International Journal of Pattern Recognition and Artificial Intelligence

ON THE INFLUENCE OF WORD REPRESENTATIONS FOR HANDWRITTEN WORD SPOTTING IN HISTORICAL DOCUMENTS

JOSEP LLADÓS, MARÇAL RUSIÑOL, ALICIA FORNÉS, DAVID FERNÁNDEZ, ANJAN DUTTA

Computer Vision Center, Computer Science Department
Edifici O, Universitat Autònoma de Barcelona
08193, Bellaterra, Spain
{*josep,marcal,afornes,dfernandez,adutta*}*@cvc.uab.es*
http://www.cvc.uab.es

Word spotting is the process of retrieving all instances of a queried keyword from a digital library of document images. In this paper we evaluate the performance of different word descriptors to assess the advantages and disadvantages of statistical and structural models in a framework of query-by-example word spotting in historical documents. We compare four word representation models, namely sequence alignment using DTW as a baseline reference, a bag of visual words approach as statistical model, a pseudo-structural model based on a Loci features representation, and a structural approach where words are represented by graphs. The four approaches have been tested with two collections of historical data: the George Washington database and the marriage records from the Barcelona Cathedral. We experimentally demonstrate that statistical representations generally give a better performance, however it cannot be neglected that large descriptors are difficult to be implemented in a retrieval scenario where word spotting requires the indexation of data with million word images.

Keywords: Handwriting Recognition; Word Spotting; Historical Documents; Feature Representation; Shape Descriptors.

1. Introduction

There is a huge amount of old manuscripts stored in libraries and archives. These documents have a historical value not only for their physical appearance but also for their contents. Such contents are daily studied by scholars and communities of use for history research. Many digitization initiatives exist worldwide to convert historical documents to digital libraries. The digitization process and the use of standard formats to store the images in digital libraries ensures the long-term preservation of the documents. On the other hand, the society at large can browse through collections by web-enabled technologies. But just browsing through milliards of digitized document images is not enough. Digital libraries have to provide tools to the users for mining the contained contents and unlock the wealth of information in these resources. Manually typing all the information and recording into databases is a te-

dious and slow process, requiring a lot of resources. There is still an important gap in the conversion pipeline from paper documents to useful information, especially when documents are handwritten and are degraded due to the ageing impact.

Handwriting recognition (HWR) is one of the most significant topics within the field old Document Image Analysis and Recognition (DIAR). It is, after decades, a key research activity with still many challenges towards the creation of digital libraries of historical manuscripts whose contents can be retrieved and crosslinked. Over the last years, relevant research achievements have been attained and some systems have been reported in the literature that are capable of transcribing handwritten documents up to a certain precision. Generally speaking a HWR system can be divided in two coupled components, namely the morphological and the language models. The morphological component formulates the recognition of a handwritten word as a pattern recognition problem. The language model allows to take into account the prior probabilities of the lexical units of a given lexicon. Handwritten text can be seen as a sequence of observations (e.g. characters, graphemes, or columns features) so following an analytical approach the classification of an unknown sequence X is formulated in terms of the likelihood that a feature vector sequence X is generated by the model of word W, and the prior probabilities $P(X)$ of (sub)words in the lexicon. In this framework, a large amount of data is required to train both the morphological and the language models. However, in historical documents such training data may not exist, or may require a tedious manual creation. Moreover, the information contained in a collection of manuscripts may not be useful to train recognizers for another collection because of different periods of time with different script styles, or different disciplines, with non-intersecting and restricted lexicons. On the other hand, some use cases do not require full transcription, or the existing technologies can hardly transcribe documents with poor quality. Rather than attempting to transcribe text to its full extend, only checking the existence of special keywords (names, dates, cities, etc.) may be enough.

1.1. *Word Spotting*

Handwritten word spotting is defined as the pattern analysis task which consists in finding keywords in handwritten document images. In historical documents, it offers a solution when searching information into digital libraries of documents. A typical example is when genealogists search for family names in birth records when they explore family linkages. A straightforward strategy to search words in handwritten documents would be to apply a HWR system to that document and then to search the words in the output text. However, it would be costly in terms of computation and training set requirements. Manmatha et al. proposed a different philosophy [27]. The important contribution of this work is that an image matching approach is sufficient for retrieving keywords in documents, without the need for recognizing the text. The problem is then converted to a validation problem rather than a recognition problem – validate whether a word image matches a given query

with a high score –.

Two main approaches of word spotting exist depending of the representation of the query. *Query-by-string* (QBS) [1] uses an arbitrary string as input. It typically requires a large amount of training materials since character models are learned a priori and the model for a query word is build at runtime from the models of its constituent characters. In *Query-by-example* (QBE) [27] the input is one or several exemplary images of the queried word. This is addressed as an image retrieval problem. Therefore, it does not require learning but collecting one or several examples of the keyword. QBS is more flexible because it allows searching for any keyword while QBE does not. In contrast, QBE can be performed "on-the-fly" since the user can crop a word in a document collection and start searching for similar ones. In this work we study the relevance of different representation models in a scenario of QBE handwritten word spotting in historical document collections.

1.2. *Related Work*

Due to its effectiveness, word spotting has been largely used for historical document indexing and retrieval, not only for old printed documents [28], but also for old handwritten ones [7,17,22,23,30,33].

It is important to remark that, although the use of a language model is very common in HWR, it may be useless when dealing with historical documents. It may be due to different reasons: the lack of enough training data to compute lexicon frequencies, constrained vocabularies (names, cities) or non stable over the time, "on the fly" querying (the user selects a collection and wants to search an arbitrary word in it), etc. In this scenario, a good description of the word images is a key issue, so the recognition relies only in the morphological model. In this paper we study the influence of different feature representations when recognizing handwriting. Different representations can be found in the literature for the description of word images. Similarly to shape descriptors, they can be classified into statistical and structural. The former represent the image as a n-dimensional feature vector, whereas the latter usually represent the image as a set of geometric and topological primitives and relationships among them.

