Case Study in Hebrew Character Searching

Irina Rabaev, Ofer Biller, Jihad El-Sana, Klara Kedem
Department of Computer Science
Ben-Gurion University
Beer-Sheva, Israel
rabaev,billero,el-sana,klara@cs.bgu.ac.il

Itshak Dinstein
Department of Electrical and Computer Engineering
Ben-Gurion University
Beer-Sheva, Israel
dinstein@ee.bgu.ac.il

***Abstract*—Searching for a letter or a word in historical documents is a practical challenge due to the various degradations present in such documents and the wide variance of handwriting. Searching in historical Hebrew documents is somewhat harder because of high similarities among Hebrew characters. In order to determine the features and their combinations appropriate for recognizing Hebrew script, we study a range of known features using a Dynamic Time Warping algorithm. In addition we describe a novel method for *feature-based* searching, which uses a number of models for the same character. This method is based on our original DTW algorithm that can match fragments of several models of the same character to match a query character. Consequently, we are not limited to any particular model of the character set. Application of this method leads to a significant improvement, even when using a small set of models.**

***Keywords*-Hebrew historical documents; word spotting; character searching; variational method; dynamic time warping;**

I. Introduction

Historical documents provide windows into various stages of the human history and, therefore, attract the interest of scholars from various disciplines as well as ordinary people. These documents, which discuss various subjects ranging from general literature and history to philosophical and natural sciences, are important for reconstructing the social and economic history of various periods and help to study the culture and the development of the various communities.

The advance of digital scanning and storage technologies have dramatically increased the accessibility and availability of historical documents by taking them out of showcases, cabinets, and dark shelters in museums, libraries, and private houses to the general public. These documents are represented as a set of images, where searching for a word or a phrase and indexing in these documents is a practical challenge.

The research of ancient Hebrew documents is of vast historical importance. A considerable large amount of historical Hebrew documents has been discovered in sanctuaries over the last centuries. Among them the most important and fascinating are the Cairo Genizah[1], containing a huge amount of documents written between the 9th and 19th centuries and the Dead Sea Scrolls[2] containing documents apparently written during the 1st and 2nd centuries BC. Due to the Jewish custom of prohibition sacred texts, many documents have been accumulated over time. Furthermore, as means of precaution, in many cases not only religious texts were archived but also more daily documents from various domains. Therefore the research of these documents can reveal many details both in religious and historical aspects. Due to the vast number of documents, the low quality, and the fact that some of the documents are ripped and scattered over the collections, computer analysis and processing is needed.

Historical documents suffer from broken and smeared characters, holes, faded ink and a number of other artifacts. Traditional Optical Character Recognition (OCR) systems fail when applied to historical manuscripts due to the reasons mentioned above. As an alternative to OCR a word spotting technique was proposed [1]. The main idea of word spotting is that the search is performed on image domain without converting the document to textual representation. The goal is to find all images in the document that are similar to the given query image. There are various methods for measuring similarity between images.

In this work we study the applicability of word-spotting approach to character searching in historical manuscripts written in Hebrew script. The Hebrew alphabet is a square block script and contains 22 letters. Five of them have additional forms which are used when the letter is at the end of a word. Hebrew is written from right to left, and does not contain vowels as letters, but set of diacritics, which are placed above and below letters to specify their pronunciation. Most texts appear without the diacritics and the pronunciation is implied by the word and the context. Hebrew is characterized by high similarities among letters (see Figure 1). This property of Hebrew script makes searching even more challenging. Figure 2 presents two sample lines from two different historical documents.

We perform intensive tests using various features on different historical manuscripts and study their performance. We have adopted dynamic time warping (DTW) to measure

[1] http://www.genizah.org/

[2] http://orion.mscc.huji.ac.il/

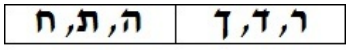


Figure 1. Two groups of very similar Hebrew letters.

the difference between the representative features of the model character and the inspected document image. DTW algorithm has been widely applied for matching word images and shown to provide better results than competing methods [2], [3]. Our objective is to find features and feature combinations that are most appropriate for Hebrew script. In addition we present a novel approach for character search. This approach is based on our original DTW algorithm that can match fragments of several models of the same character to match a query character. Application of this method leads to a significant improvement, even when using a small set of models.

In the rest of this paper we review related work, and then present our approach followed by experimental results. Finally, we draw some conclusions and suggest directions for future work.

