Word Spotting for Handwritten Documents using Chamfer Distance and Dynamic Time Warping

Raid Saabni
Computer Science Department
Ben-Gurion University Of the Negev, Israel
Traingle R&D Center, Kafr Qarea, Israel
saabni@cs.bgu.ac.il

Jihad El-sana
Computer Science Department
Ben-Gurion University Of the Negev
Beer Sheva, Israel
el-sana@cs.bgu.ac.il

A large amount of handwritten historical documents are located in libraries around the world. The desire to access, search, and explore these documents paves the way for a new age of knowledge sharing and promotes collaboration and understanding between human societies. Currently, the indexes for these documents are generated manually, which is very tedious and time consuming. Results produced by state of the art techniques, for converting complete images of handwritten documents into textual representations, are not yet sufficient. Therefore, word-spotting methods have been developed to archive and index images of handwritten documents in order to enable efficient searching within documents. In this paper, we present a new matching algorithm to be used in word-spotting tasks for historical Arabic documents. We present a novel algorithm based on the Chamfer Distance to compute the similarity between shapes of word-parts. Matching results are used to cluster images of Arabic word-parts into different classes using the Nearest Neighbor rule. To compute the distance between two word-part images, the algorithm subdivides each image into equal-sized slices (windows). A modified version of the Chamfer Distance, incorporating geometric gradient features and distance transform data, is used as a similarity distance between the different slices. Finally, the *Dynamic Time Warping* (DTW) algorithm is used to measure the distance between two images of word-parts. By using the DTW we enabled our system to cluster similar word-parts, even though they are transformed non-linearly due to the nature of handwriting. We tested our implementation of the presented methods using various documents in different writing styles, taken from *Juma'a Al Majid Center - Dubai*, and obtained encouraging results.

Keywords: Word Spotting, Handwriting Recognition, Dynamic Time Warping, Chamfer Distance

1 Introduction

Recent advances in imaging, storing, and network technology have paved the way for launching several projects designed to scan and digitize historical books and manuscripts. These projects aim to disseminate knowledge and provide access to rare documents and old manuscripts, which are kept in brick-and-mortar libraries around the world. The implications of exposing this fascinating heritage to the public are too obvious to enumerate. These documents are written in various languages and come from different regions; they discuss numerous subjects and topics; and were written over many centuries.

In this work we concentrate on historical Arabic documents, since this collection is very large and has attracted modest amounts of research attention. Between the seventh and fifteenth centuries a huge number of documents were written in Arabic in various subjects, ranging from science and philosophy, to individuals' diaries. More than seven million titles have survived the years and are currently available in museums, libraries, and private collections around the world.

Several projects have been initiated in recent years, aimed to digitize historical Arabic documents – [1,2,3Al-Azhar University, Alexandria library, Qatar heritage library]. These projects demonstrate the importance and the need for developing efficient and accurate algorithms for indexing and searching within document images. Currently, such indexes are built manually, which is a tedious, expensive and very time-consuming task. Therefore, automating this task using word spotting and keyword searching algorithms is highly desirable.

In this paper we introduce a word-spotting algorithm for handwritten documents including historical Arabic manuscripts using a novel approach for matching word-images. We assume the input for the proposed algorithm is a collection of binary images of handwritten text, of reasonable quality. This assumption is not made to boil the problem down to the simple case, but to work in ac-

cordance with fact that there are a large number of Arabic manuscripts that can be converted into the required quality using state of the art binarization techniques. After binarization, this process starts by extracting the Connected Components and text lines. The components in each line are collected and classified into main and secondary subsets, where the main components describe the continuous part of a word/word-part and the secondary components include delayed strokes, such as dots, diacritics, and additional strokes. Our current implementation relies only on clustering the images of the main components. The ordering of the word-parts along a line is used to generate words from the identified word-parts.

A slightly modified version of the Chamfer Distance is used to measure the similarity between two slices of images. Generally, we may consider the Chamfer Distance as a suitable method for matching images of complete word-parts against each other. However, this approach may fail due to the non-linear behavior, which frequently occures in handwriting scripts. In our approach, we use the Chamfer Distance, strengthened by the use of geometric gradient features extracted from the contour polyline. These features are then used to measure similarities between vertical slices subdividing the image of a word-part. This matching measurement, when implemented on these slices, is used as a cost function for a DTW-based process to measure the total similarity between the compared images.

