2011 International Conference on Document Analysis and Recognition

Fast Key-Word Searching via Embedding and Active-DTW

Raid Saabni
Faculty of Engineering, Tel-Aviv University
Triangle R&D Center, Kafr Qara
Tel-Aviv, Israel
raidsaab@post.tau.ac.il

Alex Bronstein
Faculty of Engineering, Tel-Aviv University
Tel-Aviv, Israel
bron@eng.tau.ac.il

***Abstract*—In this paper we present a novel approach for fast search of handwritten Arabic word-parts within large lexicons. The algorithm runs through three steps to achieve the required results. First it warps multiple appearances of each word-part in the lexicon for embedding into the same euclidean space. The embedding is done based on the warping path produced by the Dynamic Time Warping (DTW) process while calculating the similarity distance. In the next step, all samples of different word-parts are resampled uniformly to the same size. The kd-tree structure is used to store all shapes representing word-parts in the lexicon. Fast approximation of k-nearest neighbors generates a short list of candidates to be presented to the next step. In the third step, the Active-DTW [15] algorithm is used to examine each sample in the short list and give final accurate results. We demonstrate our method on a database of $23,500$ images of word-parts extracted from the IFN/ENIT database [6] and $22,000$ images collected from 93 writers. Our method achieves a speedup of 5 orders of magnitude over the exact method, at the cost of only a 3.8% reduction in accuracy.**

***Keywords*-Word Searching; Handwriting Recognition; Dynamic Time Warping; Embedding; Nearest Neighbor;**

I. Introduction

Methods based on Dynamic Time Warping (DTW) have been proved relatively efficient and effective for matching the shapes of handwritten words in script recognition tasks [13], [3], [11], [4]. Manmatha *et al.* [3], [11] have used DTW with a set of features taken from the upper and lower profile to match shapes of words in handwritten documents. Saabni and El-Sana [13], [14] used DTW with features extracted from contours [13] or sliding windows [14] to compare shapes for keyword searching tasks. In keyword searching and script recognition tasks, shapes have to be matched to sets of multiple appearances of different words/word-parts. To find the best match to a given shape, a similarity criterion has to be calculated to each shape in these large sets. To compare efficiently to multiple appearances of the same word-part, Sridhar *et al.* [15] presented a generative classifier called Active-DTW, that combines Active Shape Models with Elastic Matching. This classifier finds the minimum distance of the test sample to each set of shapes. In this approach, an Active Shape Model(ASM) is generated using different shapes of the same word. Using these models, a closest deformation to the tested word is obtained and the DTW distance to it is calculated. Active-DTW may reduce the time needed for comparing the test shape to one set of shapes representing a candidate (word), but cannot skip the comparison to all classes, which is computationally challenging in large lexicons.

In this paper, we present a novel approach based on the embedding of different shapes represented using their contours into a common representation space which for simplicity is selected to be Euclidean. Embedding is performed by addition or deletion of insignificant points on the contour whilst preserving important feature points. The embedding of contours to a Euclidean space enables the use of state-of-the-art methods for fast approximation of the k-nearest neighbors, such as kd-tree and local sensitivity hashing (LSH). Finding the approximate k-nearest neighbors produces a short list which enables applying expensive matching methods, yet keeps the search time reasonable and constant. In the presented approach, we use the Active-DTW algorithm which gives accurate results with high performance. The high performance of the presented approach enables matching shapes from handwritten documents to a large lexicon of words for indexing, keyword searching, or reading of hand printed documents.

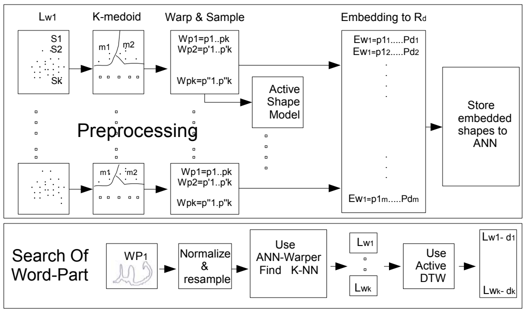

Figure 1. Schematic flow describing different steps of the proposed approach. The upper part shows the preprocessing phase; the lower part depicts the querying phase. In the fourth step of preprocessing, generation of the ASM is performed for a set of appearances after warping and resampling.

