2016 15th International Conference on Frontiers in Handwriting Recognition

Word Spotting using Radial Descriptor Graph

Majeed Kassis
Department of Computer Science
Ben-Gurion University of the Negev, Israel
majeek@cs.bgu.ac.il

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

***Abstract*—In this paper we present, the Radial Descriptor Graph, a novel approach to compare pictorial representation of handwritten text, which is based on the radial descriptor. To build a radial descriptor graph, we compute the radial descriptor and generate feature points. These points are the nodes of the graph, and each adjacent points are connected to its adjacent node to form a planar graph. Then we iteratively reduce the edges of the graph, by merging adjacent nodes, to form a multilevel hierarchical representation of the graph. To compare two pictorial representations, we measure the distance between their correspondence planar graphs, after calculating the dominant signal for each node. The graph matching is based on optimizing the function that takes into account the distance between the feature points and the structure of the graphs. The distance between two radial descriptors is computed by measuring the difference between their corresponding dominant signals. We have tested our approach on three different datasets and obtained encouraging results.**

Keywords*—*local feature, radial descriptor, graph, learning-free

I. Introduction

Word-spotting is a task of locating and retrieving a particular keyword within a document image collection without explicitly transcribing the whole collection. The result of a word-spotting query is a ranked list of word images that are similar to a query image. The query can be an actual pictorial representation from the collection or a synthetically generated representation from a given text. In document image analysis word-spotting appears under two distinct groups, where the fundamental difference concerns the search space, which could be either a set of segmented word images; i.e., segmentation-based approaches, or the complete document image; i.e., segmentation-free approaches. In this work, we address the word spotting problem with a segmentation-based approach. Another possible distinction is learning based, and example based. Learning based methods involves training phase, which uses annotated data, while example based methods do not require a learning step.

Learning based methods demonstrate better performance in comparison to example based methods in general. However, these methods depend on the amount and the quality of the annotated dataset, which are used in the training phase. The annotated datasets are not always available to perform the training phase which is often performed off-line. Word-spotting approaches were developed to overcome this limitation by measuring the distance between two pictorial representations of a text.

Recently, we presented the *Radial Descriptor* [1], and studied its application for spotting keywords in Arabic historical documents using the bag-of-features model. The radial descriptor describes the neighborhood of a feature point in a compact manner. We detect feature points on the gray scale image and generate a feature dictionary, which is used to compute the occurrence probability of each feature in the dictionary, on the processed word. To compute the distance between two pictorial representations we compared their corresponding occurrence probability histograms using a distance metric. The the bag-of-features model captures the occurrence probability of features in an image, but has limited ability to represent the spatial relations among the various feature points.

In this paper, we present a learning-free approach, which is based on the radial descriptor, and encodes spatial relations among feature points in a graph. The nodes of the graph are the feature points, and the edges link adjacent feature points. We refer to this structure as the *Radial Descriptor Graph*.

To build a radial descriptor graph, we compute the radial descriptor and generate feature points. These points are the nodes of the graph, and each adjacent points are connected by an edge to form a planar graph. Then we iteratively reduce the edges of the graph, by merging adjacent nodes, to form a multilevel hierarchical representation of the graph.

To compare two pictorial representations, we measure the distance between their corresponding multilevel planar graphs. The graph matching algorithm is based on optimizing a function that takes into account the distance between the feature points and the structure of the graphs. The distance between two radial descriptors is computed by measuring the difference between their corresponding signals.

In the rest of the paper, we first review related work and subsequently present our approach in detail, followed by experimental results. Finally, we draw conclusions and suggest directions for future work.

II. Related Work

Keyword spotting aims to detect a word in an image and was initially proposed in [2], for printed and handwritten text, respectively. The core of any word spotting procedure is a word-matching algorithm, which measures the distance between pictorial representations of words. Word-matching algorithms roughly fall into two categories: pixel-based and feature-based. Pixel-based matching approaches measure the similarity between the two images on the pixel domain using various metrics, such as Euclidean Distance Map, XOR difference, Scott and Longuet-Higgins distance, Hausdorff distance, or the Sum of Square Differences [3]–[5]. Feature-based matching approaches extract features from the images to be

2167-6445/16 $31.00 © 2016 IEEE
DOI 10.1109/ICFHR.2016.16

compared and measure the similarity on the feature space [6]–[9].