2 Related Work

Spotting Word algorithms in handwritten manuscripts provides us with the ability to search for specific words in a given collection of document images automatically, without converting them into their ASCII equivalences. This is done by clustering similar words, depending on their general shape within documents, into different classes, to generate indexes for efficient searching. Shape Matching algorithms roughly fall into two categories [8]: Pixel-based and Feature-based matching. Pixel-based matching approaches measure the similarity between the two images on the pixel domain using various metrics, such as the Euclidean Distance Map (EDM), XOR difference, or the Sum of Square Differences (SSD) [10]. In Feature-based matching, two images are compared using representative features extracted from the images. Similarity measurements, such as DTW and point correspondence, are defined on the feature domain.

You *et al.* [22] presented a hierarchical Chamfer matching scheme as an extension to traditional approaches of detecting edge points, and managed to detect interesting points dynamically. They created a pyramid through a dynamic thresholding scheme to find the best match for points of interest. The same hierarchical approach was used by Borgefors [1] to match edges by minimizing a generalized distance between them.

Many systems presented in previous work used the DTW technique. Different sets of features were used and gave good results comparing to the competing techniques [8]. Manmatha *et al.* [8] were among the first to use DTW for word-spotting. They examined several matching techniques and showed that DTW, in general, provides better results. Rath and Manmatha [14] preprocessed segmented word images to create sets of one-dimensional features, which were compared using DTW. They also analyzed a range of features suitable for matching words using DTW [13]. A probabilistic classifier was used by Rath *et al.* [12, 11], which was trained using discrete feature vectors that describe different word images.

A method to measure similarity between two word images, based on an algorithm which recovers correspondences of points-of-interest, was presented by Rothfeder *et al.* [15]. Srihari *et al.* [21] presented a system for spotting words in scanned document images for three scripts: Devanagari, Arabic, and Latin. Their system retrieved the candidate words from the documents and ranked them based on global word shape features. Shrihari *et al.* [20] used global word shape features to measure the similarity between the spotted words and a set of prototypes from known writers. Shrihari *et al.* [16] presented a design for a search engine for handwritten documents. They indexed documents using global image features, such as stroke width, slant, word gaps, as well as local features that describe the shapes of characters and words. Image indexing was done automatically using page analysis, page segmentation, line separation, word segmentation and recognition of characters and words. A segmentation-free approach was adopted by Lavrenko *et al.* [7]. They used the upper word and projection profile features to spot word images without segmenting into individual characters. They showed that this approach is feasible even for noisy documents. Another segmentation-free approach for keyword search in historical documents was proposed by Gatos *et al.* [5]. Their system combines image preprocessing, synthetic data creation, word spotting and user feedback technologies. A language independent system for preprocessing and word spotting of historical document images was presented by Moghaddam *et al.* [9], which has no need for line and word segmentation. In this system, spotting is performed using the Euclidean distance measure enhanced by rotation and DTW.

An algorithm for robust machine recognition of keywords embedded in a poorly printed document was presented by Kuo and Agazzi [6]. For each keyword, two statistical models were generated – one represents the actual keyword and the other represents all irrelevant words. They adopted dynamic programming to enable elastic

matching using the two models. Saabni and El-sana [18] presented an algorithm for searching Arabic keywords in handwritten documents. In their approach, they used geometric features taken from the contours of the word-parts to generate feature vectors. DTW uses these real valued feature vectors to measure similarity between word-parts. Different templates of the searched keywords were synthetically generated to be matched against the word-parts within the document image. Chen *et al.* [2] developed a font-independent system, which is based on Hidden Markov Model(HMM) to spot user-specified keywords in a scanned image. The system extracted potential keywords from the image using a morphology-based preprocessor and then used the external shape and internal structure of the words to produce feature vectors. Duong *et al.* [3] presented an approach that extracts regions of interest from gray scale images. The extracted regions are classified as textual or non-textual using geometric and texture features. Farooq *et al.* [4] presented preprocessing techniques for Arabic handwritten documents, to overcome the ineffectiveness of conventional preprocessing for such documents. They described techniques for slant normalization, slope correction, and line and word separation for handwritten Arabic documents.

3 Our Approach

Word spotting algorithms are usually based on clustering similar images, where clusters are used to generate indexes for word/word-part images to be used efficiently in future search tasks. In this work, we extract images of Arabic word-parts from text block images and mutually match them against each other. The distance between these word-parts is used to classify them into various clusters, where each cluster represents an Arabic word-part as shown in Figure 1. A human operator then assigns textual representation to the resulting clusters. Our matching algorithm is based on DTW and a modified Chamfer Distance that includes geometric gradient features. Here we give a detailed description for each phase of the proposed algorithm.

