Efficient Segmentation-free Keyword Spotting in Historical Document Collections

Marçal Rusiñol [*], David Aldavert, Ricardo Toledo, Josep Lladós

Computer Vision Center, Dept. Ciències de la Computació
Edifici O, Univ. Autònoma de Barcelona, 08193 Bellaterra (Barcelona), Spain

Abstract

In this paper we present an efficient segmentation-free word spotting method, applied in the context of historical document collections, that follows the query-by-example paradigm. We use a patch-based framework where local patches are described by a bag-of-visual-words model powered by SIFT descriptors. By projecting the patch descriptors to a topic space with the Latent Semantic Analysis technique and compressing the descriptors with the Product Quantization method, we are able to efficiently index the document information both in terms of memory and time. The proposed method is evaluated using four different collections of historical documents achieving good performances both on handwritten and typewritten scenarios. The yielded performances outperform the recent state-of-the-art keyword spotting approaches.

Key words: Historical Documents, Keyword Spotting, Segmentation-free, Dense SIFT Features, Latent Semantic Analysis, Product Quantization.

1 Introduction

Nowadays, in order to grant access to the contents of digital document collections, their texts are transcribed into electronic format so users can perform textual searches. When dealing with large collections, automatic transcription processes are used since a manual transcription is not a feasible solution. In

[*] Corresponding author. Tel.: +34-93-581-23-01; fax:+34-93-581-16-70.
Email addresses: `marcal@cvc.uab.cat` (Marçal Rusiñol), `aldavert@cvc.uab.cat` (David Aldavert), `ricard@cvc.uab.cat` (Ricardo Toledo), `josep@cvc.uab.cat` (Josep Lladós).

the context of digital collections of historical documents, handwriting recognition strategies [1] are applied to achieve an automatic transcription since most of those documents are manuscripts. However, handwriting recognition often do not perform satisfactorily enough in the context of historical documents. Documents presenting severe degradations or using ancient glyphs might difficult the task of recognizing individual characters, and the lexicon definition and language modeling steps are not straightforwardly solved in such context. *Keyword spotting* has become a crucial tool to provide accessibility to historical collection's contents. Keyword spotting can be defined as the pattern recognition task aimed at locating and retrieving a particular keyword from a document image collection without explicitly transcribing the whole corpus.

Two different families of keyword spotting methods can be found in the document image analysis literature. On the one hand, *learning-based* methods such as [2–4], use supervised machine learning techniques to train models of the words the user wants to spot. Those models are then used to classify whether an incoming document image contains or not one of the sought words. On the other hand, *example-based* methods such as [5–7], receive as input an instance of the keyword the user wants to retrieve from a previously indexed document image collection. Learning-based methods are preferred for applications where the keywords to spot are a priori known and fixed. If the training set is large enough they are usually able to deal with multiple writers. However, the cost of having a useful amount of annotated data available might be unbearable in most scenarios. In that sense methods running with few or none training data are preferred. It is the case of example-based methods, which are specially interesting when it is not feasible to obtain labeled data. They also present the advantage that the user is free to cast whatever query keyword he wants and is not restricted to the set of modeled words.