Statistical descriptors are the most frequent and they can defined from global and local features. Global features are computed from the image as a whole, for example the width, height, aspect ratio, number of pixels, are global features. In contrast, local features are those which refer independently to different regions of the image or primitives extracted from it. For example, position/number of holes, valleys, dots or crosses at keypoints or regions are local features. Rath and Manmatha proposed a Dynamic Time Warping-based approach which computed profile features [31,33]: the upper and lower profiles, the number of foreground pixels, and the number of transitions black/white. This set of features showed better performance than the raw pixels as input features used in their previous work [27]. In a similar way, other approaches also use information about profiles: the QBS approach

proposed by Kesidis et al. [17] for machine-printed documents compute 90 features based on zoning plus 60 features based on upper and lower profiles. Frinken et al. [7,8] proposed a word spotting approach based on neural networks, which use the set of features proposed by Marti and Bunke [29]. For each window of 1-column width, the following 9 geometric features are computed: The 0th, 1st and 2nd moment of the black pixels distribution within the window; the position of the top-most and bottom-most black pixels (upper/lower profiles); the inclination of the top and bottom contour of the word at the actual window position; the number of vertical black/white transitions; and the average gray scale value between the top-most and bottom-most black pixel.

Gradient features are also very common: In the HMM-based word spotting approach of Rodríguez and Perronnin [34], local gradient histogram features were computed. Similarly, in the multi-lingual QBS proposed by Leydier et al. [22,23], gradient features from ZOI (zones of interest) are extracted. Moghaddam and Cheriet [30] proposed a word segmentation free approach that computes a stroke gray-level estimation, and edge profile estimation (based on gradients). Other interesting statistical descriptions are proposed in the approach of Rothfeder et al. [36] based on the Harris corner detector, or the work of Van der Zant et al. [2] where the recognition is inspired in biological features which are extracted by Gabor filters.

Holistic features can also be used for word recognition [26]. Lavrenko et al. [20] propose the use of holistic features in a simple HMM approach, with only one state per word. They compute a vector of fixed length (27 dimensions) for each word, composed of scalar features (height, width, aspect ratio, area, number of ascenders and descenders of the word), and time series (projection profile, upper and lower profile), which are approximated by the lower-order coefficients of the Discrete Fourier Transform (DFT).

Structural approaches are less common in the literature. Some early works exist using graph representations for isolated digits [5,24] or Chinese characters [12,39]. The approach proposed by Keaton et al. [15] is based on the extraction of information about concavities, although results show that this description is sensitive to noise. The word spotting approach for printed documents (old and modern) described by Marinai et al. [28], uses string encoding as the input features, and uses Self Organizing Map (SOM) for clustering.

Finally, it must be said that combinations of statistical and structural descriptors have also been proposed. Kessentini et al. [18] proposed a multi-stream HMM-based approach for off-line handwritten word recognition (tested in Latin and Arabic manuscripts), combining two different types of features: directional density features (8 from chain codes, 4 from the structure of the contour, and 3 from the position of the contour), and density features (16 from foreground densities and transitions, and 10 from concavities). The authors conclude that the combination of these types of features obtain the best results. Recently, Fischer et al. [6] proposed an interesting approach for HWR in historical documents where graph similarity

features are used as observations in a classical HMM.

1.3. *Outline of the Paper*

In this paper we focus on the morphological model alone, without using a language model. We analyze the performance of the description when dealing with word recognition. Hence, the contribution of this paper is the analysis of four families of morphological models applied to word classification in a QBE word spotting framework in historical manuscripts. In this scenario we assume that a large amount of images of digitized handwritten documents are stored in a digital library. Therefore, given a query word, there may be million of candidate words into the database. The performance of the descriptors is therefore not only assessed in terms of the retrieval quality, but also other attributes like their computational cost and their ability of being integrated in a large scale retrieval application.

As baseline model, we have selected the well known approach of Rath and Manmatha [32,33] where word images are represented as sequences of column features and aligned using a Dynamic Time Warping algorithm. We have choose this representation because it can be considered the first successful handwritten word spotting approach in the state of the art.

In addition, other description models are analyzed according to the classical taxonomy that divides pattern recognition into statistical and structural approaches. Hence, we present three different description models that can be considered statistical, pseudo-structural or hybrid, and structural, respectively. The models presented in this paper do not have to be considered as contributions by themselves, since all of them correspond to implementations of published work adapted to the problem of handwritten word representation. We do not intend to discuss about the goodness of each model, but analyze them as representatives of the corresponding families. As stated previously, the actual contribution of this paper in the evaluation of the performance of different families of morphological models when applied to a scenario of handwritten word spotting in historical documents. The three models proposed in the evaluation are: the bag of visual words representation based on SIFT features proposed by Rusiñol et al. [37], as statistical descriptor; a descriptor inspired in Loci features proposed by Fernández et al. [4], as pseudo-structural representation; and finally a graph-based representation following the work of Dutta et al. [3], as structural descriptor.

To perform the experiments, two different datasets of historical handwritten documents have been used. The first one consists of a set of 20 pages from a collection of letters by George Washington [33]. The second corpus consists of 27 pages from a 18th century collection of marriage records from the Barcelona Cathedral [4]. Both collections are accurately segmented into words and transcribed. In order to evaluate the performance of the different word representation methods in a word spotting framework we have chosen some performance assessments based on precision and recall measures.

The rest of the paper is organized as follows. In Section 2 we overview the four representations model used in the our performance evaluation protocol. Afterwards, Section 3 describes the experimentation framework with historical databases. We discuss the results obtained and the pros and cons of the different approaches. Finally, in Section 4 the conclusions are drawn.

2. Representation Models for Handwritten Words

As we have introduced in Section 1 we have implemented four models as representative of different classes to evaluate the importance of shape word representation in a word spotting context. This section overviews the different approaches. The reader is referred to the corresponding original works for detailed descriptions.

2.1. *Sequences of Column Features and DTW*

As a reference system, we have chosen the well-known word spotting approach described by Rath and Manmatha [32,33], which is based on the Dynamic Time Warping algorithm. The Dynamic Time Warping (DTW) algorithm was first introduced by Kruskal and Liberman [19] for putting samples into correspondence in the context of speech recognition. It is a robust distance measure for time series, allowing similar samples to match even if they are out of phase in the time axis (see Figure 1). DTW can distort (or warp) the time axis, compressing it at some places and expanding it at others, finding the best matching between two samples.

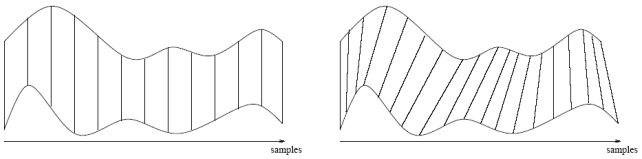


Fig. 1. Bin-to-bin and DTW alignment.

In order to apply DTW for word spotting, a common practice is to represent the word images as a sequence of column-wise feature vectors. In the word spotting approach described by Rath and Manmatha [33], the following four features are computed for every column of a word image:

- the number of foreground pixels in every column.
- the upper profile (the distance of the upper pixel in the column to the upper boundary of the word's bounding box).
- the lower profile (the distance of the lower pixel in the column to the lower boundary of the word's bounding box).
- the number of transitions from background to foreground and vice versa.