Recent document processing algorithms extract interest points directly from gray scale images [10] and utilize these points for various applications, such as word spotting [11], [12] and writer identification [13]. Most of these algorithms impose a grid or define patches to control the distribution of feature points [11], [14]. Defining the size of this grid and the number of sample points is done in an ad-hoc manner. These algorithms demonstrate improvement over binary-prerequisite-based algorithms for gray scale images. Other works are based on bag-of-visual-words model, such as [10], [15]–[17]. The performance of these algorithms deteriorates as the degradation level increases [18]. The idea of using these points to compare similarity among components is based on the hidden assumption that these points faithfully represent the processed text components.

III. Radial Descriptor Graph

To build the *Radial Descriptor Graph* for an input image, I, we compute the radial descriptor [1] features for I and select the feature points with highest variance. We generate the initial graph using these feature points, and iteratively merge operation to reduce the size of the graph and remove redundant features. Next we discuss, in detail, the steps build the Radial Descriptor Graph, ways to compare such graphs, and its application for word spotting.

A. Features Selection

The radial descriptor aims to describe the neighborhood of a given pixel on an image. We define the r-neighborhood of the pixel $p_{x,y}$, denoted $N(p_{x,y}, r)$, as the set of pixels within the circle of radius r centered at $p_{x,y}$. We refer to the pixels on the circle as the boundary pixels and they are denoted $B(p_{x,y}, r)$. We define the average value of the neighborhood as the average of the internal pixels, as formulated in Equation 1, where $In(p_{x,y}, r) = N(p_{x,y}, r) - B(p_{x,y}, r)$

$$Avg(p_{x,y}) = \sum_{q \in In(p_{x,y},r)} \frac{I[q]}{|Internal|} \tag{1}$$

The radial descriptor of the pixel $p_{x,y}$ at radius r is defined as the difference between the intensity of the boundary pixels and the average intensity of the neighborhood, as described by Equation 2. The radial descriptor, $\mathcal{R}(p_{x,y}, r)$, is a vector, whose order is derived from the order of the boundary pixels.

$$\mathcal{R}(p_{x,y}, r) = \{I[q] - Avg(p_{x,y}, r) | q \in B(p_{x,y}, r)\} \tag{2}$$

The radial descriptor function, $\mathcal{R}$, is a function of the radial angle and the intensity difference, i.e $\mathcal{R} : \mathbf{\Phi} \rightarrow \mathbf{R}$. $\mathcal{R}$ has several interesting properties, it is periodic (period = 2π) and the rotation around the pixel $P_{x,y}$ corresponds to translation along the ϕ axis. Thus, one could apply Fourier transform to obtain a rotation invariant descriptor.

Since one radial descriptor level cannot describe the neighborhood of a pixel, we will require multiple levels for each point. Using Gaussian scale space provides an elegant solution for this issue.

The Gaussian scale space generates multiple representations of an input image at various resolutions. We apply this technique to generate a pyramid representation of the input image, and compute the radial descriptors of the pixels using the same radius for all the representation levels. In this scheme, the radii range and the smooth levels are determined by the generated pyramid.

$$variance(\mathcal{R}) = \int_0^{2\pi} |\mathcal{R}(\phi, R)| d\phi \tag{3}$$

To quantify the importance of a feature point using the radial descriptor, we define the *variance* of the radial function as the sum of the area between the ϕ axis and the function $\mathcal{R}$, as seen in Equation 3. The variance encodes the topography of the neighborhood of a pixel, and its value is proportional to the intensity changes within the neighborhood – the variance is zero for flat neighborhoods and increases according to the changes in the intensity difference. It is important to note that the variance of $\mathcal{R}$ is not rich enough to distinguish between two different radial descriptors, which necessitates measuring the distance between the descriptors. The discretized radial descriptor is a vector of values and one could use any distance metric to measure the similarity of two radial descriptors. The construction of the features is done bottom-up along the pyramid. At each level the radial features are computed for each pixel, and the average variance is calculated.