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.23

The rest of this paper is organized as follows: in Section II we briefly overview the state-of-the-art in handwriting word matching and searching. In Section III, we describe our approach in details. Experimental results and some directions for future work are presented in Sections IV and V.

II. Related Work

Shape matching algorithms constituting the core of all handwriting recognition systems can be roughly classified as *pixel-* and *feature-based* approaches [3]. Pixel-based matching approaches measure the similarity between the two images in the pixel domain using various metrics, such as the Euclidean distance map (EDM), XOR difference, or the sum of square differences (SSD) [7]. In feature-based matching, images are compared using representative features extracted from them. Similarity measurements such as DTW and point correspondence are typically defined on the feature domain.

The term *word spotting* refers to a family of algorithms in handwritten document analysis that search specific words in a given collection of document images without converting them into their ASCII equivalents. This is usually done by clustering similar words into different classes, depending on their general shape within documents, in order to generate indices for efficient searching.

Many systems for word spotting and searching presented in previous work are based on variants of DTW. Different sets of features were used and gave good results comparing to the competing techniques [3]. Manmatha *et al.* [3] were among the first to introduce DTW for word spotting. They examined several matching techniques and showed that DTW, in general, provides better results. Rath and Manmatha [11] preprocessed segmented word images to create sets of one-dimensional features, which were subsequently compared using DTW. They also analyzed a range of features suitable for matching words using DTW [10]. Rath *et al.* [9], [8] also used a probabilistic classifier trained using discrete feature vectors that describe different word images.

Shrihari *et al.* [16] presented a design of 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.* [2]. They used the upper word and projection profile features to spot word images without segmenting into individual characters. The authors showed the feasibility of the approach even for noisy documents. Another segmentation-free approach for keyword search in historical documents was proposed by Gatos *et al.* [1]. Their system combines image preprocessing, synthetic data creation, word spotting and user feedback techniques. A language-independent system for preprocessing and word spotting of historical document images was presented by Moghaddam *et al.* [4], 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. Saabni and El-Sana [13] 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.

Sridhar *et al.* [15] introduced the Active-DTW classifier. The proposed algorithm finds the minimum distance of the test sample to each of the N classes. The distance to a class, is the DTW distance to the closest deformation that can be obtained from active shape model of that class. The final distance at this point is the DTW-Distance between the test sample and the optimal deformation. The recognized class according to this Active-DTW classifier would be the class with the minimum Active-DTW distance to the test sample, i.e. recognized class. Vandan *et al.* [12] , uses the Active-DTW [15] classifier proposed by Sridhar *et al.*, to propose a supervised adaptation framework for the Active-DTW classifier in an on-line handwriting recognition system which allows recognition to begin with a small number of training samples, and adapts the classifier to the new samples presented to the system during recognition.

III. Our Approach

Let L be a lexicon of n Arabic word-parts, S_{w_i} be the set of all available shapes of different appearances of the word-part $w_i \in L$ and SL be the union of S_{w_i} for each $w_i \in L$. Let w_t be the word-part to be recognized (searched) by matching to all word-parts in the lexicon L. In our approach all shapes are represented as their extracted contour in clockwise direction. Using DTW as a similarity measurement, calculating the minimum distance or minimum average distances will have to compare w_t to all shapes in SL. In this case, the complexity is the product of the size of SL and the time needed for a single matching operation by DTW. Embedding SL into an Euclidean space, will enable fast approximate search with sub-linear time of the size of L and improve the efficiency. In a second phase We use Active-DTW on a short list of word-parts to determine the final results.

Three stages are applied in the fast search for a given shape of a word-part in a large lexicon. In the first stage, all shapes in SL are embedded into an Euclidean space. This is done by a warping process guided by the Warping Path (WP) generated by the calculation of the DTW distance to the relevant medoid (Section III-A). The warping process is applied separately on each set S_{w_i}, resulting in sequences of

2D points with the same size. Different warped sets will have different sizes, therefore a re-sampling process is performed to embed all the warped samples to the same Euclidean space, producing the set ($L_{ws} \subset R^d$), Lexicon of Warped Shapes. To enable fast k-neighbors search, we use the ANN library [5] based on kd-tree to store all shapes from L_{ws}. In the third step, when a query is presented to the system, its contour is first resampled to the same Euclidean space R^d. Using the generated ANN [5], we find the closest k-neighbors coming from different clusters. To determine the final searching results, we use each one of the k word-parts in the short list, to perform a second matching phase based on Active-DTW. Here we give a detailed description for each stage of the proposed algorithm.