In this way, two word images can be easily compared using DTW. Given a word image A with M columns and a word image B with N columns, a feature vector for each column is computed and normalized [0,1]. Afterwards, the matrix $D(i,j)$ (where $i = 1..M, j = 1..N$) of distances is computed using dynamic programming:

$$D(i,j) = min \left\{ \begin{array}{l} D(i,j-1) \\ D(i-1,j) \\ D(i-1,j-1) \end{array} \right\} + d2(x_i, y_j) \tag{1}$$

$$d2(x_i, y_j) = \sum_{k=1}^{4} (f_k(a_i) - f_k(b_j))^2 \tag{2}$$

where $f_k(a_i)$ corresponds to the k-th feature of the column i of the image A, and $f_k(b_j)$ corresponds to the k-th feature of the column j of the image B.

Performing backtracking along the minimum cost index pairs (i,j) starting from (M,N) yields the warping path. Finally, the matching distance $DTWCost(A,B)$ is normalized by the length Z of this warping path, otherwise longest time series should have a higher matching cost than shorter ones:

$$DTWCost(A,B) = D(M,N)/Z \tag{3}$$

For word spotting, every query word is compared with the word images in the database using the DTW algorithm.

DTW is robust to the elastic deformations typically found in handwritten words, because the algorithm is able to distort the "time" axis for finding the best matching. In addition, DTW is able to handle word images of unequal length, allowing the comparison without resampling. Although there are some efficient and fast DTW approaches in the literature [16,38], the original DTW is usually considered very time consuming. The classic DTW method requires the computation of a matrix for comparing each pair of words, which has a complexity order of $O(n^2)$. Therefore, in case of large datasets, faster similarity measurements among words would be preferable.

2.2. *Bag-of-visual-words Descriptor*

In this section, we give the details of a word image representation that is based on the bag-of-visual-words (BoVW) model powered by SIFT [25] descriptors. The BoVW model has already been used for handwritten word representation [37]. The main motivation of representing handwritten words by this model is that in other Computer Vision scenarios it has obtained a good performance albeit its simplicity. The advantages of the presented method are twofold. On the one hand, by using the BoVW model we achieve robustness to occlusions or image deformations. This is a great advantage in the context of word representations for spotting applications

since we can handle noisy word segmentations. On the other hand the use of local descriptors adds invariance to changes of illumination or image noise. This is important when dealing with historical documents, since the image conditions might vary from page to page. Besides, the descriptors obtained by the BoVW model can be compared using standard distances and subsequently any statistical pattern recognition technique can be applied.

We detail below the following steps for the BoVW handwritten word representation. Initially we need a reference set of some of the word images in order to perform a clustering of the SIFT descriptors to build the codebook. Once we have the codebook, the word images can be encoded by the BoVW model. In a last step, in order to produce more robust word descriptors, we add some coarse spatial information to the orderless BoVW model.

2.2.1. *Codebook generation*

For each word image in the reference set, we densely calculate the SIFT descriptors over a regular grid of 5 pixels by using the method presented by Fulkerson et al. [9]. Three different scales using bin sizes of 3, 5 and 10 pixels size are considered. These parameters are related to the word size, and in our case have been experimentally set. We can see in Figure 2 an example of dense SIFT features extracted from a word image. The selected scales guarantee that a descriptor either covers part of a character, a complete character or a character and its surroundings. As shown by Zhang et al. [41], the larger amount of descriptors we extract from an image, the better the performance of the BoVW model is. Therefore, a dense sampling strategy has a clear advantage over approaches defining their regions by using interest points. Since the descriptors are densely sampled, some SIFT descriptors calculated in low textured regions are unreliable. Therefore, descriptors having a low gradient magnitude before normalization are directly discarded.

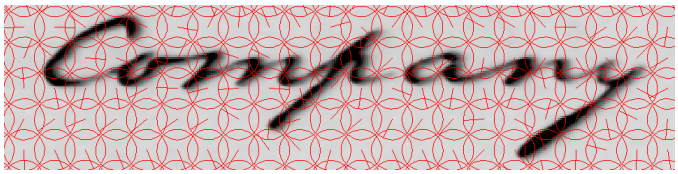


Fig. 2. Dense SIFT features extracted from a word image.

Once the SIFT descriptors are calculated, by clustering the descriptor feature space into k clusters we obtain the codebook that quantizes SIFT feature vectors into visual words. We use the k-means algorithm to perform the clustering of the feature vectors. In the experiments carried out in this paper, we use a codebook with dimensionality of $k = 20.000$ visual words.

2.2.2. *BoVW feature vectors*

For each of the word images, we extract the SIFT descriptors, and we quantize them into visual words with the codebook. Then, the visual word associated to a descriptor corresponds to the index of the cluster that the descriptor belongs to. The BoVW feature vector for a given word is then computed by counting the occurrences of each of the visual words in the image.

2.2.3. *Spatial information*

The main drawback of bag-of-words-based models is that they do not take into account the spatial distribution of the features. In order to add spatial information to the orderless BoVW model, Lazebnik et al. [21] proposed the Spatial Pyramid Matching (SPM) method. This method roughly takes into account the visual word distribution over the images by creating a pyramid of spatial bins.

This pyramid is recursively constructed by splitting the images in spatial bins following the vertical and horizontal axis. At each spatial bin, a different BoVW histogram is extracted. The resulting descriptor is obtained by concatenating all the BoVW histograms. Therefore, the final dimensionality of the descriptor is determined by the number of levels used to build the pyramid.

In our experiments, we have adapted the idea of SPM to be used in the context of handwritten word representation. We use the SPM configuration presented in Figure 3 where two different levels are used. The first level is the whole word image and in the second level we divide it in its right and left part and its upper, central and lower part. With the proposed configuration we aim to capture information about the ascenders and descenders of the words as well as information about the right and left parts of the words.

Since the amount of visual words assigned to each bin is lower at higher levels of the pyramid, due to the fact that the spatial bins are smaller, the visual words contribution is usually weighted. In our configuration, the visual words in the second level appearing in the upper and lower parts are weighted by $w = 8$ whereas the ones appearing in the central parts are weighted by $w = 4$, while in the first level we use a weight $w = 1$.

Since we used a two levels SPM with 7 spatial bins, we therefore obtain a final a descriptor of 140.000 dimensions for each word image.

2.2.4. *Normalization and Distance Computation*

Finally, all the word descriptors are normalized by using the $L2$-norm. In order to assess whether two word images are similar or not, we use the cosine distance between its feature vectors.