However, one of the main drawbacks of keyword spotting methods, either learning or example-based, is that they usually need a layout analysis step that segments the document images into words [8–11] or text lines [4,6]. But this segmentation step is not always straightforward and might be error prone. In fact, although word and text line segmentation is a quite mature research topic, it is far from being a solved problem in critical scenarios dealing with handwritten text and highly degraded documents [12,13]. Any segmentation errors affect the subsequent word representations and matching steps. This dependence on a good word segmentation motivated the researchers of the keyword spotting domain to recently move towards complete segmentation-free methods. In [14,15], Leydier et al. proposed a word spotting methodology based on local keypoints. For a given query image, interest points are extracted and encoded by a simple descriptor based on gradient information. The word spotting is then performed by trying to locate zones of the document images with similar interest points. This retrieved zones are then filtered and only the ones sharing the same spatial configuration than the query model are returned. A similar approach using SIFT keypoints for spotting both words and graphical symbols in line drawings was presented in [16] by Rusiñol and Lladós. However, directly matching local keypoints might be too computationally expensive when dealing with large datasets, and thus researchers started to apply the bag-of-visual words (BoVW) paradigm for keyword spotting purposes. For instance, Roy et al. proposed in [17] to cluster local image features into a codebook of representative character primitives for typewritten keyword spotting. Another segmentation-free word spotting method is presented in [18] by Gatos and Pratikakis. In that case, the authors propose to use a sliding-window approach with a patch descriptor that encodes pixel densities. The hypothetic locations where the queried word is likely to appear are found by a template matching strategy. The method proposed by Almazán et al. [7] presents another sliding-window approach where local patches are represented by gradient-based descriptors and the retrieval step is performed by using an exemplar support vector machine framework. In [19], Rothacker et al. combined the use of a patch based BoVW representation with HMMs to efficiently and accurately spot keywords in handwritten documents. Finally, the recent work by Howe [20] presents a multi-writer keyword spotting method that models the possible stroke distortions by inferring a generative word appearance model. The literature dealing with segmentation-free keyword spotting methods is rather scarce since it is a relatively new and unexplored research topic. However, we strongly believe that bypassing the segmentation step is a must in the context of historical document collections where achieving a perfect word or text line segmentation is unfeasible. So, architectures that dismiss the segmentation step present a clear asset in the context of historical documents.

In addition, quite often, keyword spotting methods rely on computing expensive distances exhaustively between the query and the words in the collection such as DTW [5] or learning and applying complex models such as HMMs [2,3,19] or neural networks [4]. In that sense, in large-scale scenarios, the complexity issue should to be taken into account by proposing efficient and scalable methods both in terms of memory usage and response time.

In this paper we present an efficient segmentation-free keyword spotting method based on a BoVW model powered by SIFT descriptors in a patch-based framework. Since an explicit word segmentation is avoided, the proposed method can be applied in scenarios where word segmentation might be problematic such as documents that do not follow a classical Manhattan layout, or even be used to spot handwritten annotations that do not follow a regular text line structure. Other preprocessing steps such as binarization, slant correction, etc. are also avoided, directly processing the raw image. The proposed architecture follows the query-by-example paradigm and do not involve any supervised learning method, thus do not rely on any previous content transcription. Our proposal adapts techniques that have been successfully applied in other computer vision problems to the historical documents context. By using such general representations instead of relying on hand-crafted features, both handwritten and typewritten documents are handled indifferently.

This work is a significantly extended version of our previous conference paper [21] that introduced our proposed methodology. Specifically, we have enhanced our preliminary version by including an indexation scheme aimed to scale the proposed method to handle large datasets. A multi-length patch representation is also introduced, which increases the retrieval performance by taking into account the different possible lengths of the query words. A thorough analysis and evaluation of all involved parameters of the method is presented in order to assess the configuration maximizing the retrieval performance. Finally, a performance comparison with the recent state-of-the-art literature in keyword spotting is also presented.

The remainder of this paper is organized as follows. In Section 2, we present how the document corpora are constructed and organized. We detail the feature extraction from document pages and the encoding system used in order to efficiently query the collection. Section 3 details the retrieval stage. We show how queries are treated and how regions of interest are determined within document pages. Experimental results are presented in Section 4. We study the influence of the method's parameters and compare our performance against a number of state-of-the-art keyword spotting approaches. Finally, conclusions and further research lines are drawn in Section 5.