3.1 Line Extraction and Component Labeling

Zhixin *et al.* [19] presented a novel approach based on a generalized Adaptive Local Connectivity Map (ALCM) using a steerable directional filter for extracting text lines from text images. This method manages to extract lines, even when text blocks have multi-skewed lines, as frequently appears in handwritten manuscripts. We have used this method to extract text lines as collections of sequentially ordered components. The resulting images – each representing a word-part – are used for the matching process.

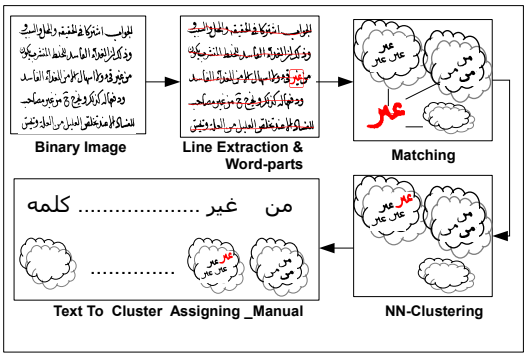

Figure 1. This figure depicts the spotting process starting from top-left with the binary image and ending with bottom-left with the clusters of spotted words.

3.2 Computing the Similarity Distance

The Chamfer matching technique evaluates the similarity distance between a template image, I_t, and a candidate input image, I_i, by overlaying the edge map of I_i on a Distance Transform Map (DTM), I_t, and measuring the fitness in terms of pixel values in the DTM matching edge pixels. This distance is usually computed using Equation 1, which computes the root mean square average of the sum of the values if the DTM (I_t) is covered by pixels of the model edge map of the input image. The Chamfer matching distance is a simple and effective technique to measure distances between edges in the two images. However, it does not take into consideration the local behavior of pixels. In the proposed matching algorithm we modify the Chamfer Distance by integrating the difference in the local behavior (neighborhood) of pixels into the input edge and the overlayed pixels in the template image, (see Figure 2). The idea of the presented approach is to improve the Chamfer Distance to include the difference between the geometric behavior of the compared pixels, in addition to the value of the DTM. Formally, let B_w be a binary image containing the word-part w, and $C_w = \{P_i\}_{i=1}^{l}$ be the contour of the word-part w in B_w, where $X(p_i)$ and $Y(p_i)$ are the x and y coordinates of of the pixel P_i in B_w. We assume, without loss of generality, that contours are extracted consistently in a clockwise direction. For an $\epsilon > 1$ neighborhood of a pixel, $p_i \in C_w$, we define $\alpha(p_i)$ to be the angle between the line $\overline{(p_i, p_i + \epsilon)}$ (along the contour) and the x-axis. For each pixel $p_j \in C_w$, where $i < j < i + \epsilon$, we assign $\alpha(p_j)$ as equal to $\alpha(p_i)$. Let DT_w be the DTM of the image B_w and DTC_w be the DTM of the edge model of B_w (the edge model includes only pixels from the class C_w). To

generate the *Gradient Edge Map* (GEM) we assign to each pixel inside and outside the contour the proper $\alpha(p)$ imitating a dilation process tracking the closest pixel *p* which have been already assigned a value (See Figure 2). Formally, to generate, GEM_w, for the given B_w, which has the same size as B_w, we apply Algorithm 1.

Algorithm 1 Generating the Gradient Edge Map of w

for each pixel $p_i \in C_w$ **do**
  $GEM_w(X(p_i), Y(P_i)) \leftarrow \alpha(p_i)$
end for
$min_{val} \leftarrow 0$
Do {
$m \leftarrow findMinValue\ in\ DT_w\ where\ m\ >\ min_{val}$
for each $DT_w(i,j) \equiv m$ **do**
  $q \leftarrow the\ closest\ pixel\ to\ p_i\ with\ value\ <\ m$
  $GEM_w(i,j) \leftarrow \alpha(q_i)$
end for
$min_{val} \leftarrow m$
} $While\ (GEM_w\ still\ has\ empty\ cells)$