II. Related Work

The quality of keyword searching and word spotting approaches highly depends on the matching algorithm. Different word matching algorithms were proposed in the literature and we will briefly overview them in the rest of this section. Image similarities can be measured on spatial domain or on feature domain. Feature-based image matching algorithm compare images using various features that are extracted from the images.

Kolcz *et al.* [4] used a line-oriented search strategy, where each document page was treated as a sequence of lines of text and each line was represented by a sequence of pixel columns. Matching was performed on profile-oriented features using DTW. Rath and Manmatha [2] combined four different features into a single multi-dimensional vector and applied DTW on this feature vector. In addition, a number of matching techniques were compared and it was shown that DTW approach produces better results. Saabni and El-Sana [3] used geometric features extracted from the contours of the word-parts and experimented with two different classifiers - Hidden Markov Model (HMM) and DTW. In [5] they embedded a modified Chamfer Distance based on geometric gradient features into DTW. Rodriguez and Perronnin [6] proposed a feature set inspired by the Scale-Invariant-Feature Transform keypoint descriptor. They compared their feature set with state-of-art ones using DTW and HMM. In both cases they reported a significant improvement compared to state-of-the-art feature sets. However, Pramod Sankar *et al.* [7] found that the gradient features perform worse than the classical profile features while comparing them on historical documents. Lavrenko *et al.* [8] used HMM for word recognition in historical documents. HMM worked on features extracted from the word images. Gatos and Pratikakis [9] used the block-based document image descriptors in their matching algorithm. Ntzios *et al.* [10] classified characters according to the cavities in the skeletonized character and protrusible segments that appear in character skeletons. Konidaris *et al.* [11] obtained an improved performance for word spotting by combining synthetic data and user feedback into the word matching process. Shrihari *et al.* [12] used gradient-based binary features and correlation similarity measure to evaluate the distance between the spotted words and a set of prototypes from known writers.

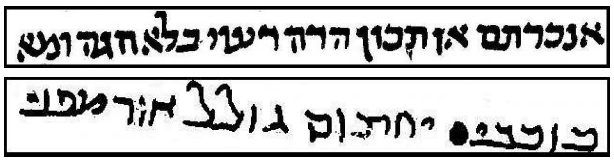


Figure 2. Samples from the two historical documents on which we apply some of our tests. Note the noisy and broken characters.

III. Our Approach

In this paper we study character searching applied to handwritten historical Hebrew documents. We use dynamic time warping algorithm which works on feature vectors extracted from a binary image. The DTW method performs a non-linear sequence alignment, which is suitable for handwritten text, since handwritten text is characterized by the variability in size of the characters. Our goal is to find features and their combinations that best suit the Hebrew binarized historical documents. We experimented with two historical documents from Cairo Genizah. The first document contained one page of about 300 characters with average letter size of 65×81 pixels, the second contained four pages of about 1260 characters with average letter size of 27×28 pixels.

A. Feature Extraction

We concentrate on the following features which are recommended in the literature [13], [2], [4].

Let W be a window of a size $N_c \times N_r$ along a row of text, and let B be a binary image of a character contained in W. Let (r, c) denote the row and column coordinates of the pixels in W, and let $B(r, c)$ be the value of pixel (r, c), where $r = 1, \ldots, N_r$ and $c = 1, \ldots, N_c$. Let the foreground pixels have the value "1", and the background pixels have the value "0". Define the *Upper Boundary (UB)* of the character in B to be the sequence of pixel coordinates *UB(c)* such that $UB(c) = (r_{max}, c)$, where r_{max} is the highest "1" valued pixel in column c. Similarly, the *Lower Boundary (LB)* of the character is the sequence of pixels $LB(c)$, such that $LB(c) = (r_{min}, c)$, where r_{min} is the lowest "1" valued

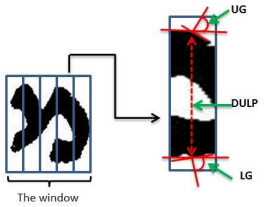


Figure 3. Feature extraction.

pixel in column c. The features used in this study are as follows:

1) *UG(c)* is the gradient angle of the upper boundary at $UB(c)$, and *LG(c)* is the gradient angle of the lower boundary at $LB(c), c = 1, \ldots, N_c$.
2) *VP(c)*, the vertical projection profile, is the sum of the pixel values belonging to pixels in column c. It is the number of "1" valued pixels in column c.
3) *TP(c)*, the transition profile, is the number of pairs of consecutive pixels belonging to column c, where the upper one has the value "1" and the lower one has the value "0".
4) *DULP(c)*, the distance between the upper and lower boundary at column c. $DULP(c) = UB(c) - LB(c)$.