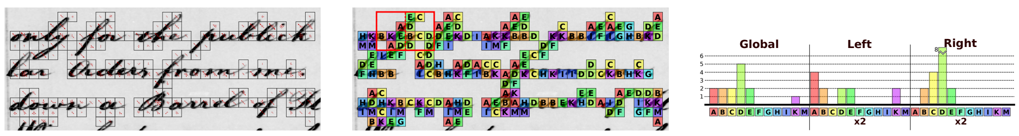
2 Off-line Corpus Representation

The word spotting problem is addressed by dividing the original document images into a set of densely sampled local patches. These local patches are the basic structure used to spot the words within the document: once a query image is given, the local patches are used to determine the page locations where the query keyword has a greater likelihood to appear. With such a procedure having an explicit word segmentation is avoided as well as any other word preprocessing steps (i.e. binarization, slant correction, etc.). These local patches must roughly match the size of the text in the document. More precisely, the height H of the local patches should roughly match the height of the text in the document. This height parameter H can be either set automatically, by for instance using a projection profile algorithm, or it can by manually set by the user.

Then, for a given height H, four different widths W_ℓ are defined in order to cope with queries of different lengths. Specifically, the geometry of the patches has been set to $H \times H$, $2H \times H$, $3H \times H$ and $4H \times H$ and are densely sampled using a regular grid of $\frac{H}{3} \times \frac{H}{3}$ pixels. The most convenient patch width will be determined at query time. This setup guarantees that there is enough overlapping between the local patches and the document words so that each word in the document is covered by at least a patch. Although a salient patch detection strategy will effectively reduce the amount of patches to be processed [18], by densely sampling them no assumption has been made on which portions of the documents are important to the final user.

2.1 Local Patch Descriptor

Local patches are described using the BoVW signature so that, first visual words are extracted from the document images. The visual words are obtained by densely sampling SIFT descriptors over the image by using the method proposed by Fulkerson et al. in [22]. The SIFT descriptors are sampled over a regular grid of 5×5 pixels at three different scales: $\frac{H}{2}$, $\frac{3H}{4}$ and H. This multi-scale representation is used to capture from fine to coarse characteristics from the word characters. The finer scale characterizes sub-parts of a character while the coarser scale characterizes whole characters and their surroundings.



a) **b)** **c)**

Fig. 1. Local patch descriptor signature: **a)** SIFT descriptors extracted from the document image. **b)** SIFT descriptors encoded into visual words and **c)** visual word accumulation in the local patch signature.

The performance of the BoVW model depends on the amount of visual words extracted from the image. In the related literature it has been noted that the larger is the amount of descriptors extracted from an image, the better the performance is [23]. Therefore, a dense sampling strategy has a clear advantage over approaches using interest points. However, a dense sampling over the image results in some SIFT descriptors calculated in low textured areas that are unreliable. In order to avoid this, descriptors having a low gradient magnitude before normalization are directly discarded.

Once the SIFT descriptors are calculated, a codebook is used to quantize them into visual words. The codebook is obtained by clustering the descriptor feature space into K different clusters by using the k-means algorithm. Then, visual words are obtained by simply assigning to each SIFT descriptor the nearest codeword of the codebook.

After SIFT descriptors have been encoded into visual words, these visual words are used to create the signatures of the local patches: a local patch p_j of the document is described by a histogram $\mathbf{f}_j = \left[f_j^1, f_j^2, ..., f_j^K\right]$ which accumulates the frequencies of each visual word within the local patch. This K-dimensional descriptors do not take into account the spatial distribution of the visual words within the local patch. This is a drawback of the BoVW representation, since words with the same letters but resorted altogether may have a very similar signature. For instance, anagrams are completely indistinguishable using this representation. Therefore, the Spatial Pyramid Matching (SPM) method proposed by Lazebnik et al. in [24] is used in order to add spatial information to the unstructured BoVW model. This method roughly takes into account the distribution of the visual words over the local patches by creating a pyramid of spatial bins.