A. *k-Medoids and warping shapes of word-parts*

In the presented approach, the process aims to embed shapes into the Euclidean space by removing or adding less significant points from the contour in order to keep the most significant data points. Embedding to the Euclidean space, enables using the state of the art methods for fast approximations of the k-nearest neighbor algorithms such as kd-tree, Locality Sensitivity Hashing (LSH) and others.

The main idea in this process is to warp all samples within the same set which includes multiple appearances of the same word-part to the medoid sample.Due to the large covariance between different shapes of the same Arabic word-part, the warping process will cause much damage when used to warp contours of shapes distant from the medoid. Therefore, the process starts with finding the k-medoids of each set S_{w_i}. Medoids, in this case, are the most central object within a given set with respect to the DTW distance used as the similarity metric. The range of k is predefined based on previous knowledge of the processed word-part, and the exact k is fixed using the average distance within each cluster and between the clusters. As a result, the set S_{w_i} is divided into k subsets (clusters) $\{SW_{m_j}\}_{j=1}^{k}$. The cluster SW_{m_j} is the subset around the j-th medoid m_j. To warp a given shape $s \in SW_{m_j}$, we compute the DTW distance of s to the medoid m_j , using the generated warping path, we warp s to m_j. This results in a sequence of 2D points with the same size as the medoid. In the last step, the set of all warped shapes from the different SW_{m_j} are uniformly resampled and represented as vectors in a common Euclidean space.

Formally, let $C_s = \{p_i\}_{i=1}^{l}$ be the contour of a shape $s \in S_{w_i}$, where $X(p_i)$ and $Y(p_i)$ are the x and y coordinates of the pixel p_i. We assume, without loss of generality, that contours are extracted consistently in a clockwise direction. Let $C_m = \{q_i\}_{i=1}^{l_m}$ be the contour of medoid shape closest to s. The warping process starts with calculating the DTW distance of C_s to C_m using the following formula

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

where $cost$ is the euclidean distance between the points p_i and q_i

We use the warping path(W) generated by calculating the DTW distance of C_s to C_m in order to generate the warped sequence (W_{Seq}) of C_s, using the following formula:

$$W_{Seq} = \left\{ \begin{array}{ll} q(i-1) & W(i,j) = D \\ \alpha.q(i-1) + (1-\alpha).p(i-1) & W(i,j) = I \\ \beta.q(i-1) + (1-\beta).p(i-1) & W(i,j) = S \end{array} \right\}$$

$W(i,j)$ contains the operations: –D for Delete, S for Substitute and I for Insertion– done in the warping process when comparing point i from C_s to point j in C_m.

The resulting warped contour of C_s has exactly l_m points as the medoid sample C_m it was warped to. The deleted/added points are the insignificant ones considering the DTW metric since their deletion costs the minimum distance between the two samples. We repeat this process with each $C_s \in S_{w_i}$ and it's closest medoid. All warped shapes in each S_{w_i} are resampled to the same length as the maximal length of all medoids of this set generating the set WS_{w_i}.

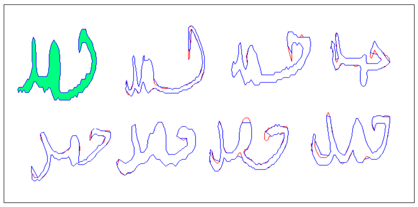

Figure 2. The left top shape (green) is the medoid sample for Arabic word-part. Other shapes are appearances of the same word-part. Each sample shows changes done to the contour by warping it to the medoid. The contour after warping is plotted in blue color above the original in the color.

B. *k-Nearest neighbor approximation*

Computing exact nearest neighbors in high dimensions is a very difficult task. Few methods seem to be significantly better than a brute-force computation of all distances. However, it has been shown that by computing nearest neighbors approximately, it is possible to achieve significantly faster running times with a relatively small actual errors. Searching words in each WS_{w_i} using the Euclidean distance will cost in time that is linear in the size of the lexicon. In the presented approach, we use the Matlab wrapper for the ANN library [5] – a C++ library with routines for both exact and approximate nearest neighbor searching. The tool aims at solving the problem of searching k nearest neighbors (k-NN) in a large set of multi-dimensional points. It first builds a data structure based on the set of reference points, and then searches neighbors for each query point using this structure. The ANN library implements two types of data structures for searching, the kd-trees and box decomposition trees.

