, pp. 413–424.

Yalniz, I.Z., Altingovde, I.Z., Gudukbay, U., Ulusoy, O., 2009a. Integrated segmentation and recognition of connected ottoman script. Opt. Eng. 48 (11), 1–12.

Yalniz, I.Z., Altingovde, I., Gudukbay, U., Ulusoy, O., 2009b. Ottoman archives explorer: A retrieval system for digital ottoman archives. ACM J. Comput. Cultural Heritage 2 (3), 1–20.

Zahour, A., Likforman-Sulem, L., Boussellaa, W., Taconet, B., 2007. Text line segmentation of historical arabic documents. In: Ninth Internat. Conf. on Document Analysis and Recognition – ICDAR 2007, vol. 1, pp. 138–142.

Zhang, B., Srihari, S.N., Huang, C., 2004. Word image retrieval using binary features. In: Proc. SPIE, Document Recognition and Retrieval XI, San Jose, CA, USA, vol. 5296.