Different spatial configurations have been evaluated and, a two level SPM with a single vertical partition, differentiating the left and the right parts of the patch, gives the best compromise between the retrieval performance and the number of dimensions of the obtained descriptor. 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 weighted according to the spatial coverage. In our case, the visual words assigned to the left and right spatial bins contribute twice to the final histogram. Finally, a local patch is described by a $3 \times K$ dimensions descriptor $\mathbf{f}_j = \left[\mathbf{f}_j^G, \mathbf{f}_j^L, \mathbf{f}_j^R\right]$, where $\mathbf{f}_j^G$, $\mathbf{f}_j^L$, $\mathbf{f}_j^R$ are the patch descriptor sub-vectors corresponding to the global, left and right spatial bins of the spatial pyramid. Finally, all the patch descriptors from the corpus are re-weighted by applying the *tf-idf* model [25] and normalized using the $\mathbf{L}_2$ norm. The *tf-idf* weighting strategy emphasizes the visual words that are frequent in a particular local patch and infrequent in the complete corpus. It assigns to each visual word f_j^i a weight in the local patch p_j given by $\text{tf-idf}_{f,p} = \text{tf}_{f,p} \times \text{idf}_f$, where $\text{tf}_{f,p}$ is the visual word frequency, i.e. the number of occurrences of visual word f_j^i in local patch p_j, and idf_f is the inverse local patch frequency computed as

$$\text{idf}_f = log\left(\frac{M}{\text{df}_f}\right),$$

where M is the total number of local patches and the document frequency df_f corresponds to the number of local patches in the collection that contain the visual word f_j^i. The different steps used to obtain the signatures of the local patches are summarized in Figure 1.

2.2 Latent Semantic Analysis Transform

Ideally, two instances of the same character are always represented by the same set of visual words. However, the clusters obtained using the k-means algorithm might not be optimal, so that some salient structures in the descriptor space might not be properly represented. Since the number of visual words of the codebook is not inferred from the descriptor space, this space may be under- or over-clustered. Besides, the shape of a character is likely to change from word to word in the context of keyword spotting, specially in handwritten documents. Therefore, the LSA technique introduced by Deerwester et al. in [26] has been applied to represent the local patch descriptors in a way which eludes unreliability, ambiguity and redundancy of individual visual words.

The LSA technique assumes that exists some underlying semantic structure in the descriptor space. This semantic structure is defined by a set of abstract *topics* where each topic is a representative distribution of visual words. The topics are estimated in an unsupervised way using the singular value decomposition (SVD) algorithm. Then, local patches are represented by a mixture of topics instead of a histogram of visual words. The goal is to obtain a transformed space where patches having similar topics but encoded by different visual words will lie close. In the context of our problem where a document is represented by millions of local patches, the LSA technique has the advantage over similar alternatives that the SVD can be calculated incrementally [27]. This allows to obtain the transformation space matrix processing the whole corpus of local patches in a very efficient way.

In order to obtain the space transformation matrix, the document patches of the global level descriptors $\mathbf{f}_j^G$ are arranged in a visual-word-by-patch matrix $\mathbf{A} \in \mathbb{R}^{K \times M}$, where K is the codebook size and M is the number of patches of the document. The LSA obtains the transformed space by decomposing the visual-words-by-patch matrix in three matrices by a truncated SVD. In order to reduce the descriptor space to T topics, where $T \ll K$, we proceed as follows:

$$\mathbf{A} \simeq \hat{\mathbf{A}} = \mathbf{U}_T \mathbf{S}_T \left(\mathbf{V}_T\right)^\top ,$$

where $\mathbf{U}_T \in \mathbb{R}^{K \times T}$, $\mathbf{S}_T \in \mathbb{R}^{T \times T}$ and $\mathbf{V}_T \in \mathbb{R}^{M \times T}$. Then, a patch descriptor $\mathbf{f}_j$ is projected into the transformed space vector $\hat{\mathbf{f}}_j$ by applying the transformation matrix $\mathbf{X}_T = \mathbf{U}_T \left(\mathbf{S}_T\right)^{-1}$ to each spatial sub-vector $\mathbf{f}_j^G$, $\mathbf{f}_j^L$, $\mathbf{f}_j^R$ separately as follows:

$$\hat{\mathbf{f}}_j = \left[{\mathbf{f}_j^G}^\top \mathbf{X}_T, {\mathbf{f}_j^L}^\top \mathbf{X}_T, {\mathbf{f}_j^R}^\top \mathbf{X}_T \right] .$$

This setup has obtained better experimental results than applying the LSA technique directly to the local patch descriptor $\mathbf{f}_j$. By separately transforming each sub-vector, the spatial information encoded by the SPM scheme is maintained.

