2014 14th International Conference on Frontiers in Handwriting Recognition

Word Spotting using Radial Descriptor

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

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

***Abstract*—Word spotting provides an efficient mechanism for word searching and indexing of historical documents. In this paper we present a novel feature descriptor, *radial descriptor*, and study its application for spotting word parts on Arabic historical documents. The radial descriptor aims to capture the intensity variance of the neighborhood of a point at various scale space levels. Features with high variance along multiple levels are used to describe the shape of a word according to the bag-of-features model. The distance between two word-parts is computed as the distance between their occurrence probability histograms. We have tested our approach on a large dataset of Arabic word-parts and received encouraging results.**

Keywords*—*Word Spotting, Keyword Searching, Arabic Documents, Historical Documents, Feature Descriptor

I. Introduction

Advances in digital scanning and electronic storage have driven the digitization of historical documents for preservation and analysis of cultural heritage. This process enables important knowledge to be accessible to the general public, while protecting historical documents from deterioration due to frequent handling. These documents are usually stored as a collection of images, which complicates automatic indexing or searching for keywords. To optimally utilize the digital availability of these documents, it is essential to develop an indexing and searching mechanism to replace the tedious manual processing of these tasks. One may consider using off-line handwriting recognition to convert these document images into text files. However, the research on off-line handwritten script recognition has been limited to domains with small vocabularies. The low image quality of historical documents due to diverse aging-related and deteriorative factors further complicates the recognition task. These limitations leave word spotting [1], as the main practical alternative for key-word searching in images of documents. In word spotting, pictorial representation of words are classified in a way that assigns the various pictorial representations of the same word to the same class.

In this paper we present a novel feature, the *radial descriptor*, and its application for spotting keywords in Arabic historical documents using the bag-of-features model. The radial descriptor describes the neighborhood of a point and manages to detect feature points. 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. Three distance metrics were tested: (1) χ^2 distance (2) Euclidean distance (3) Cosine similarity. Out of the three, χ^2 distance metric, provided the best results, and thus is adopted to measure the similarity between the occurrence probability histograms.

In the rest of the paper, we first review closely related work and subsequently present our approach, followed by experimental results, and performance 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], [4], [5]. Feature-based matching approaches extract features from the images to be compared and measure the similarity on the feature space [6], [7], [8], [9].

Some approaches spot words within lines, thus require the document to be segmented into lines or connected components in a preprocessing step [10], while others work directly on unsegmented pages and treat the spotting task as an image retrieval task [11], [12], [13]. Hidden Markov Models [14], and Neural Networks [15] were used by many researches for key-word searching and spotting tasks.

Rothfeder *et al.* [5] extract points-of-interest from pictorial representation of word images and draw correspondence between these points to measure similarity among such images. Srihari *et al.* [16] retrieve candidate words from the documents and rank them based on global word shape features. Yalniz and Manmatha [17] extract SIFT features, quantize their descriptors, and cluster them into visual terms using hierarchical k-means. Keyword spotting methods for handwritten documents were derived from a neural network-based system for unconstrained handwriting recognition [18]. Rath *et al.* [19], [20] extract discrete feature vectors that describe word images and use them to train a probabilistic classifier, which is then used to estimate the similarity between word images. Global word shape features were also applied to measure the similarities among words of handwritten documents [21], [22].

A segmentation-free approach was adopted by Lavrenko *et al.* [23], where they use the upper envelope

2167-6445/14 $31.00 © 2014 IEEE
DOI 10.1109/ICFHR.2014.71

and projection profile features to spot word images without segmenting them into individual characters. They show that this approach is feasible even for noisy documents. Gatos *et al.* [13] developed a segmentation-free approach for keyword search in historical documents, which combines image preprocessing, synthetic data creation, word spotting, and user feedback technologies. Moghaddam *et al.* [11] present a language independent system for preprocessing and word spotting of historical document images that did not require line and word segmentation. The distance between images is measured using Euclidean distance and dynamic time warping.