B. Radial Descriptor Graph

Given a set of radial descriptor feature points of an image, we define the radial descriptor graph $G(V, E)$ as an undirected planar graph, where V is the set of feature points, and two adjacent points $v_i \in V$ and $v_j \in V$ connected by an edge $e_{ij} \in E$. We apply Delaunay triangulation to construct the initial graph and iteratively reduce the edges of the graph, by merging adjacent nodes, to form a multilevel hierarchical representation of the graph. Since the feature points sit on the image space, which is planar, the Delaunay triangulation guarantees the generation of a planar graph [19]. An example of the graph can be seen in figure 1

1) Merge Operation: The reduction of the number of the features is done by iteratively applying the merge operation on the graph $G(V, E)$. The merge operation on node $v_i \in V$ with descriptor $\mathcal{R}(v_i, r)$, is performed as follows: we compute the descriptor $\mathcal{R}(v_i, r+\delta)$, where δ is the distance between the node v_i and the closest adjacent node v_a. We update the descriptor of v_i, remove $v_a \in V$ from G, and connect the adjacent nodes of v_a to the updated v_i. Note that we don't need to connect the nodes which are already connected to v_i. We refer to the node v_a as the child of node v_i. The merge operation is illustrated in figure 2

2) Graph Hierarchy: To generate the graph hierarchy, we apply the merge operation iteratively on the graph. Let us define the weight, $\daleth(v_i)$, of a node, v_i, as the difference between the descriptor $\mathcal{R}(v_i, r)$, and $\mathcal{R}(v_i, r+\delta)$, where δ is the minimal Euclidean distance between v_i and v_a. In each iteration we apply the merge operation on the node with the lowest weight.

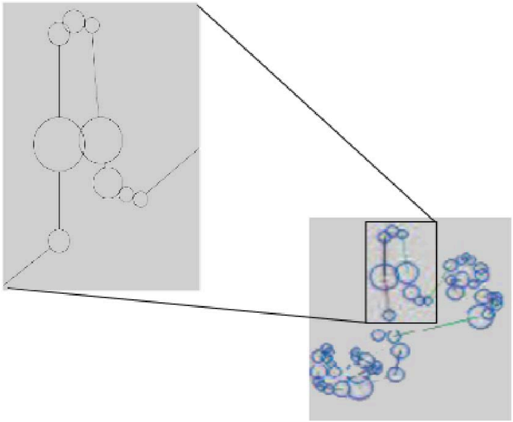

Fig. 1. Illustration of a sub-word graph and its edges

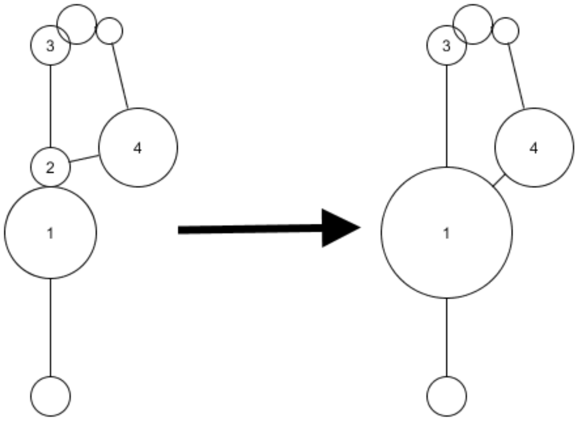

Fig. 2. The merge operation: (a) segment of the original graph, (b) the segment of the graph after merging node 2 into node 1. Node 1 removed, the size of node 1 increases, and its adjacent connection is updated accordingly.

To generate the graph $G_{i+1}(V_{i+1}, E_{i+1})$, at level L_{i+1}, from $G_i(V_i, E_i)$ at level L_i, we determine $|V_{i+1}|$ and apply the merge operation $k = |V_i| - V_{i+1}|$ times. The nodes in level V_{i+1} will be connected to its descendants in the level L_i. The decedents list $\{u_{i0}, u_{i1}, .., u_{ik}\} \subset L_i$ of node $v_i \in L_{i+1}$ are denoted the children of v_i.

At each level the size of level L_{i+1} is half the size of level L_i, and the number of the levels automatically determined as a function of the size of initial graph. An example of two layers in the hierarchy can be seen in figure 3.

3) Distance Metric: We define $C(v_i, (v_0, v_1))$ as the undirected chain graph generated from v_i and its two adjacent nodes v_0 and v_1, ordered either as v_0, v_i, v_1 or as v_1, v_i, v_0. The distance between two given chains $C(v_i, (v_0, v_1))$, and $C(u_j, (u_0, u_1))$ in a radial graph is formulated in equation 4.