Formally, let $C_w = \{P_i\}_{i=1}^{l}$ be the contour of a word-part $s_w \in L_{ws}$. We define $\alpha(p_i)$ to be the angle between the segment $\overline{p_{i-1}p_i}$ and X-Axis to quantify the relation between adjacent points. For a $\delta > 1$ neighborhood of a pixel $p_i \in C_w$, we define $\beta(p_i)$ to be the angle between the line $\overline{(p_i, p_i + \delta)}$ (along the contour) and the x-axis. For each pixel $p_j \in C_w$, where $i < j < i + \epsilon$, we assign $\beta(p_j)$ as equal to $\beta(p_i)$. The two angles are interpolated linearly using Equation (2), where ω is a normalized positive weight that controls the blending of the two angles and δ determines the width of the neighborhood. In our case we have set δ to be 3 and ω to be 0.5. The maximal dimension d we have warped and resampled all contours to is 681.

$$f(p_j) = (1 - \omega)\alpha(p_j) + \omega\beta(p_j) \tag{2}$$

Using Equation 2, we generate a feature vector $F(s_i)$ for each $s_i \in L_{ws}$, where the coordinate j of the vector F is $f(p_j)$. Using the ANN Warper we generate a kd-tree based structure, to store all feature vector $F(s_i)$ for each contour $s_i \in L_{ws}$.

C. Fast search with k-NN and Active-DTW

Let B_w be a binary image containing the word-part w to be searched. Let $C_w = \{P_i\}_{i=1}^{l}$, be the contour of the main component of w. We assume, without loss of generality, that contours are extracted consistently in a clockwise direction as done with the pre processed shapes. Using the same δ to define $\alpha(p_i)$ and $\beta(p_i)$ and ω to generate the feature vector $F(C_w)$.

Our matching algorithm accepts the word-part w and returns the similarity distance of it to the closest sample within a given lexicon. In a preprocessing step the image B_w is normalized to the same height as the images in the lexicon. Then we extract the contour, simplify and normalize it the same way it was done in the preprocessing step of the set L_{ws}. As a pruning step we start searching the k-nearest neighbor of the test sample w approximately, using the ANN library and $F(C_w)$. The result is a ranked list of k' sets of k different word-parts. In the next step we use Active-DTW on each set to give the final recognition result.

D. Matching with Active-DTW

The objective of the Active-DTW Classifier [15] is to find the optimal deformation D_{opt}, that can be obtained from a set of shapes using Active Shape Models. The Active Shape Models uses PCA to capture the principal variations of a given set of samples from the mean prototype for that class. This principal variation data is used to generate a new sample by applying them to the prototype. The distance is computed as the minimum distance of a test sample to the closest deformation that can be obtained from that class.

We are given a sample to be tested, denoted w_t, and a set C_{w_i} of warped contours with the length n representing the word-part $w_i \in L$. To find the Active-DTW distance of w_t to the set C_{w_i}, we seek the optimal deformation $D_{opt} \in D$ which minimizes the Equation (3), where D is the set of all possible deformations.

$$Dist(w_t, C_{w_i}) = \min_{C_d \in D} \|(w_t - C_d)\| \tag{3}$$

To compute this distance, the process starts with producing the covariance matrix of the set C_{w_i}. Using PCA, we calculate the set of Eigenvectors V, the set of eigenvalues λ, and μ, the mean vector of the set C_{w_i}. Let C_d be any valid deformation vector for the set C_{w_i} that can be generated by some parameter vector β, and let β_{opt} be the parameter that generates the optimal deformation, then

$$Dist(W_t, C_{w_i}) = \min_{C_d \in D} \|(W_t - (\mu + \beta_{opt}V')\| \tag{4}$$

where

$$\beta_{opt} = \underset{-3\sqrt{\lambda} \leq \beta \leq 3\sqrt{\lambda}}{argmin} \|(W_t - (\mu + \beta)V')\| \tag{5}$$

Following [15] the minimization of

$$\beta_{opt} = \underset{\beta}{argmin}(\|\beta\|^2 - 2\beta V(W_t - \mu)' + \|W_t - \mu\|^2) \tag{6}$$

subject to

$$-3\sqrt{\lambda} \leq \beta \leq 3\sqrt{\lambda} \tag{7}$$

can be carried out using constrained quadratic optimization problem, the best fitting deformation will be found by

$$C_d^{opt} = \mu + \beta_{opt}V' \tag{8}$$

When C_d^{opt} is found, the Active-DTW distance is be computed as the DTW distance of w_t and C_d^{opt},

$$ADTWD = DTW(W_t, C_d^{opt}), \tag{9}$$

and the recognition result of the word-part w_t will be the label of the set C_{w_i}, $i \leq k$ with the smallest Active-DTW distance to w_t.