Liorente *et al.* [24], propose a direct image retrieval framework based on Markov Random Fields. They use different kernels in a non-parametric density estimation to explore semantic relationships among concepts. Kuo and Agazzi [2] present a robust algorithm for the recognition of keywords embedded in poorly printed documents using two statistical models for each keyword that represent the actual keyword and the irrelevant words. Chen *et al.* [25] present a font-independent word-spotting system, which is based on Hidden Markov Models, external shape, and internal structure of the words. Duong *et al.* [26] extract regions of interest from gray scale images and classifies them to either textual or non-textual representations using geometric and texture features.

Feature descriptor is another important field of image recognition and detection. Early approaches relied on features that are derived from the local image intensities [27], [28], [29], [30]

Scale-Invariant Feature Transform (SIFT) by Lowe [31], [32] is probably the most well-known and widely used local descriptor, many follow up works were stimulated by this feature descriptor after it was presented. Based on SIFT, [33] applied principal component analysis (PCA) to the image gradients to derive a more compact representation. [34] extended SIFT, and [35] proposed SURF as a faster alternative to SIFT, adopting similar approaches for scale and rotation invariance combined with efficient approximations to speed up the computation. Other notable descriptors are BRIEF [36], ORB [37], BRISK [38], and FREAK [39].

III. Radial Descriptor

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 depicted in Figure 1 and 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 shown in Figure 1 and 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.

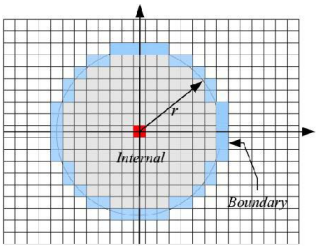

Fig. 1. The radial descriptor and its basic parameters.

It is obvious that one radius (one radial descriptor) cannot describe the neighborhood of a pixel, faithfully. Such an observation raises several questions concerning the number and range of the radii, and whether there is a need to apply any smoothing on the neighborhood or not. To address these questions we adopted the Gaussian scale space, which provides elegant solutions for these questions.

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. The radius is fixed for all levels and of value two or three. A small radius is sufficient because the multi-scale nature of the descriptor which ensures to capture

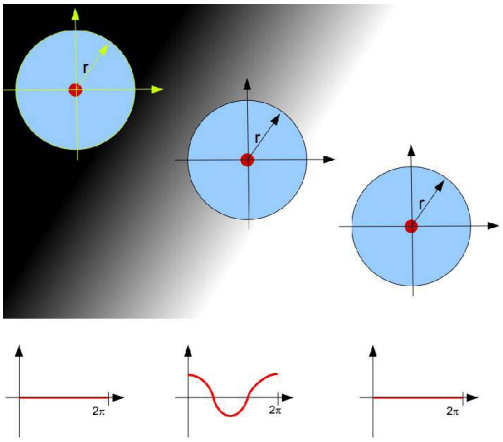

Fig. 2. The radial descriptor and selected region in an image and their corresponding radial descriptor functions.

both fine and coarse features of the neighborhood. In our tests, it is concluded that a radius of three provides the best results.

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

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, as shown in Figure 2. 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. Then only the highest k features by variance are chosen. This assures that each pictorial image has the same number of features regardless of their variance. We refer to these features, with highest variance, in a pictorial image, as the *dominant features* of the image, and are used to represent the image. The final radial function of a feature point is represented as a linear interpolation of its radial functions along the pyramid levels.

Figure 3 shows the dominant features for various thresholds. As can be seen, the dominant features stick to the presumable boundary of the pattern; i.e., they mimic sampling the boundary of the pattern. As the number of dominant features increases, the sampling becomes denser.

Fig. 3. The dominant features left: 50, middle: 100, and right: 200

IV. Word Spotting

We apply the radial descriptor to extract the dominant features from the pictorial representations of word-parts. These features concentrate on the presumed contours of the word-parts and capture their shape in a local manner. The descriptors of these feature points are used to estimate the similarity between different patterns.