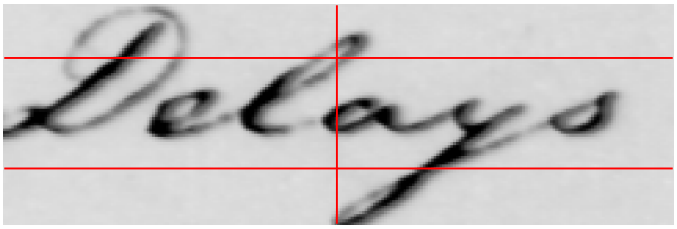


Fig. 3. Second level of the proposed SPM configuration. Ascenders and descenders information and right and left parts of the words is captured.

2.3. *Pseudo-Structural Descriptor*

Fernández et al. have proposed a pseudo-structural descriptor for word spotting [4] based in the characteristic Loci features proposed by Glucksman [11] for the classification of mixed-font alphabets. A characteristic Loci feature consists of the number of the intersections in four directions: up, down, right and left (see Figure 4). For each background pixel in a binary image, and each direction, we count the number of intersections (an intersection means a black/white transition between two consecutive pixels). Hence, each keypoint generates a code, called *Locu number*, of length 4. Characteristic Loci features were designed for digit and isolated letter recognition. In our proposal we adapt the idea to handwritten word images. Thus, we are encoding more complex images, with a potentially high number of classes, and intraclass variability due to the nature of handwriting. To cope with this, we have introduced three variations in the basic descriptor:

- We have added the two diagonal directions, as we can see in Figure 4. This gives more information to the descriptor and more sturdiness to the method.
- The range of the number of the intersections is quantized and bounded in intervals. It allows a compact representation when the indexation structure is constructed because similar Locu numbers are clustered in the same bucket.
- Two modes are implemented to compute the feature vector, namely background and foreground pixels are taken as keypoints.

Before encoding word images by Locu numbers, a preprocessing step binarizes the image and the skeletons are computed by an iterative thinning operator until lines of width of 1 pixel are obtained.

The feature vector (Locu number) is computed by assigning a label to each background (or foreground) pixel, referred as keypoints, as it is shown in Figure 4. We generate a Locu number of length 8 for each keypoint. These values correspond to the counts of intersections with the skeletonized image along the 8 directions starting from the keypoint. A word image is finally encoded with a histogram of Locu numbers. Since the histograms of Locu numbers are integrated in an indexation

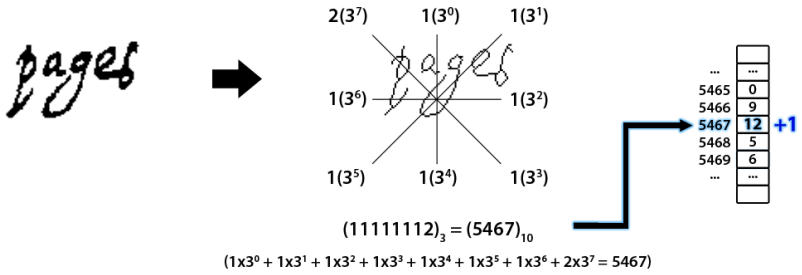


Fig. 4. Characteristic Loci feature of a single point of the word page.

Table 1. Intervals for each direction in characteristic Loci feature.

	Values		
Direction	0	1	2
Vertical	{0}	[1, 2]	[3, $+\infty$]
Horizontal	{0}	[1, 4]	[5, $+\infty$]
Diagonal	{0}	[1, 3]	[4, $+\infty$]

structure, we reduce the dynamic range of Locu numbers (length of histograms). The length of the histograms of Locu numbers exponentially increases when the number of possible intersection counts is higher. By delimiting the number of possible values we reduce the number of combinations, and consequently the computational cost. For example, with 3 possible values and 8 directions, we obtain 3^8 (6.561) combinations; with 4 possible values we have 4^8 (65.536). Thus, to reduce the dimension of the feature space, the counts for the number of intersections have been limited to 3 values (0, 1 and 2).

Contrarily to the original work [11], we do not define fixed upper bounds for intersection counts. The casting from the actual count to the range $[0 \dots 3]$ is defined independently for each direction according to different intervals. The horizontal direction has a larger interval than the vertical direction. In the original approach the digits or characters have a similar height and width, but in our approach the width of the words is usually bigger than the height. According to the dimensions of the words the range of the intervals are directly proportional. Diagonal directions are a combination of the two other directions. Table 1 shows the intervals for each direction.

According to the above encoding, for each keypoint (background or foreground pixel), an eight-digit number in base 3 is obtained. For instance, the Locu number of point P in Figure 4 is $(22111122)_3 = (6170)_{10}$. It generates a vote in the corresponding entry of the histogram of Locu numbers for this word image. Since

the Locu numbers range between 0 and 6561 (= 3^8), the length of histograms (dimension of the feature space) is 6561. The histograms are used as indexation keys in the word spotting process. The indexation strategy is out of the scope of this paper. The reader can find further details in the original work [4].

2.4. *Graph-based Descriptor*

Graphs are widely adapted by the research community since a long back as a robust tool to represent structural information of images. Graph theory provides efficient methods to compare structural representations by means of a graph isomorphism formulation. However, graph representations are rarely used in HWR. The inherent elastic variability of handwritten strokes has to be modeled in the graphs and therefore the recognition involves the implementation of an error-tolerant graph isomorphism, which is a computationally expensive problem (it belongs to the class of NP-complete problems). In addition, the use of graphs in a word spotting problem would require multiple matchings between the query graph and the graphs representing words in the database.

To have a complete view of the performance of the different families of representations, we have adapted a graph matching approach [3] to the word spotting problem. The graph representation for a word image is constructed from its skeletal information. The skeleton of the image is polygonally approximated using a vectorization algorithm [35]. This particular vectorization algorithm works only with a pruning parameter which is used to eliminate the very small noise from the image. For our case we set the parameter to 5 i.e. all isolated pixels having total pixel size less than or equal to 5 will be eliminated. Thus, graph nodes correspond to critical points in the skeleton (extrema, corners or crossings) and the lines joining them are considered as the edges of the graph (see Figure 5). As it can be seen in the example, when the skeletons are vectorized a rough polygonal approximation is intentionally done to keep the main structure properties in the graph but neglecting the details. Otherwise, two instances of the same word would be converted into very different graphs.