The features are illustrated in Figure 3.

B. Matching

The input to the matching algorithm is a binarized text image segmented into text lines. The query images were selected from the documents to be searched.

The features listed above were used with the dynamic time warping algorithm for letter searching. As Hebrew is written from right to left the DTW algorithm is applied to a window sliding horizontally over the text lines from right to left. Each pixel column is tested as a starting point of a character image. Since the handwritten text is characterized by size variation, for each pixel column we checked a number of windows with width varying from $0.7w$ to $1.3w$ with steps $0.05w$, where w is the width of the query image . From each window we extracted feature vectors (as shown in Figure 3) and applied DTW on these feature vectors. The performance of searching was evaluated by inspecting the *Precision-Recall (PR)* curves. *Precision* is the ratio of the number of correct character images retrieved to the total number retrieved images. *Recall* is the ratio of the number of correct images retrieved to the total number of correct images in the database. Precision can be interpreted as the probability that a retrieved image is correct, while recall can be interpreted as the probability that a correct image is retrieved. The algorithm may retrieve parts of the characters instead of whole characters. Such cases were considered as a correct match if at least 60% of the character was retrieved, otherwise it was considered as an incorrect match.

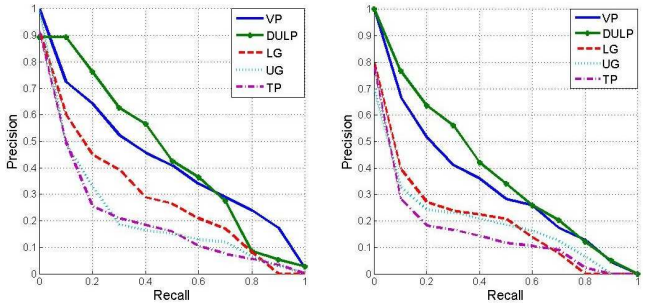


Figure 4. PR-curves corresponding to each individual feature for two different historical documents (see Figure 2).

A PR curve is obtained by changing the threshold used in the matching decision. When the threshold is low, a test character image is retrieved only if the obtained distance between the model and the test character is low. In this case the precision is high and the recall is low. On the other hand, when the threshold is high, test character images with high distances to the model character image may be falsely retrieved, therefore the precision is low and the recall is high.

C. Single feature evaluation

We have experimented with each one of the features separately and Figure 4 presents their comparative performances. We evaluated the feature performances by analyzing the Precision-Recall curves. Among the single features, the DULP and VP features seem to perform best. This is not surprising. Hebrew alphabet is characterized by high similarities of letter shapes and these two features contain most of the information about the form of the letter. In addition, the VP feature is more robust than other features as it is less sensitive to noise, while LG, UG and TP are substantially affected even by one noise pixel above or below the character. We checked LG and UG for a number of different fonts and in most cases LG performed better. This, again, follows from the high similarities among Hebrew letters. For example, for the eleven letters from Figure 5 the UG's are very similar. There is a smaller number of Hebrew letters with similar LGs. The TP feature performs worst compared to the other features. One reason is that the number of transitions in Hebrew alphabet is mostly equal to 1 or 2, and rarely reaches 3. Consequently, change of TP due to noise can cause misidentifying a character.

D. Feature combinations

Each feature has its individual matching performance that relates to different attributes of the characters. It was shown

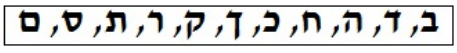
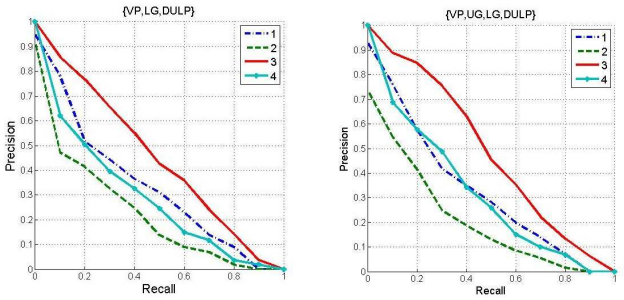
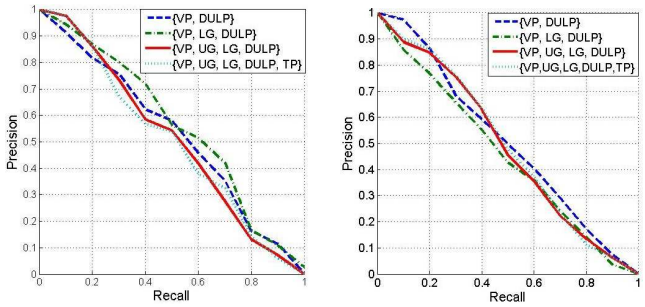


Figure 5. Eleven Hebrew letters with very similar UGs.