To apply Bag-of-Features [40] using the proposed radial features, we compute the radial features for each image in the training set and combine them into one set of features. The code-book is constructed by classifying these features into a set of k classes, $\{C_0, C_1, ..., C_{k-1}\}$ and a pattern is represented by its occurrence probability histogram according to the computed code-book.

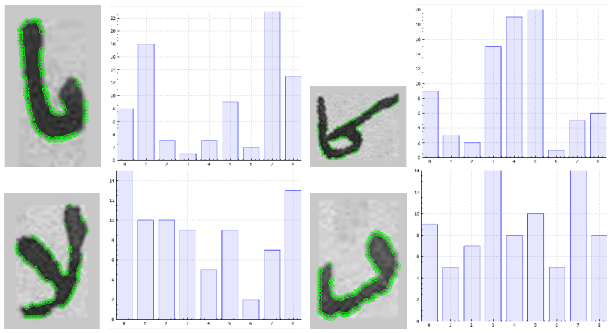

Fig. 4. Left: The image, and its chosen features denoted as green circles around high variance points. Right: The image histogram.

The similarity of two patterns is computed by measuring the distance between their occurrence probabilities, which could be done by χ^2, or any other suitable comparison method. Let P be a set of images that represent a set of Arabic word-parts. For each image we compute its radial descriptors and select the set of dominant features. We combine the features extracted from P and cluster them into n classes that represent the code-book Γ. The feature classification is performed using unsupervised learning. Expectation-maximization is used to compute the maximum likelihood estimate of the Gaussian mixture probability values for these features, and each feature is assigned the class that received the highest probability. We take the mean of each cluster, as its representative.

For each image $p \in P$, we compute the histogram of its radial features with respect to the code-book Γ and normalize it to obtain the occurrence probability. We initialize the occurrence histogram to zero, and for each feature point on p we determine the closest cluster and update the occurrence histogram accordingly.

In order to measure the distance between two images p_i and p_j, we compute the occurrence probability histograms according to the constructed code-book and calculate the distance between the two histograms using χ^2 metric.

Figure 4 presents four word-parts and their corresponding occurrence probability histogram. As can be seen, the dominant features concentrate along the boundary of the word-part, capturing its shape. The dominant features of the entire dataset are classified into nine classes that form the code-book. The histogram on the right of each image represents the occurrence probability of the dominant features of that image according to the nine classes of the code-book. It is easy to realize the difference between the various word-parts, even for a small number of classes.

V. Experimental Study

In this section we present our study to explore the ability of the radial descriptor to measure the similarity of the various pictorial representations of the same word-part and distinguish them from other word-parts. To evaluate this, we have performed various experiments on word-parts of diverse image qualities.

For the first experiment, we prepared a set of word-parts that include multiple instances of different word-parts, and

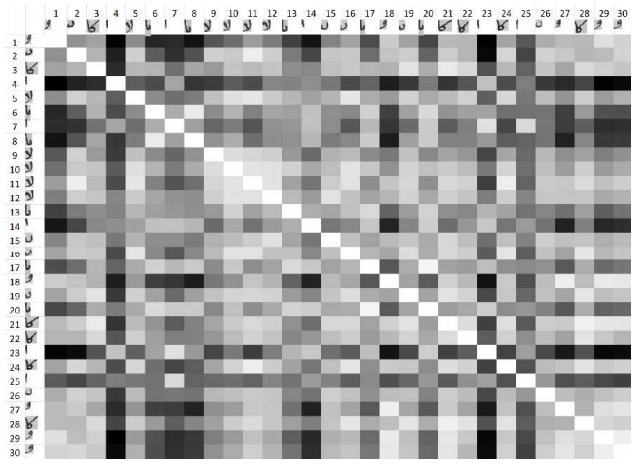

Fig. 5. The pairwise distance between 30 different word-parts depicted in gray scale, where white and black represent zero and maximal distance, respectively.

measured the pair-wise distances between them. We computed the radial descriptors for each image, I, using a radius size of three pixels, e.g. $r = 3$, and three pyramid levels. The top 80 dominant features were selected to represent I.