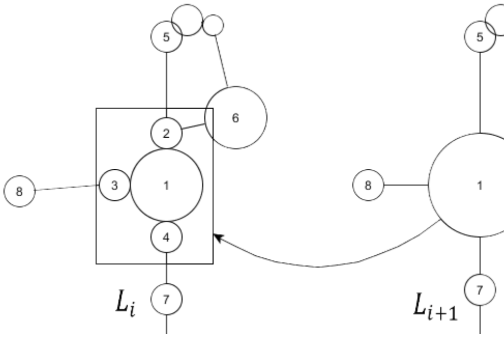

Fig. 3. On left, partial graph of layer L_i. Merge operations applied on nodes 2, 3, and 4, and are merged into node 1. Right, layer L_{i+1} after the merge operations.

$$d(C(v_i, (v_0, v_1)), C(u_j, (u_0, u_1)) = \frac{d(\mathcal{R}(v_i, r_i), \mathcal{R}(u_j, r_j)}{2} + \frac{d(\mathcal{R}(v_0, r_{i,0}), \mathcal{R}(u_0, r_{j,0})}{4} + \frac{d(\mathcal{R}(v_1, r_{i,1}), \mathcal{R}(u_1, r_{j,1})}{4} \quad (4)$$

We define the distance between a node $v \in G_i$, and a node $u \in G_j$ as the maximum distance between all possible corresponding chains as formulated in equation 5

$$d(u, v) = \max_{i,j,k,l} d(C(v, (v_i, v_j)), C(u, u_k, u_l)) \quad (5)$$

We define the weight of an assignment of nodes between two radial descriptor graphs, G_i, and G_j, as the sum of the distance between the corresponding nodes. The distance, $d(G_i, G_j)$, between two radial descriptor graphs, G_i, and G_j, is defined as the weight of the assignment with minimum distance, which is computed using the Hungarian algorithm.

Given two hierarchical graphs H_1, and H_2, with the same initial size, and the same number of levels. To measure the distance between the two hierarchies, we begin with computing the distance between the two graph at the highest level, and propagate the minimum assignment to the lower levels. The propagation of the assignment is done by refining the assignment of two nodes v_i, u_j, in level L_k to the assignment between the children of v_i, and the children of u_j in L_{k-1}. The assignment of the refined nodes is computed locally; i.e., the assignment with the minimum distance of the children of v_i to children of u_j.

To compute the distance between two pictorial representations w_a, and w_b of two words, we build the hierarchical representation of the radial descriptor graphs, H_a and H_b, respectively, and measure the distance between H_a, and H_b. Two images represent the same text if the distance between their corresponding hierarchies is below a predefined threshold.

IV. Experimental Results

We have implemented this framework in C++ using OpenCV [20], and evaluate its performance using various datasets collected from Harvard's Open Collections Program, Islamic Heritage project [21], as well as on datasets collected

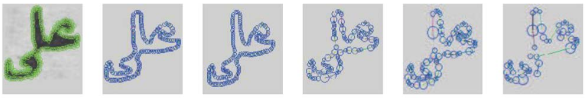

Fig. 4. Graph merging progress for a given sub-words image.

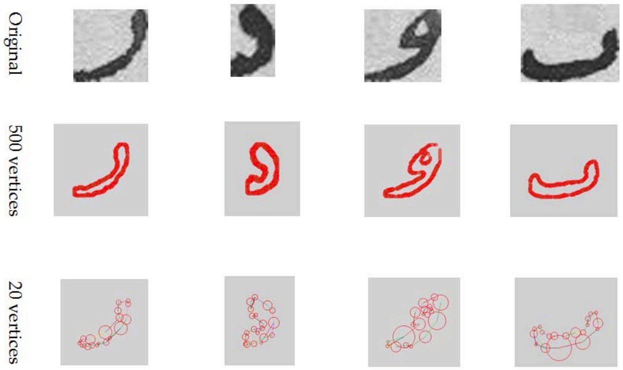

Fig. 5. Examples of initial graph and merged graphs for different sub-words.

from our Arabic historical book collection. We have manually annotated 680 pages from 5 different books, then generated a dataset of sub-words from the annotated data. Fully annotated raw data can be found at [22] website for research use.