To generate GEM for an input image (w_1), we apply the same algorithm by updating only foreground pixels. To calculate the modified Chamfer Distance between two equal-size images B_{w1} and B_{w2}, we generate DT_{w1}, DT_{w2}, GEM_{w1} and GEM_{w2}. We then overlay B_{w1} over DT_{w2} and sum the values at GEM_{w2} using Equation 1, where k is the number of pixels with foreground values in B_{w2}, and V_{ij} is defined based on Equation 2.

$$\frac{1}{3}\sqrt{\frac{1}{k}\sum_{i=0}^{n}\sum_{j=0}^{m}V_{ij}^2} \tag{1}$$

$$\begin{aligned} V_{ij} =& B_{w2}(i,j)(DT_{w1}(i,j) \\ &+ (GEM_{w1}(i,j) - GEM_{w2}(i,j))^2) \end{aligned} \tag{2}$$

3.3 Matching

Word spotting methods usually rely on a matching algorithm to cluster similar pictorial representation of words. In this paper we use a hybrid scheme which uses the Chamfer Distance and geometric gradient features of pixels to measure the distance between two images. Applying DTW to a series of windows sliding horizontally over the images, assimilates with the inherent non-linear nature of handwriting. We adapted a holistic approach and avoided segmenting words into letters. The search for a given keyword is performed by determining its word-parts in the right order. Our matching algorithm accepts two binary images, w_1 and w_2, representing two word-parts, and returns the similarity distance, $d(w_1, w_2)$, between them. The width of the two images may vary but they are normalized to the same height h. We define δ_w to be the width of the sliding window. To compute the similarity distance d(w_1 ,w_2) between the two word-parts, we apply the following steps:

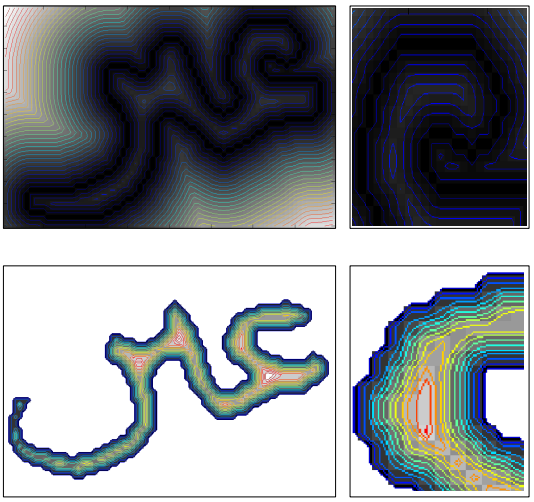

Figure 2. In the first row we can see the Gradient Edge Map for the template word-part image of the word-part Ghayr. In the second row we see the Gradient Edge Map for the same word-part as an input image.

1. Compute the Distance Transform DT_{w1} of the image w_1.
2. Compute the Gradient Edge Map GEM_{w1} and GEM_{w2} of the word-parts w_1 and w_2 respectively.
3. Subdivide w_1 and GEM_{w1} into n windows of width δ_w; i.e, $n = width(w_1)/\delta_w$;
4. Subdivide w_2 and GEM_{w2} into m windows of width δ_w; i.e., $m = width(w_2)/\delta_w$.
5. Create a matrix D ($n \times m$), where the entry $D(i,j)$ is the similarity distance between the two windows, w_1^i and w_2^j (see detail in Section 3.2)
6. Apply DTW to the matrix D to find the path with minimum cost from the upper left entry to the bottom right one; this is the warping path. The value in the bottom-right entry, $D(n,m)$, normalized by the warping path, is the distance between w_1 and w_2.

The distance D(w_1 ,w_2) = (d(w_1, w_2) + d(w_2, w_1)) /2, is the similarity distance between the two images w_1 and w_2.

3.4 Dynamic Time Warping

DTW is an algorithm for measuring similarity between two sequences, which may vary in time or speed. It is suitable for matching sequences with missing information or with non-linear warping, which could be used to classify handwritten words/word-parts. The Chamfer Distance, as described in the previous subsection, has been modified to take local behavior of pixels into consideration. Another weakness of the Chamfer Distance is the inability to take non-linear behavior (which is frequent in handwriting styles) into consideration. In the presented approach, we discard the idea of using the Chamfer Distance on complete images, and use it only on sliding windows with the same width as a cost function for a DTW based method. For $1D$ sequences, DTW runs at polynomial time complexity and is usually computed by dynamic programming techniques using Equation 3.

$$D(i,j) = min\{D(i,j-1), D(i-1,j), D(i,j)\} + cost \quad (3)$$

Here we have converted the $2D$ images to a $1D$ sequence of n windows, with the width δ_w sliding horizontally along the image from left to right. In our approach, we use a non overlapping sliding window in both images, which have the same normalized height h.