A code-book was generated by classifying these dominant features into nine classes and χ^2 was applied to measure the distance between the occurrence probability histogram of the word-parts. Figure 5 presents the pair-wise distances of a sample set that includes thirty different word-parts – six word-parts and five different instances of each one. White and black colors represent zero and maximal distance, respectively, and the gray color varies accordingly.

As can be seen in Figure 5, the distance between the various instances of the same word-part is usually small and much smaller than the distances from the instances of other word-parts. For example, the distance between various instances of the word-part in Figure 5 at column 5 is small, as seen at the rows $9 - 12$.

The second experiment aims to study classifying various pictorial representation of similar and different words-parts. We considered a dataset of 22 different word-parts, and for each word-part we computed the radial descriptor at three levels, using a three pixel radius, and selected the top 200 dominant features. Then we classified these features into 19 different clusters and generated the code-book. Finally, for each word-part, we compute its occurrence probability histogram.

We compared the histogram of a query image with the histogram of each word-part using the χ^2 distance metric. Figure 6 presents the distances from a query image of the

Query Pattern

Candidate	Distance	Candidate	Distance
	**0.0647**		0.1315
	0.1344		0.1445
	0.1478		0.1965
	0.2237		0.2293
	0.251		0.2685
	0.2937		0.3
	0.3108		0.3258
	0.3332		0.3447
	0.3855		0.415
	0.4751		0.5101
	0.5684		0.6386

Fig. 6. Left: The match candidate. Right: The distance between our image, and the candidate.

word-part فا and the 22 images in a descending order. As can be seen, our detection procedure performs well in estimating the distance from other word-parts.

The third experiment evaluates the performance of word-spotting. We experimented with two datasets, the word-parts on the first dataset were written by one writer and the second dataset includes word-parts written by different writers. The first consists of 594 different word-parts from 10 different documents, and the second dataset consists of 322 different word-parts from 10 different documents. For the first dataset, we chose one instance of each word-part type, and used it to generate our code-book. Then, we evaluated the classification of the remaining 572 word-parts.

Table I shows samples of the classification results of the 26 different instances of each word-part. We have tested 572 different word-parts from the first dataset, where the learning set contained one instance of each of the 22 word-part. For the learning process we chose 200 dominant features for each pictorial image and classified these features into 19 clusters. We define a hit rate of rank i if the correct detection of a query patterns is found in the best top i results. In Table I, column i of each row denotes the rate our descriptor ranked the query word-part at the top i results.

As can be seen, the presented approach successfully detected 75.52% of the 572 pictorial images when looking at the best result only, and detected 90.55%, 96.68%, 99.48%, 99.83%, when looking at best two, three, four, and best five, respectively.

The second dataset consisted of word-parts of different writers. For the training set we chose one instance of a word-part for each writer. We used 27 word-parts to generate the code-book, and evaluated the classification of the remaining

Part	Top1	Top2	Top3	Top4	Top5
	100%	100%	100%	100%	100%
	100%	100%	100%	100%	100%
	80.77%	96.15%	100%	100%	100%
	100%	100%	100%	100%	100%
	100%	100%	100%	100%	100%
	57.69%	88.46%	96.15%	100%	100%
	84.62%	100%	100%	100%	100%
	46.15%	57.69%	73.08%	96.15%	100%
	96.15%	100%	100%	100%	100%
	76.92%	96.15%	100%	100%	100%
	88.46%	100%	100%	100%	100%
	61.54%	88.46%	100%	100%	100%
	50%	69.23%	73.08%	92.31%	96.15%
	88.46%	100%	100%	100%	100%
	46.15%	76.92%	96.15%	100%	100%
	34.62%	50%	88.46%	100%	100%
	88.46%	100%	100%	100%	100%
	61.54%	100%	100%	100%	100%
	88.46%	100%	100%	100%	100%
	38.46%	69.23%	100%	100%	100%
	76.92%	100%	100%	100%	100%
	96.15%	100%	100%	100%	100%
Average	75.52%	90.56%	96.68%	99.48%	99.83%

TABLE I. DETECTION RATE OF 26 DIFFERENT INSTANCES OF EACH WORD-PART, 572 IN TOTAL. THE TRAINING SET INCLUDED ONE INSTANCE OF EACH WORD-PART.