The database consists of books written by various writers, in the years 1088-1451. The books where scanned using a very high quality camera, namely Hasselblad H5D-60 Medium Format Digital SLR Camera from 1m distance. They are stored in an uncompressed TIFF format, where each image is roughly of size 6000x4000 pixels. Each image is roughly 100mb of size, so due to size limitations the released dataset contains images of reduced size. The datasets are st quite large, totaling 5Gb.

The first experiment was done on a dataset of 660 sub-words taken from Islamic Heritage project documents, written by one writer, 22 different sub-words, with 30 instances each. The initial graph consisted of 512 features of highest variance, radius of size 3. The average detection was 84.3%, 91.6%,94.1%, 95.5%, 95.5% for top 1 to top 5 respectively.

The second experiment aims to evaluate the detection rates on data taken from multiple writers. The dataset consists of 2909 instances, written by 8 different writers taken from 80 documents. The examples set consisted of 1978 instances, and the test set consisted of 931 instances. Again, the initial graphs contained 512 features with highest variance. For top 1 to top 5 hit rates reached 75.65%, 83.80%, 88.15%, 89.67%, 90.87%, respectively.

On the next series of tests, we used a different dataset generated from 5 different books, each book written by a different writer. We conducted three types of tests. First, we tested our method on 5 different datasets of size 21 types and 100 instances of each sub-word, totaling 2100 sub-words, for each writer. This test aims to evaluate our method's strength of detecting sub-words written by one writer. The second test is done on a combination of the five datasets used in the first test. The dataset is of size 10500 sub-words, and the results show the achieved detection using datasets that include sub-words written by different writers. Results can be seen in Table I, and Table II.

Book	Top1	Top2	Top3	Top4	Top5
Book 1	81.27%	88.10%	91.75%	92.86%	93.33%
Book 2	82.06%	88.73%	91.75%	93.65%	94.28%
Book 3	93.65%	96.35%	96.67%	97.62%	97.94%
Book 4	86.35%	92.86%	95.40%	96.03%	96.03%
Book 5	77.30%	87.14%	90.16%	91.75%	93.17%

TABLE I. Hit rate results of our method on each data-set taken from a book. 2100 sub-words per set.

Top1	Top2	Top3	Top4	Top5
83.40%	89.84%	92.48%	94.00%	95.11%

TABLE II. Hit rate results of our method on the combined data-set written by five writers. 10500 sub-words total.

We also applied our algorithm on the Washington dataset. This dataset consists of document images from George Washington Collection of the Library of Congress. The dataset and the corresponding query set is publicly available. The most popular experimental set is the one that uses 10 good quality pages and it has 2381 queries in total.

Although the George Washington dataset is widely used, there is no standard experimental setup, and each work evaluated the results according to their proposed algorithm. For instance, learning-based algorithms usually use cross validation to avoid evaluating the method on the same data used to fit their model. The main issue with cross validation is that each method opted to use difference number of folds of different sizes. These changes make that a direct comparison between methods impossible.

In our tests, we used the Mean average precision (mAP) score to test our spotting results. Mean average precision is a widespread measure for the performance of information retrieval systems. The metric is defined as the average of the precision value obtained after each relevant word is retrieved.

In this test we used 10 pages from the Washington collection containing 2381 images. We compared our method with other methods that used the same comparison methodology: used exact same set and queries, their methods are segmentation based, and example based. For our parameters we used an initial graph 512 vertices of highest variance, while the other parameters stayed the same as previous tests. Results can be seen at Table III.

Reference	Dataset	Method	mAP
Rothfeder *et al.* [5]	10 pages 2381 queries	Corner Feature Correspondences	0.36
Rath and Manmatha [23]	10 pages 2381 queries	Dynamic Time Warping	0.41
Proposed Method	10 pages 2381 queries	Radial Descriptor Graph	0.56
Zagoris *et al.* [17]	10 pages 2381 queries	Document Specific Local-Features	0.62

TABLE III. mAP comparison results of our method and other methods using same comparison methodology. Dataset of 10 pages with 2381 queries, and mAP.

V. Conclusions, and Future Work

We have presented a learning-free approach based on the Radial descriptor. Instead of training a model, we generate a planar graph from n dominant features. Then, we iteratively merge adjacent nodes, according to a well defined criteria, to reduce the size of the graph to $\frac{n}{2}$, effectively generating a multilevel planar graph. We present an algorithm to measure the distance between two multilevel graphs. We applied this novel technique to several data-sets in different languages and received encouraging results.