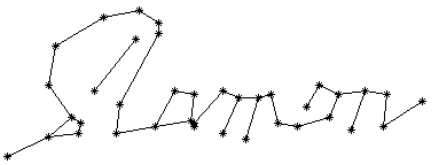


Fig. 5. Graph representation from the skeleton of a word. The critical points detected by the vectorization method are considered as the nodes and the lines joining them are considered as the edges.

Once words are represented by graphs, word spotting is solved by a graph match-

ing algorithm. To avoid the computational burden, we have proposed a method based on graph serialization. Graph serialization consists in decomposing graphs in one dimensional structures like attributed strings, so graph matching can be reduced to the combination of matchings of one dimensional structures. In addition, graphs are factorized, i.e. represented in terms of common one-dimensional substructures. Thus, given a graph, we serialize it extracting its acyclic paths. Paths are then clustered, so the representation can be seen as a bag-of-paths (BoP) descriptor for describing the words.

An acyclic path between any two connected nodes in a graph is the sequence of line segments from the source node to the destination following the order. For describing a word with the BoP descriptors, we compute all possible acyclic paths between each pair of the connected nodes in a graph corresponding to a word image and repeat the same procedure for all the word images to be considered for recognition. A vector of attributes is associated to each path capturing its shape information. For our case we use the Zernike moments descriptors of order 7 for that purpose. These settings are experimentally chosen to give the best performance. Let us call the set of descriptors of all the graph paths as P, and a set $S \subseteq P$ of prototype paths are selected by a random prototype selection technique from the set P. Then each of the paths in a single word are classified as one of the prototype paths and ultimately represent the graph presenting a word as the histogram of number of occurrences of the prototype paths (see Figure 6). For this experiment, we select $|S| = 500$, this actually determines the dimension of the BoP descriptors, this parameter is also chosen to give the best performance. The Euclidean distance is used to compare two BoP representations.

3. Experimental Results

To carry out the performance study of the different methods we used two different data collections of historical handwritten documents. Let us first give the details of the collections and the evaluation measures and then proceed with the result analysis.

3.1. *Datasets and Evaluation Measures*

To perform the experiments, we used two different datasets of historical handwritten documents. The first one consists of a set of 20 pages from a collection of letters by George Washington [33]. The second evaluation corpus contains 27 pages from a collection of marriage registers from the Barcelona Cathedral [4]. Both collections are accurately segmented into words and transcribed. In the George Washington collection we have a total of 4860 segmented words with 1124 different transcriptions whereas in the Barcelona Cathedral collection we have 6544 word snippets with 1751 different transcriptions. All the words having at least three characters and appearing at least ten times in the collections were selected as queries. For the George Washington collection we have 1847 queries corresponding to 68 different words

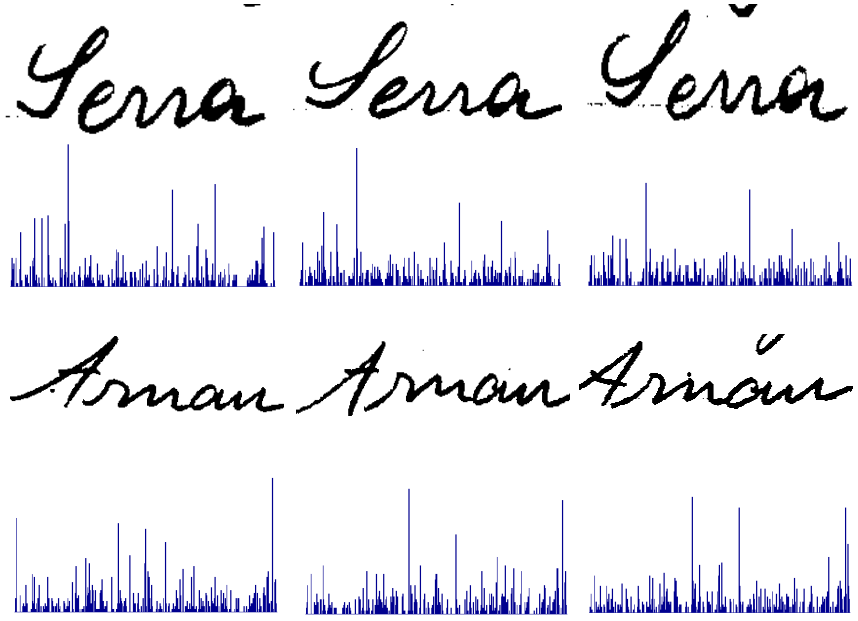


Fig. 6. The histogram of the number of occurrences of the prototype paths represents the graphs extracted from the handwritten word image.

and for the Barcelona Cathedral collection we have 514 queries from 32 different words. Both subsets selected for the experiments are single writer collections.

In order to evaluate the performance of the different word representation methods in a word spotting framework we have chosen some performance assessments based on precision and recall [40] measures. Let us briefly review the used measures.

Given a query, let us label as *rel* the set of relevant objects with regard to the query and as *ret* the set of retrieved elements from the database. The precision and recall ratios are then defined as follows:

$$Precision = \frac{|ret \cap rel|}{|ret|}, \qquad Recall = \frac{|ret \cap rel|}{|rel|}. \tag{4}$$

In order to give a better idea on how the different systems rank the relevant words we use the following metrics. The precision at n or $P@n$ is obtained by computing the precision at a given cut-off rank, considering only the n topmost results returned by the system. In this evaluation we will provide the results at the 10 and 20 ranks. In a similar fashion, the R-precision is the precision computed at the R-th position in the ranking of results, being R the number of relevant words for that specific query. The R-precision is usually correlated with the mean average precision mAP, which is computed using each precision value after truncating at each relevant item in the ranked list. For a given query, let $\mathrm{r}(n)$ be a binary function

Table 2. Retrieval results for the George Washington collection.

	P@10	P@20	R-precision	mAP	$nDCG$
DTW	0.346	0.286	0.191	0.169	0.539
BoVW	0.606	0.523	0.412	0.422	0.726
Pseudo-Struct	0.183	0.149	0.096	0.072	0.431
Structural	0.059	0.049	0.036	0.028	0.362

on the relevance of the n-th item in the returned ranked list, the mean average precision is defined as follows:

$$mAP = \frac{\sum_{n=1}^{|ret|}(P@n \times \mathrm{r}(n))}{|rel|}. \tag{5}$$

Finally, we also use the normalized discounted cumulative gain [13] $nDCG$ evaluation measure. For a given query, the discounted cumulative gain DCG_n at a particular rank position n is defined as:

$$DCG_n = rel_1 + \sum_{i=2}^{n} \frac{rel_i}{\log_2 i}. \tag{6}$$

The normalized version, $nDCG$, is obtained by producing an ideal DCG for the position n. It is computed as:

$$nDCG_n = \frac{DCG_n}{IDCG_n}. \tag{7}$$

3.2. *Results*

We present in Figures 7 and 8 an example of the qualitative results for a given query for the Barcelona Cathedral and the George Washington collection respectively. We can see that all the methods present some false positives in the first ten responses. However, it is interesting to notice that this false positive words are in most of the cases similar to the query in terms of shape. In Figure 7, when asking for the word "*Farrer*", we obtain similar results such as "*Farer*", "*Ferrer*", "*Carner*", "*Fuster*" or "*Serra*". In the case of the George Washington collection, the behavior is similar. When asking for the word "*Company*" we obtain as false alarms similar words as "*Conway*", "*Commissary*", "*Companies*", "*Country*", "*complete*", etc.