In order to efficiently store and retrieve the patch descriptors, an indexing structure is needed. The PQ indexation framework proposed by Jégou et al. in [28] has been used. This method allows both to reduce the amount of memory needed to store the local patch descriptors by means of binary codes and to reduce the computational cost of searching the nearest neighbors by using a sub-linear approximate distance computation. The method is governed by two parameters m and c that will determine the achieved compression rates. The product quantizers decompose the local patch descriptor space into a Cartesian product of m local sub-vectors. The original $\hat{\mathbf{f}}_j$ descriptors are mapped into the $T^* = 3T/m$ dimensional sub-vectors as

$$\begin{aligned}\hat{\mathbf{f}}_j &= [\underbrace{\hat{f}_j^1, ..., \hat{f}_j^{T^*}}_{u_1(\hat{\mathbf{f}}_j)}, ..., \underbrace{\hat{f}_j^{(m-1)T^*+1}, ..., \hat{f}_j^{3T}}_{u_m(\hat{\mathbf{f}}_j)}] \\ &= \left[u_1(\hat{\mathbf{f}}_j), ..., u_m(\hat{\mathbf{f}}_j)\right].\end{aligned}$$

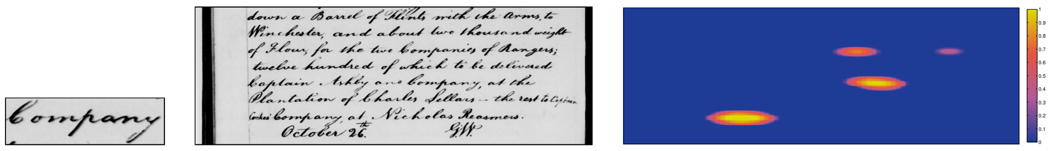
Then, each sub-space is quantized separately using c sub-quantizers. Finally, the descriptors are represented by a short code composed of its sub-space quantization indexes calculated as,

$$PQ(\hat{\mathbf{f}}_j) = \left[\kappa_1(u_1(\hat{\mathbf{f}}_j)), ..., \kappa_m(u_m(\hat{\mathbf{f}}_j))\right],$$

where $\kappa_i(\cdot)$ is the index of the sub-quantizer associated with the i-th sub-vector. For instance, if in the LSA step, the number of topics is set to $T = 512$, then the dimensionality of the patch descriptor is $3 \times T = 1536$ because of the SPM scheme. Given a PQ configuration which divides the original space into $m = 128$ sub-vectors of 12 dimensions and uses $c = 256$ sub-quantizers, the 1536-dimensional local patch descriptor is effectively represented by a 128 bytes code.

3 Word Retrieval

To perform the retrieval, the *query-by-example* paradigm is followed, where the user inputs the system a sample image of the sought word. In our segmentation free approach, a set of putative patches which are visually similar to the given query are first obtained. Then, a voting scheme aims at finding the locations within the document pages with a high likelihood to find the query word.



down a Barrel of Flints with the Arms, to Winchester, and about two thousand weight of Flour, for the two Companies of Rangers; twelve hundred of which to be delivered Captain Ashby and Company, at the Plantation of Charles Sellars — the rest to Captain Cocke's Company, at Nicholas Reasmens. October 26th. GW

Fig. 2. Example of the voting procedure. **a)** Query, **b)** sample page, **c)** obtained voting space.