(a) (b)

Figure 6. Comparative performance of different cost functions for the combinations {VP, LG, DULP} and {VP, UG, LG, DULP}. (1) Sum of normalized DTW costs of each feature, (2) sum of DTW costs of normalized features, (3) DTW on multi-dimensional vector with Euclidean distance, (4) DTW on multi-dimensional vector with squared Euclidean distance. It can be seen that (3) is the best.



(a) (b)

Figure 7. Comparative performance of combination of features for two different historical documents from Figure 2.

that combining different features yields better results [13], [2], [4], [11]. We have to define how to use the DTW on a feature combination. We compare the performance of four different approaches.

1) Compute the sum of DTW costs of each feature separately, then record minimum sum and the corresponding window size. The DTW costs are normalized to [0, 1].
2) Same as (1), but instead of normalizing the DTW costs, the feature vectors are normalized to [0, 1].
3) Each individual feature is normalized to the range [0, 1], then the feature set per image column is extracted and combined into a single multi-dimensional vector as described in [2]. A single DTW is performed on this multi-dimensional vector. To build the costs matrix for the DTW algorithm we use Euclidean distance. Again, the minimum DTW cost and the corresponding window size are recorded.
4) Same as (3), but instead of Euclidean distance the squared Euclidean distance is used.

We experimented with the cost methods above on a variety of feature combinations. Figure 6 shows an example of the PR curves corresponding to each one of the approaches for two feature sets: {VP, LG, DULP} and {VP, UG, LG, DULP}. For other feature combinations the relations of the PR curves were similar to those shown in Figure 6. Inspecting Figure 6 it is clear that the best cost function is (3).[3]

To study which feature combination best suits Hebrew square letters, we run the DTW with a combination of the two, three, four and five features according to their rank of significance (as revealed in Figure 4). Figure 7 shows PR curves for each one of the combinations applied to the historical documents.

It is noticeable in Figure 7 that the performances of the feature sets are very close, which means that adding feature to the feature set {VP, DULP} does not lead to a significant improvement.

E. The multi-model DTW algorithm

In this paragraph we propose a variational feature-based character search. This variational method uses a number of models per letter and is not restricted to one predetermined model per letter. From each model our algorithm chooses the fragments that best suit the candidate image. All the models were randomly chosen from the searched document. We developed a new DTW algorithm to minimize, at each matrix entry, (i, j), the distance between the feature vector of the image at pixel column j and the feature vectors of all the model images at pixel column i. More precisely, let $F_k = (f_1^k, f_2^k, , f_u^k)$ be the feature vectors of model k, for all $k = 1, \ldots, \#models$, and let $F_c = (f_1^c, f_2^c, , f_v^c)$ be the feature vector of the candidate image, where u, v are the widths of the model characters and the candidate image respectively (all models were normalized to the same width – their average width – while preserving their aspect ratios). The cost matrix C, of size $u \times v$, is calculated as follows:

$$C(i, j) = \min_k dist(f_i^k, f_j^c).$$

Each of the chosen models contributes from its feature vector only the image columns that best fit the tested image in the window. The main advantage of our variational approach is that it captures local information of the candidate image with respect to the given models. The minimal DTW cost indicates that the probability that the candidate image belongs to the same class as the chosen models is high.

Figure 8 shows the PR curves corresponding to four different runs on the same document: with one (blue), two

[3]Notice the significant gap between the performances of DTW that uses Euclidean distance (3) and DTW that uses squared Euclidean distance (4). The latter performs worse. Rath and Manmatha [2] reported that the Euclidean squared distance performed better on their data. We suspect that the difference follows from the properties of Hebrew script. We have experimented with Euclidean and squared Euclidean distances on different fonts and in all cases the Euclidean distance performed better.