In Figure 9 we present the precision and recall plots for both collections and in Tables 2 and 3 we present the evaluation measures for the George Washington and the Barcelona Cathedral collection respectively.

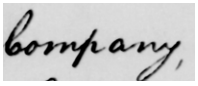
We can see that in both scenarios, the BoVW method outperforms the other three. It is however interesting to notice that the George Washington collection

Fig. 7. Example of qualitative results for the Barcelona Cathedral collection. (a) DTW-based method, (b) BoVW method, (c) Pseudo-structural method and (d) Structural method.

Table 3. Retrieval results for the Barcelona Cathedral collection.

	P@10	*P*@20	*R*-precision	*mAP*	*nDCG*
DTW	0.288	0.201	0.214	0.192	0.503
BoVW	0.378	0.289	0.303	0.3	0.592
Pseudo-Struct	0.273	0.189	0.199	0.178	0.476
Structural	0.155	0.12	0.118	0.097	0.382

Query:



Results:

a) b) c) d)

Fig. 8. Example of qualitative results for the George Washington collection. (a) DTW-based method, (b) BoVW method, (c) Pseudo-structural method and (d) Structural method.

seems to be easier for the BoVW method whereas the Barcelona Cathedral is the one where we obtain better results for the other three methods. The BoVW method extracts the word representation directly from the gray-level word images, whereas the DTW-based, the pseudo-structural and the structural methods need a preliminary step of binarization. In the George Washington collection there are many degraded word images that might be difficult to binarize hindering the performance of the word representations that need a preprocessing step including binarization. We can see an example of the performance of the Otsu binarization process for the two collections in Figure 11. As we can appreciate, the binarization process in the Barcelona Cathedral dataset is quite robust and stable, whereas the George

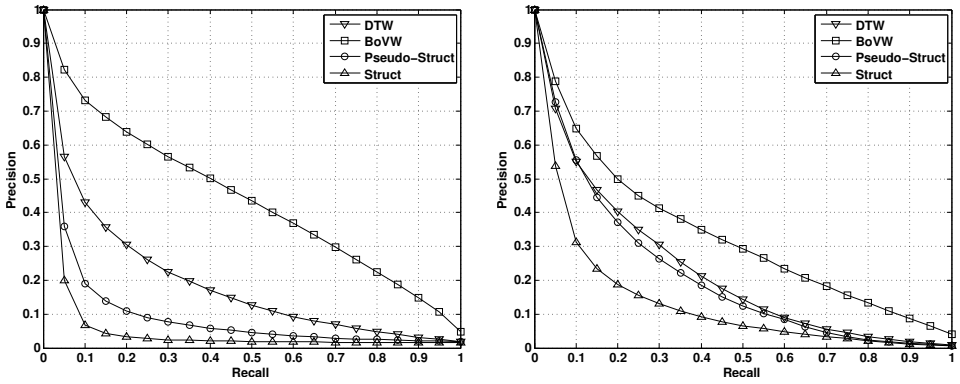


Fig. 9. Precision and Recall curves for the a) George Washington collection and b) Barcelona Cathedral collection.

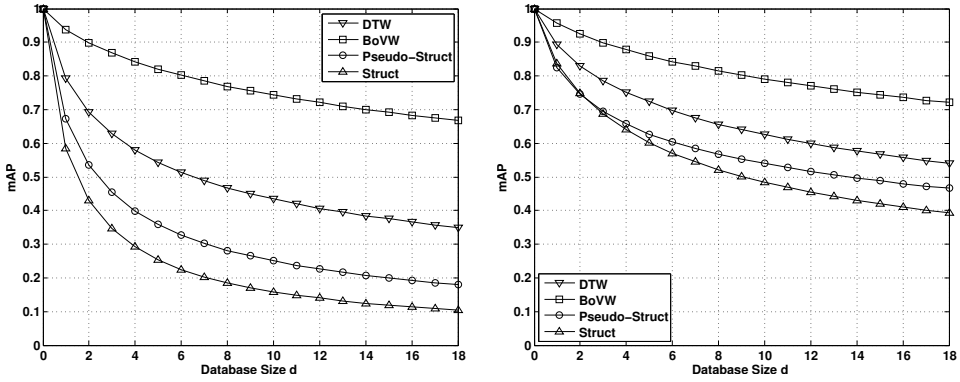


Fig. 10. Scalability test results for the a) George Washington collection and b) Barcelona Cathedral collection.

Washington collection is more noisy and the binarization performs quite different for words from the same class.

Even if the BoVW method performs better in both cases than the rest, it also presents an important drawback. Its feature vector is huge (140.000 dimensions) respect the other three methods. This is an issue when facing large collections since the feature vectors must fit in memory for real-time retrieval. Binarization techniques such as LSH[10] or product quantization[14] should be taken into account. However, even if those indexing methods will reduce the size of the descriptor so it can be used in real scenarios, they will also harm its discriminative power decreasing its performance.

Notice that the pseudo-structural method performs very close to the DTW

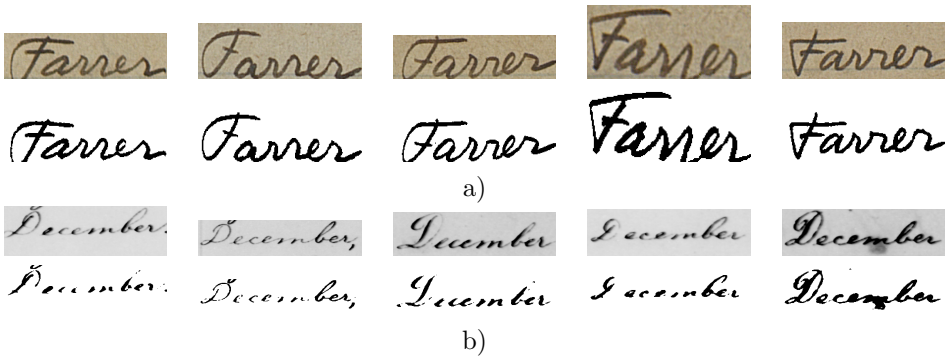


Fig. 11. Binarization examples for different word instances of the same class for the a) George Washington collection and b) Barcelona Cathedral collection.

reference method for the Barcelona Cathedral collection. However, DTW is very time consuming. Computing distances among words with the DTW method is very time consuming since the algorithm has a $O(n^2)$ complexity. On the other hand, the pseudo-structural word representation is a compact signature that can be easily indexed, and thus scales much better than a DTW-based method.