Part	Top1	Top2	Top3	Top4	Top5
	97.96%	97.96%	97.96%	97.96%	100%
	89.8%	91.84%	93.88%	93.88%	97.96%
	46.94%	77.55%	95.92%	97.96%	97.96%
	100%	100%	100%	100%	100%
	89.8%	95.92%	97.96%	100%	100%
	69.39%	79.59%	87.76%	93.88%	97.96%
	69.23%	92.31%	100%	100%	100%
	61.22%	87.76%	95.92%	100%	100%
	38.46%	80.77%	88.46%	96.15%	96.15%
	65.38%	73.08%	80.77%	88.46%	92.31%
	26.1%	26.1%	39.13%	43.48%	52.17%
	82.61%	100%	100%	100%	100%
	46.15%	73.08%	88.46%	92.31%	100%
	87.76%	93.88%	95.92%	97.96%	100%
	51.02%	81.63%	95.92%	97.96%	97.96%
	79.59%	87.76 %	95.92%	100%	100%
	96.15%	100%	100%	100%	100%
Average	72.97%	86.49%	93.09%	95.65%	97.3%

TABLE II. DETECTION RATE OF DIFFERENT INSTANCES OF EACH WORD-PART, 666 IN TOTAL. THE TRAINING SET INCLUDED 27 WORD-PARTS. THE AVERAGE HIT-RATE IS NORMALIZED BY THE NUMBER OF INSTANCES WE ATTEMPTED TO MATCH FOR EACH WORD-PART.

666 word-parts. Table II reports the results along the same format of Table I.

Table II presents samples of the classification results of the different instances of each word-part. We have tested 666 different word-part instances, where the learning set contained 27 word-parts. We chose 200 dominant features for each pictorial image and classified these features into 19 clusters. The hit rate is 72.97%, 86.49%, 93.09%, 95.65%, 97.3%, for the ranks 1-5 respectively.

VI. PERFORMANCE

The tests were done on a laptop with Intel i7-2630QM processor, using a multi-threaded application, on Windows 8.1. The application has been implemented in C++ using OpenCV [41]. The experiment was conducted using 8 threads, the training set included 22 word-parts, and the test set included 572 word-parts. The time it took to calculate 200 features per word-part for the 22 word-parts of the entire training set is 89 seconds. On overage it took 4.045 seconds to calculate 200 features for one pictorial image. Also, to generate the code-book, the application took 9 seconds. The code-book generation is not a multi-threaded process. To detect the 572 word-parts we have used 8 threads, and the detection time is 37 minutes and 10 seconds, which is 4.23 seconds per pictorial image.

VII. CONCLUSION

In this paper we present the radial descriptor and study its application for spotting word parts on Arabic historical documents. The radial descriptor aims to capture the intensity variance of the neighborhood of a point at various scale space levels. The dominant features, are used to describe the shape of a word-part according to the bag-of-features model. The distance between two word-part images is computed as the distance between the occurrence probabilities of the two.

We studied the performance of the radial descriptor for word-spotting using image representation of Arabic handwritten word-parts and received encouraging results. The radial descriptor managed to capture the various representations of a given Arabic word-parts directly on the gray scale images.

REFERENCES

[1] R. Manmatha, C. Han, and E. M. Riseman, "Word spotting: New approach to indexing handwriting," *IEEE Computer Society Conference on Proceedings Computer Vision and Pattern Recognition (CVPR 96)*, 1996. [Online]. Available: citeseer.ist.psu.edu/article/manmatha95word.html

[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. A. Aleksander Kolcz, Joshua Alspector, "A line-oriented approach to word spotting in handwritten documents," *Pattern Analysis and Applications*, vol. 3, no. 2, pp. 153 – 168, 2000.