IV. Experimental results

To evaluate the proposed approach we have used the IFN/ENIT [6] off-line Arabic database, which is frequently used to train and evaluate Arabic handwriting recognition systems. The database has been slightly modified to work with the main part of word-parts as connected components, i.e, split components for single word-parts have been rejoined to single one. Touching components have been split to the word-parts they represent. The database has been reorganized as a list of sets each containing multiple shape of a word-part. The manually modified version of the $IFN/ENIT$ database includes 836 word-parts with $22,500$ different shapes. Additionally, a set of $22,000$ images of 500 word-parts have been collected using 53 students. 10% percent of each set have been taken out of the lexicon to be used as testing samples. To compare results, we have used the basic DTW algorithm in three different settings: 1)DTW: DTW was used directly on the samples without preprocessing, 2)$ADTW$: Active-DTW was used after the warping process without using ANN, and 3)$ANN + ADTW$: We have used ANN followed by Active-DTW on the top k ranked list.

Factor	DTW	$ADTW$	$ANN + ADTW$
Precision	89.4%	87.2%	85.6%
Time (msec)	735,280	453,584	6.47

Table I
COMPARISON OF THE THREE DIFFERENT APPROACHES. OUR APPROXIMATE METHOD ACHIEVES A 5-TIME DECREASE IN THE SEARCH TIME AT THE EXPENSE OF AN INSIGNIFICANT DROP IN PRECISION.

V. CONCLUSION

We have presented a novel approach for fast searching of Arabic word-parts via embedding and Active-DTW. The embedding is done using a novel technique for warping contours to medoids and resample to the required size. Using approximated methods for K-NN we enable our method to generate a short list of candidates. Our experimental results show that searching handwritten word-parts within large lexicon can be done $100,000$ times faster than comparing to all samples, without real decrease in accuracy. The scope of future work includes working with other approximation techniques for K-NN such as Locality Sensitive Hashing(LSH). We also consider embedding using warping on other features such as shape context and extending the work to the case of words by concatenating separated word-parts to one continuous contour or using other embedding techniques directly on shapes of words.

VI. ACKNOWLEDGMENTS

We would like to thank Prof' D. Ron from Tel-Aviv University for her insightful comments which led to several improvements in the presentation of this paper. This work has been supported in part by the 'Vatat and Pikovsky Valazzi Post Doc scholarship' and the DFG-Trilateral Grant No.8716.

REFERENCES

[1] 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*, volume 1, pages 54–58, 29 Aug.-1 Sept. 2005.

[2] 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.

[3] 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.

[4] R. F. Moghaddam and M. Cheriet. Application of multi-level classifiers and clustering for automatic word spotting in historical document images. In *ICDAR*, pages 511–515, 2009.

[5] D. M. Mount and S. Arya. *ANN: A Library for Approximate Nearest Neighbor Searching*. University of Maryland, Jan 2010. http://www.cs.umd.edu/ mount/ANN/.

[6] M. Pechwitz, S. S. Maddouri, V. Mrgner, N. Ellouze, and H. Amiri. Ifn/enit - database of handwritten arabic words. In *In Proc. of CIFED 2002*, pages 129–136, 2002.

[7] 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.

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

[9] 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.

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

[11] 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 521–527, 18-20 June 2003.

[12] V. Roy, S. Madhvanath, A. Ragunath, and R. Sharma. A framework for adaptation of the active-dtw classifier for online handwritten character recognition. In *10th International Conference on Document Analysis and Recognition*, 2009.

[13] 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*, pages 716–722, 2008.

[14] R. Saabni and J. El-Sana. Word spotting for handwritten documents using chamfer distance and dynamic time warping. In *Document Recognition and Retrieval XVIII*, 2011.

[15] M. Sridha, D. Mandalapu, and M. Patel. Active-dtw : A generative classiffier that combines elastic matching with active shape modeling for online handwritten character recognition. In *International Conference on Frontiers in Handwriting Recognition*, 1999.

[16] S. Srihari, C. Huang, 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.