Finally, we can see that the structural method performs poorly in both collections. This word representation needs a preliminary step of transforming the raster image to a vectorial image in order to build a graph representing the words. When working with historical documents, slight degradations affect too much to the vectorization process yielding to an important performance loss of this word representation. However, it seems from the performance that the vectorization step does not have much effect on the performance but in time complexity for computing the paths. This can happen due to the appearance of noise in the image for which the vectorization method can create spurious critical points which increases the path computation time. Since the appearance of spurious points does not change the overall structure of a path, the performance will not change drastically.

In Figure 10 we can see the results for a scalability test. For each given query, having R relevant words in the collection we performed the retrieval experiments by iteratively increasing the negative samples by a factor of R in the considered dataset. In the first step, $d = 0$, the collection has just the R relevant items, so all the methods yield a mean average precision of 1. In the next iteration, $d = 1$ we add a random selection of R non-relevant items. At $d = 2$ the collection consist of the R relevant words and $2 \times R$ non-relevant words, and so on. We can see that again the BoVW method outperforms the rest in both collections. The pseudo-structural and structural methods for the George Washington collection present a sudden drop in performance at the early steps, that is that adding just a few negative samples to the dataset already hinders a lot the performance of these methods. On the other hand, for the Barcelona Cathedral collection, increasing the dataset do not entail such a

Table 4. Pros and Cons Summary.

	Size	Time	Indexability	Preprocessing	Performance	Scalability
BoVW	−−	+	+	+	++	− (size)
DTW	+	−−	−	−	+	− (time)
Pseudo-structural	++	++	++	−	+	−(discriminative power)
Structural	++	++	++	−−	−	− (discriminative power)

performance loss and both the pseudo-structural and the structural method have the same behavior in the early stages. We can also note that even if the methods present a sudden performance drop at the early stages, the performance tends to remain stable in the last stages.

In order to summarize the analysis of the results of this study, we present in Table 4 a summary of the pros and cons of each of the word representations. For each of the methods we highlight their strengths and the weaknesses. Concerning the complexity of the methods, both the BoVW and the DTW-based methods present some defficiencies. The feature vectors from the BoVW method occupy a lot of memory whereas the time complexity for the distance computation in the DTW method is also high. On the other hand, both the structural and pseudo-structural methods are efficient in terms of space and time, which makes them a good candidates for being indexed. As we shown above, the need of preprocessing steps might severely hinder the performance of the methods. In that sense, the BoVW method, which can be directly computed from the gray-level images, is a better choice than the methods that need a binarization step, and much better that the structural method that needs a vectorization step in order to build the word graphs. This is also reflected by the observed performance of these methods. Finally, notice that none of the word representations under study scale well, either by its high-dimensionality, by the complexity of the distance computation or by their performance when increasing the dataset size.

4. Conclusions

Handwriting recognition approaches usually combine morphological (i.e. shape) and language models. When dealing with historical documents, for different reasons, not always the language model is neither available nor useful. In word spotting applications, when the purpose is to retrieve a page containing a queried word from a large database, the representation of the shape of the word takes a high relevance. In this paper we have compared different descriptors to assess the strengths and weaknesses of statistical and structural models when recognizing handwritten words in a word spotting application. We have compared four approaches, namely sequence alignment using DTW as a classical reference, a bag of visual words approach as statistical model, a pseudo-structural model based on a Loci features

representation, and a structural approach where words are represented by graphs. The four approaches have been tested with two collections of historical data.

The overall conclusion we can draw according to the results described in Section 3 is that the better performance is achieved with a statistical model based in the image photometry. In our work, it has been implemented with a local descriptor based on a bag of visual words representation constructed from dense SIFT features extracted from the word image. The main drawback of this approach is its high memory requirements to store such a large descriptor. This makes it unable for a retrieval framework unless that dimensionality reduction and hashing strategies are implemented, which would harm its performance. The main drawback of DTW is that it is very time consuming, so it would difficult to integrate it in a real time retrieval application. However, DTW is robust to the elastic deformations typically found in handwritten words, so it appears to be more robust in multiple writer scenarios. As we could expect, the methods based on structural features underperform the statistical ones. The ones we have implemented rely on the geometric and topological information of the image contours or skeletons. This information varies a lot among different instances of the same word class. The good points of the approaches we have presented is its ability to be integrated in a retrieval framework. Of course, it is not an intrinsic feature of structural methods.

There are many parameters to take into account if one wants to implement a word spotting application. Depending on the different characteristics that define the system, the selection of the most appropriate method can be decided. We identify the following factors as relevant when designing a word spotting system. First, the availability of a lexicon that allows to infer a language model. A language model requires enough training data statistically representative. Intra class variability and inter class separability is also a key issue. This might depend on whether we have a single or multiple writer scenario, if the number of classes is known a priori, or if the collection covers a large period of time with strong variation in script styles. When the designed application requires a real time retrieval it is important to choose a compact descriptor. In that case, the loose of performance can be compensated with a relevance feedback paradigm. The later factor is related to the ability of compiling off line the whole set of words of the collection to build an indexation structure for the retrieval process. A last consideration is the querying mode, namely query-by-example or query-by-string. The former requires learning the model from individual characters, while the latter can be performed "on the fly" asking the user to crop an example word as a query.

As a final conclusion regarding feature descriptors, there is no universal representation able to cope will all the above criteria. Although in our study a statistical descriptor outperforms the other ones, we can not neglect the use of structural information. It is also very important the indexability of the descriptor. A representation which is very accurate and has a good performance in small sets, can dramatically reduce its performance when it is used for retrieving large collections of document images.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Education and Science under projects TIN2008-04998, TIN2009-14633-C03-03 and Consolider Ingenio 2010: MIPRCV (CSD200700018), the EU project ERC-2010-AdG-20100407-269796, the grant 2009-SGR-1434 of the Generalitat de Catalunya and the research grants UAB-471-01-8/09 and AGAUR-2011FI-B-01022.