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

[12] Y. Leydier, F. Le Bourgeois, and H. Emptoz, "Omnilingual segmentation-free word spotting for ancient manuscripts indexation," in *Proceedings. Eighth International Conference on Document Analysis and Recognition*, 29 Aug.-1 Sept. 2005, pp. 533–537Vol.1.

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

[14] V. F. A. Fischer, A. Keller and H. Bunke, "Hmm-based word spotting in handwritten documents using subword models," in *in 20th Intl Conf. on Pattern Recognition*, 2010, p. 34163419.

[15] J. S. S. Fernandez, A. Graves, "An application of recurrent neural networks to discriminative keyword spotting," in *in 17th Intl Conf. on Artificial Neural Networks, ser. Lecture Notes in Computer Science*, vol. 4669, 2007, p. 220229.

[16] C. H. S. N. Srihari, H. Srinivasan and S. Shetty, "Spotting words in latin, devanagari and arabic scripts," *Vivek: Indian Journal of Artificial Intelligence*, vol. 16, no. 3, pp. 2–9, 2003.

[17] I. Z. Yalniz and R. Manmatha, "An efficient framework for searching text in noisy document images," in *Document Analysis Systems*, 2012, pp. 48–52.

[18] V. Frinken, A. Fischer, R. Manmatha, and H. Bunke, "A novel word spotting method based on recurrent neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 2, pp. 211–224, 2012.

[19] V. L. Toni Rath and R. Manmatha, "A statistical approach to retrieving historical manuscript images," 2003.

[20] T. Rath, V. Lavrenko, and R. Manmatha, "Retrieving historical manuscripts using shape," *Technical report, Center for Intelligent Information Retrieval, Univ. of Massachusetts Amherst.*, 2003.

[21] P. B. S. N. Srihari, H. Srinivasan 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.

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

[23] V. Lavrenko, T. M. 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)*. Washington, DC, USA: IEEE Computer Society, 2004, p. 278.

[24] A. Llorente, R. Manmatha, and S. Rüger, "Image retrieval using markov random fields and global image features," in *Proceedings of the ACM International Conference on Image and Video Retrieval*, ser. CIVR '10. New York, NY, USA: ACM, 2010, pp. 243–250.

[25] 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*, vol. 5, 27-30 April 1993, pp. 1–4vol.5.

[26] J. Duong, M. Côte, H. Emptoz, and C. Y. Suen, "Extraction of text areas in printed document images," *DocEng '01 Proceedings of the 2001 ACM Symposium on Document engineering*, pp. 157–165, 2001.

[27] C. Schmid, R. Mohr *et al.*, "Local grayvalue invariants for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 530–534, 1997.

[28] W. T. Freeman and E. H. Adelson, "The design and use of steerable filters," *IEEE Transactions on Pattern analysis and machine intelligence*, vol. 13, no. 9, pp. 891–906, 1991.

[29] F. Schaffalitzky and A. Zisserman, "Multi-view matching for unordered image sets, or how do i organize my holiday snaps?," in *Computer VisionECCV 2002*. Springer, 2002, pp. 414–431.

[30] G. Carneiro and A. D. Jepson, "Multi-scale phase-based local features," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 1. IEEE, 2003, pp. I–736.

[31] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2. Ieee, 1999, pp. 1150–1157.

[32] ——, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[33] Y. Ke and R. Sukthankar, "Pca-sift: A more distinctive representation for local image descriptors," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2. IEEE, 2004, pp. II–506.

[34] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 10, pp. 1615–1630, 2005.

[35] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 404–417.

[36] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *Computer Vision–ECCV 2010*. Springer, 2010, pp. 778–792.

[37] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: an efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2564–2571.

[38] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2548–2555.

[39] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 510–517.

[40] J. Sivic and A. Zisserman, "Video google: a text retrieval approach to object matching in videos," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, Oct 2003, pp. 1470–1477 vol.2.

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