Generally in this approach, we compare the width W_{B1} of B_{w1} and W_{B2} of B_{w2} by computing the ratio R = W_{B1}/W_{B2} between the two images. If R is in the range $[0.5, 1.5]$, then we use the presented approach to measure the distance between w_1 and w_2, otherwise, we consider them as different word-parts.

3.5 Clustering Process

There are many methods used in the literature for clustering elements. The *Nearest Neighbor based Clustering* method is one of the simplest known, therefore it was the first option to be used in the clustering process. In our process, the newly encountered (candidate) word-part is compared with the already clustered word-parts and added to the cluster, which includes the closest word-part (Nearest Neighbor). If the distance to any clustered word-part is more than a predefined threshold the candidate word-part creates a new cluster. During the process, whenever two different clusters become closer (pairwise distance) than a predefined threshold, they are merged into one cluster. In the next step a human operator approves the results and assigns the text code to each cluster.

4 Experimental results

We have evaluated our system using 100 pages from various documents, with different writing styles, obtained from *Juma'a Al Majid Center - Dubai*. These pages were selected due to being of reasonable quality, and having been binarized using state of the art techniques. In our experiments, we have concentrated mainly on the quality of the mutual matching results of word-parts, when considering the main parts of words (without the additional strokes). We have classified secondary elements – dots, diactrics, and delayed strokes – based on their size and position. The number of different word-parts in all pages analyzed was $29,654$. Among them we have selected 60 meaningful word-parts, assigned textual representations, and used these for performance evaluation (see Table 1). As can be seen in Figure 3, similar word-parts with different shapes from the same document were successfully clustered into the same group.

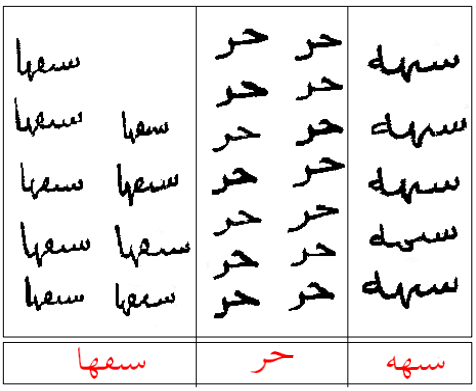

Figure 3. In this figure we present three different resultant clusters from the presented system of three Arabic word-parts. The manually assigned word-parts are in red and the different shapes of the same word-part in each cluster are in black.

The word-part ranking column in ,Table 1, determines the method we have used to classify a word-part to a candidate cluster. In the first row, the value (1) states that the candidate word-parts have been clustered immediately to the closest cluster. The second and third rows ($< k$), state that if the right cluster was one of the k (k is 5 or 10) highest results, it was considered as a successful clustering step.

5 Conclusion

We have presented a matching algorithm to be used for spotting and keyword searching tasks for handwritten

Table 1. The percentage of correct classifications of the 60 word-parts. The precision is computed manually by dividing the number of correctly clustered word-parts by the total number of clustered word-parts (true + false positive).

Word-Part Ranking	Correctly Classified
1	89.4%
< 5	95.8%
< 10	99.3%

Arabic documents. Our experimental results show that the the modified Chamfer Distance, as a hybrid scheme of features, measures the distance between shapes of word-parts efficiently. Using the DTW technique on sliding windows, with the modified Chamfer Distance as a cost function, can overbear the obstacle of the non-linear nature of handwriting. The scope of future work includes working directly on the gray-scale images and incorporating the delayed strokes into the clustering procedure. In historical Arabic documents, disjointed word-parts may touch each other, creating one component. In this work, we do not handle these cases and we believe that determining touching components and separating them into their parts is, vital for various document processing applications. We plan to improve the clustering results by using more advanced methods for clustering. An additional improvement we have considered is to provide better seeds for starting the clustering process. We plan to use an algorithm for synthetic generation[17] of images of a predefined set of the targeted word-parts, and by doing that, enabling automatic clustering of word-parts, with no need for human operator assistance.

References

[1] G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(6):849–865, 1988.

[2] F. Chen, L. Wilcox, and D. Bloomberg. Word spotting in scanned images using hidden markov models. In *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, volume 5, pages 1–4vol.5, 27-30 April 1993.