In the future, we plan to further explore and optimize the number of layers in the graph, and the size at each layer. We also plan to improve the comparison metric we've proposed to take into account the structure of the entire graph, and not just local neighborhood. We believe that this direction will provide even better results.

References

[1] M. Kassis and J. El-Sana, "Word spotting using radial descriptor," in *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*. IEEE, 2014, pp. 387–392.

[2] S. S. Kuo and O. E. Agazzi, "Keyword spotting in poorly printed documents using pseudo 2-d hidden markov models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 8, pp. 842–848, 1994.

[3] T. Rath, S. Kane, A. L. and. Partridge, and R. Manmatha, "Indexing for a digital library of george washingtons manuscripts: A study of word matching techniques," *CIIR Technical Report, University of Massachusetts Amherst.*, 2002.

[4] Y. Lu and C. L. Tan, "Word spotting in chinese document images without layout analysis," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 3, 11-15 Aug. 2002, pp. 57–60vol.3.

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

[6] D. J. A. Bhardwaj and V. Govindaraju., "Script independent word spotting in multilingual documents," in *in 2nd Intl Workshop on Cross Lingual Information Access*, 2008, p. 4854.

[7] Y. Leydier, F. Lebourgeois, and H. Emptoz, "Text search for medieval manuscript images," *Pattern Recognition.*, vol. 40, no. 12, pp. 3552–3567, 2007.

[8] A. F. S. T. Adamek, N. E. Connor, "Word matching using single closed contours for indexing historical documents," *Journal on Document Analysis and Recognition*, vol. 9, no. 2, p. 153165, 2007.

[9] 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.

[10] M. Rusinol, D. Aldavert, R. Toledo, and J. Lladós, "Browsing heterogeneous document collections by a segmentation-free word spotting method," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*. IEEE, 2011, pp. 63–67.

[11] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, "Efficient Exemplar Word Spotting," in *British Machine Vision Conference*, 2012, pp. 67.1–67.11.

[12] A. Kovalchuk, L. Wolf, and N. Dershowitz, "A simple and fast word spotting method," in *14th International Conference on Frontiers in Handwriting Recognition, ICFHR 2014, Crete, Greece, September 1-4, 2014*, 2014, pp. 3–8.

[13] S. Fiel and R. Sablatnig, "Writer retrieval and writer identification using local features," in *In Proc. 10th International Workshop on Document Analysis Systems*, March 2012, pp. 145–149.

[14] V. Dovgalecs, A. Burnett, P. Tranouez, S. Nicolas, and L. Heutte, "Spot It! Finding Words and Patterns in Historical Documents," in *12th International Conference on Document Analysis and Recognition*, 2013, pp. 1039–1043.

[15] R. Shekhar and C. Jawahar, "Word image retrieval using bag of visual words," in *Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on*. IEEE, 2012, pp. 297–301.

[16] J. Lladós, M. Rusinol, A. Fornés, D. Fernández, and A. Dutta, "On the influence of word representations for handwritten word spotting in historical documents," *International journal of pattern recognition and artificial intelligence*, vol. 26, no. 05, p. 1263002, 2012.

[17] K. Zagoris, I. Pratikakis, and B. Gatos, "Segmentation-based historical handwritten word spotting using document-specific local features," in *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*. IEEE, 2014, pp. 9–14.

[18] I. Rabaev, I. Dinstein, J. El-Sana, and K. Kedem, "Segmentation-free keyword retrieval in historical document images," in *Image Analysis and Recognition - 11th International Conference, ICIAR 2014, Vilamoura, Portugal, October 22-24, 2014, Proceedings, Part I*, 2014, pp. 369–378.

[19] M. De Berg, M. Van Kreveld, M. Overmars, and O. C. Schwarzkopf, *Computational geometry*. Springer, 2000.

[20] L. OpenCV, "Computer vision with the opencv library," *GaryBradski & Adrian Kaebler-OReilly*, 2008.

[21] J. CETIS, "Open educational resources–opportunities and challenges for higher education," 2008.

[22] M. Kassis, "The VML Arabic Historical Documents Dataset," http://www.cs.bgu.ac.il/ majeek, 2016, [Online; accessed 2016].

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