References

1. H. Cao and V. Govindaraju. Template-free word spotting in low-quality manuscripts. In *Proceedings of the Sixth International Conference on Advances in Pattern Recognition*, pages 135–139, 2007.
2. T. Van der Zant, L. Schomaker, and K. Haak. Handwritten-word spotting using biologically inspired features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1945–1957, 2008.
3. A. Dutta, J. Lladós, and U. Pal. Symbol spotting in line drawings through graph paths hashing. In *Proceedings of the Eleventh International Conference on Document Analysis and Recognition*, pages 982–986, 2011.
4. D. Fernández, J. Lladós, and A. Fornés. Handwritten word spotting in old manuscript images using a pseudo-structural descriptor organized in a hash structure. In *Pattern Recognition and Image Analysis*, volume 6669 of *Lecture Notes in Computer Science*, pages 628–635. 2011.
5. A. Filatov, A. Gitis, and I. Kil. Graph-based handwritten digit string recognition. In *Proceedings of the Third International Conference on Document Analysis and Recognition*, pages 845–849, 1995.
6. A. Fischer, K. Riesen, and H. Bunke. Graph similarity features for HMM-based handwriting recognition in historical documents. In *Proceedings of the International Conference on Frontiers in Handwriting Recognition*, pages 253–25, 2010.
7. V. Frinken, A. Fischer, H. Bunke, and R. Manmatha. Adapting BLSTM neural network based keyword spotting trained on modern data to historical documents. In *Proceedings of the Twelveth International Conference on Frontiers in Handwriting Recognition*, pages 352–357, 2010.
8. V. Frinken, A. Fischer, R. Manmatha, and H. Bunke. A novel word spotting method based on recurrent neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.
9. B. Fulkerson, A. Vedaldi, and S. Soatto. Localizing objects with smart dictionaries. In *Computer Vision - ECCV*, volume 5302 of *Lecture Notes in Computer Science*, pages 179–192. 2008.
10. A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *Proceedings of the Twentyfifth International Conference on Very Large Data Bases*, pages 518–529, 1999.
11. H.A. Glucksman. Classification of mixed-font alphabetics by characteristic loci. In *Digest of the First Annual IEEE Computer Conference*, pages 138–141, 1967.
12. A.J. Hsieh, K.C. Fan, and T.I. Fan. Bipartite weighted matching for on-line handwritten Chinese character recognition. *Pattern Recognition*, 28(2):143–151, 1995.
13. K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.
14. H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(33):117–128,

2011.
15. P. Keaton, H. Greenspan, and R. Goodman. Keyword spotting for cursive document retrieval. In *Proceedings of the Workshop on Document Image Analysis*, pages 74–81, 1997.
16. E. Keogh and C.A. Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7(3):358–386, 2005.
17. A.L. Kesidis, E. Galiotou, B. Gatos, and I. Pratikakis. A word spotting framework for historical machine-printed documents. *International Journal on Document Analysis and Recognition*, 14(2):131–144, 2011.
18. Y. Kessentini, T. Paquet, and A.M. Ben Hamadou. Off-line handwritten word recognition using multi-stream hidden Markov models. *Pattern Recognition Letters*, 31(1):60–70, 2010.
19. J.B. Kruskal and M. Liberman. The symmetric time-warping problem: From continuous to discrete. In *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*, pages 125–161, 1983.
20. V. Lavrenko, T.M. Rath, and R. Manmatha. Holistic word recognition for handwritten historical documents. In *Proceedings of the First International Workshop on Document Image Analysis for Libraries*, pages 278–287, 2004.
21. S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 2169–2178, 2006.
22. Y. Leydier, F. Lebourgeois, and H. Emptoz. Text search for medieval manuscript images. *Pattern Recognition*, 40(12):3552–3567, 2007.
23. Y. Leydier, A. Ouji, F. LeBourgeois, and H. Emptoz. Towards an omnilingual word retrieval system for ancient manuscripts. *Pattern Recognition*, 42(9):2089–2105, 2009.
24. D. López and J.M. Sempere. Handwritten digit recognition through inferring graph grammars. a first approach. In *Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition*, pages 483–491, 1998.
25. D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
26. S. Madhvanath and V. Govindaraju. The role of holistic paradigms in handwritten word recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):149–164, 2001.
27. R. Manmatha, C. Han, and E.M. Riseman. Word spotting: A new approach to indexing handwriting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 631–635, 1996.
28. S. Marinai, E. Marino, and G. Soda. Indexing and retrieval of words in old documents. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, pages 223–227, 2003.
29. U.V. Marti and H. Bunke. Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system. *International Journal on Pattern Recognition and Artificial Intelligence*, 15(1):65–90, 2001.
30. R.F. Moghaddam and M. Cheriet. Application of multi-level classifiers and clustering for automatic word spotting in historical document images. In *Proceedings of the Tenth International Conference on Document Analysis and Recognition*, pages 511–515, 2009.
31. T.M. Rath and R. Manmatha. Features for word spotting in historical manuscripts. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, pages 218–222, 2003.
32. T.M. Rath and R. Manmatha. Word image matching using dynamic time warping.

In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 521–527, 2003.
33. T.M. Rath and R. Manmatha. Word spotting for historical documents. *International Journal on Document Analysis and Recognition*, 9(2–4):139–152, 2007.
34. J.A. Rodriguez-Serrano and F. Perronnin. Handwritten word-spotting using hidden Markov models and universal vocabularies. *Pattern Recognition*, 42(9):2106–2116, 2009.
35. J. Rosin and G. West. Segmentation of edges into lines and arcs. *Image and Vision Computing*, 7(2):109–114, 1989.
36. J.L. Rothfeder, S. Feng, and T.M. Rath. Using corner feature correspondences to rank word images by similarity. In *Proceedings of the Computer Vision and Pattern Recognition Workshop*, pages 30–30, 2003.
37. M. Rusiñol, D. Aldavert, R. Toledo, and J. Lladós. Browsing heterogeneous document collections by a segmentation-free word spotting method. In *Proceedings of the Eleventh International Conference on Document Analysis and Recognition*, pages 63–67, 2011.
38. S. Salvador and P. Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.
39. P.N. Suganthan and H. Yan. Recognition of handprinted Chinese characters by constrained graph matching. *Image and Vision Computing*, 16(3):191–201, 1998.
40. C.J. van Rijsbergen. *Information retrieval*. Butterworth-Heinemann Newton, MA, USA, 1979.
41. J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal on Computer Vision*, 73(2):213–238, 2007.