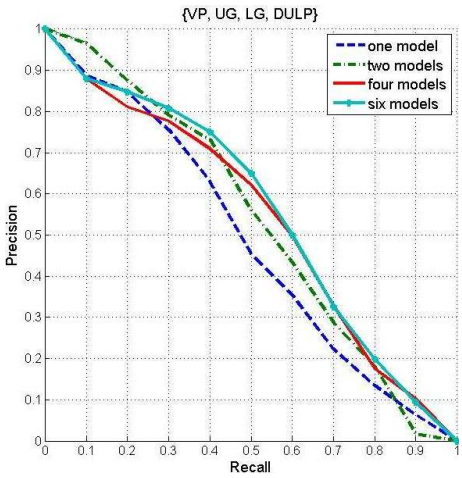


Figure 8. Comparative performances of four different runs: using one, two, four and six different models of the same character.

(green), four (red) and six (turquoise) different models. As can be seen an improvement of about 45% in precision rate is achieved at recall 0.5 by using six models instead of one. On the other hand there isn't a significant improvement between using four and six models. This might indicate that it is enough to use four models.

IV. Conclusions

We tested a number of features and cost functions for the DTW algorithm, trying to determine the best feature combination for search of handwritten characters in Hebrew historical documents. We found that combined DULP and VP features distinguish Hebrew squared characters better than other single features and other feature combinations. In addition we showed the increased benefit of employing a number of models for each character and extending the DTW algorithm to pick parts of the models that best fit the tested image window. This method yielded superior performance compared to using only one model per character. In the future we plan to search for additional features that are appropriate for Hebrew script and to use a set of models that cover much of the variance of a letter in a set of documents. We plan to extend our methods from character searching to word searching in historical documents in Hebrew.

Acknowledgment

This research was supported in part by the Israel Science Foundation grant no. 1266/09, DFG-Trilateral Grant no. 8716, the Lynn and William Frankel Center for Computer Sciences, and the Paul Ivanier Center for Robotics and Production Management at Ben-Gurion University, Israel. We would like to thank Dr. Uri Ehrlich from the Goldstein-Goren department of Jewish thought, Ben-Gurion University of the Negev, for supplying the documents.

References

[1] R. Manmatha and W. B. Croft, "Word spotting: Indexing handwritten archives," in *Intelligent Multi-media Information Retrieval Collection, M. Maybury (ed.)*. AAAI/MIT Press, 1997.

[2] T. Rath and R. Manmatha, "Word image matching using dynamic time warping," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03)*, vol. 2, 2003, pp. II–521–II–527.

[3] R. Saabni and J. El-Sana, "Keyword searching for Arabic handwritten documents," in *The 11'th International Conference on Frontiers in Handwriting recognition (ICFHR2008), Montreal*, 2008, pp. 716–722.

[4] A. Kolcz, J. Alspector, M. Augusteijn, R. Carlson, and G. V. Popescu, "A line-oriented approach to word spotting in handwritten documents," *Pattern Analysis & Applications*, vol. 3, pp. 153–168, 2000.

[5] R. Saabni and J. El-Sana, "Word spotting for handwritten documents using Chamfer distance and dynamic time warping," in *DRR*, San Francisco, USA, 2011.

[6] J. A. Rodriguez and F. Perronnin, "Local gradient histogram features for word spotting in unconstrained handwritten documents," in *The 11th International Conference on Frontiers in Handwriting Recognition*, Concordia University, Montreal, Canada, 2008.

[7] K. Pramod Sankar, C. V. Jawahar, and R. Manmatha, "Nearest neighbor based collection OCR," in *DAS' 10 Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, 2010.

[8] V. Lavrenko, T. Rath, and R. Manmatha, "Holistic word recognition for handwritten historical documents," PARC. Institute of Electrical & Electronics Engineering, 2004, MM, pp. 278–287.

[9] B. Gatos and I. Pratikakis, "Segmentation-free word spotting in historical printed documents document analysis and recognition," in *ICDAR '09. 10th International Conference*, 2009, pp. 271–275.

[10] K. Ntzios, B. Gatos, I. Pratikakis, T. Konidaris, and S. J. Perantonis, "An old greek handwritten OCR system based on an efficient segmentation-free approach," *International Journal on Document Analysis and Recognition*, vol. 9, no. 2-4, pp. 179–192, 2007.

[11] T. Konidaris, B. Gatos, K. Ntzios, I. E. Pratikakis, S. Theodoridis, and S. J. Perantonis, "Keyword-guided word spotting in historical printed documents using synthetic data and user feedback," in *IJDAR*, vol. 9(2-4), 2007, pp. 167–177.

[12] S. Srihari, H. Srinivasan, and C. Bhole, "Handwritten Arabic word spotting using the CEDARABIC document analysis system," in *Proc. Symposium on Document Image Understanding (SDIUT 05), College Park, MD*, 2005, pp. 123–132.

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