[3] J. Duong, M. Côte, H. Emptoz, and C. Suen. Extraction of text areas in printed document images. In *DocEng '01: Proceedings of the 2001 ACM Symposium on Document engineering*, pages 157–165, New York, NY, USA, 2001. ACM.

[4] F. Farooq, V. Govindaraju, and M. Perrone. Preprocessing methods for handwritten arabic documents. *Document Analysis and Recognition, International Conference on*, 0:267–271, 2005.

[5] B. Gatos, T. Konidaris, K. Ntzios, I. Pratikakis, and S. Perantonis. A segmentation-free approach for keyword search in historical typewritten documents. In *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, pages 54–58Vol.1, 29 Aug.-1 Sept. 2005.

[6] S. Kuo and O. Agazzi. Keyword spotting in poorly printed documents using pseudo 2-d hidden markov models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(8):842–848, 1994.

[7] V. Lavrenko, T. Rath, and R. Manmatha. Holistic word recognition for handwritten historical documents. In *DIAL '04: Proceedings of the First International Workshop on Document Image Analysis for Libraries (DIAL'04)*, page 278, Washington, DC, USA, 2004. IEEE Computer Society.

[8] R. Manmatha and T. Rath. Indexing handwritten historical documents - recent progress. *the Proc. of the Symposium on Document Image Understanding (SDIUT-03)*, pages 77–85, 2003.

[9] R. Moghaddam, D. Rivest-Hénault, and M. Cheriet. Restoration and segmentation of highly degraded characters using a shape-independent level set approach and multi-level classifiers. In *ICDAR2009, Barcelona, Spain*, pages 828–832, 2009.

[10] T. 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, CIIR Technical Report MM-36., 2002.

[11] T. Rath, V. Lavrenko, and R. Manmatha. Retrieving historical manuscripts using shape. Technical report, CIIR Technical Report., 2003.

[12] T. Rath, V. Lavrenko, and R. Manmatha. A statistical approach to retrieving historical manuscript images. *Technical Report of the Center for Intelligent Information Retrieval, University of Massachusetts*, 2003.

[13] T. Rath and R. Manmatha. Features for word spotting in historical manuscripts. In *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*, pages 218–222vol.1, 3-6 Aug. 2003.

[14] T. Rath and R. Manmatha. Word image matching using dynamic time warping. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–521–II–527vol.2, 18-20 June 2003.

[15] Jamie L. Rothfeder, Shaolei Feng, and Toni M. Rath. Using corner feature correspondences to rank word images by similarity. *In: Proc. of the Workshop on Document Image Analysis and Retrieval (DIAR),Madison, WI, June 21*, 2003.

[16] C. Huang S. Srihari and H. Srinivasan. A search engine for handwritten documents. *Document Recognition and Retrieval XII, San Jose, CA,Society of Photo Instrumentation Engineers (SPIE)*, pages pp. 66–75, January 2005.

[17] R. Saabni and J. El-Sana. Efficient generation of comprehensive database for online arabic script recognition. In *ICDAR '09: Proceedings of the 2009 10th International Conference on Document Analysis and Recognition*, pages 1231–1235, Washington, DC, USA, 2009. IEEE Computer Society.

[18] Raid Saabni and Jihad El-Sana. Keyword searching for arabic handwritten documents. In *The 11'th International Conference on Frontiers in Handwriting recognition (ICFHR2008), Montreal*, pages 716–722, 2008.

[19] Zhixin Shi, Srirangaraj Setlur, and Venu Govindaraju. A steerable directional local profile technique for extraction of handwritten arabic text lines. In *ICDAR*, pages 176–180, 2009.

[20] S. Srihari, H. Srinivasan, P. Babu, and C. Bhole. Handwritten arabic word spotting using the cedarabic document analysis system. *Proc. Symposium on Document Image Understanding (SDIUT 05), College Park, MD*, November 2005.

[21] S. Srihari, H. Srinivasan, C. Huang, and S. Shetty. Spotting words in latin, devanagari and arabic scripts. *Vivek: Indian Journal of Artificial Intelligence,*, 16(3):2–9, 2003.

[22] J. You, E. Pissaloux, W. Zhu, and H. Cohen. Efficient image matching: A hierarchical chamfer matching scheme via distributed system. *Real-Time Imaging*, 1(4):245 – 259, 1995.