3.1 Candidate Search

Given a query image which has been cropped by the user from the images in the collection, the proposed method first densely samples the SIFT descriptors. Then, quantizes them into visual words using the codebook. Afterwards, the patch descriptor is obtained by accumulating the visual words into the different bins of the spatial pyramid histograms. Subsequently, the obtained descriptor is normalized using the *tf-idf* model obtaining $\mathbf{f}_q$ which is projected into the transformed LSA space by

$$\hat{\mathbf{f}}_q = \left[{\mathbf{f}_q^G}^\top \mathbf{X}_T, {\mathbf{f}_q^L}^\top \mathbf{X}_T, {\mathbf{f}_q^R}^\top \mathbf{X}_T \right].$$

Finally, the cosine distance is computed between the query descriptor $\hat{\mathbf{f}}_q$ and the document patch descriptors $\hat{\mathbf{f}}_j$ as a similarity measure to select the patches from the documents where the query keyword is more likely to appear.

The cosine distance is calculated by using the asymmetric distance computation [28] method from the PQ framework. First, the dot product is separately calculated for each m sub-vector between the query and all the c sub-quantizers obtaining a distance matrix $\mathbf{D} \in \mathbb{R}^{m \times c}$, where the i-th row of the matrix is computed as

$$d_i = \left[\langle u_i(\hat{\mathbf{f}}_q), u_i(\hat{\mathbf{f}}_1) \rangle, ..., \langle u_i(\hat{\mathbf{f}}_q), u_i(\hat{\mathbf{f}}_c) \rangle \right],$$

where $\langle \cdot, \cdot \rangle$ is the dot product between the two sub-vectors. Then, following the multi-length scheme, the query width determines which local patches agree in terms of word length. According to that, just the most similar width W_ℓ^* to the query is taken into account. Finally, combining the PQ codes $[\kappa_1, ..., \kappa_m]$ of the selected local patches and the matrix $\mathbf{D}$, the approximated cosine distances are obtained

$$\delta_{qj} = 1 - \sum_{i=1}^{m} \mathbf{D}_{i,\kappa_i(u_i(\hat{\mathbf{f}}_j))}$$

between the query $\hat{\mathbf{f}}_q$ and the patch descriptors $\hat{\mathbf{f}}_j$ that match the query length.

3.2 Candidate Localization

Once the most similar local patches have been retrieved, the regions of the document which gather most support have to be found and selected as putative retrieved locations.

For each document page image, a 2-D voting space is constructed in where each retrieved local patch will cast its votes. In our implementation each cell of the voting space has a geometry equal to the local patch sampling step ($\frac{H}{3} \times \frac{H}{3}$ pixels). Then, each selected local patch casts a vote to the cell where its geometric center falls, weighted by the approximate distance δ_{qj}. Afterwards, the contribution at each cell of the voting space is smoothed by using an elliptic Gaussian filter $g_{\perp}(x, y; W_{\ell}^{*}, H)$. For instance, Figure 2 shows an example of the smoothed voting space obtained for a given query. Finally, the candidate locations are found by searching the local maxima (i.e. points having a greater value than their 8-neighbors) in the smoothed voting space. The resulting list of putative document regions R_D is obtained by using a priority queue which selects up to 10.000 locations having the higher associated scores. These regions are centered at the highly ranked locations and have a geometry of $W_{\ell}^{*} \times H$ pixels.

4 Experimental Results

Let us first introduce the datasets and the evaluation measures used to assess the performance of the proposed system and then analyze the obtained results.

4.1 Dataset and Evaluation Measures

In order to perform the experiments, we have used three datasets of handwritten documents and one dataset of typewritten documents. The first image corpus (GW20 dataset) consists of a set of 20 pages from a collection of letters by George Washington [5] dated 1755. Its ground-truth has a total of 4860 segmented words with 1124 different transcriptions. In order to test the scalability of the method, we have used a much larger set of images from the George Washington letters [1] composed of 1.500 pages (GW1500 dataset), however, there is no ground-truth for this dataset. The third evaluation corpus (BCN dataset) contains 50 pages from a collection of handwritten marriage licenses written in 1617 from the Barcelona Cathedral [29]. In that collection

[1] Library of Congress `http://memory.loc.gov/ammem/gwhtml/`

just some words are transcribed. We have 6735 segmented words corresponding to 21 different transcriptions. Finally, although the main aim of our method is to deal with handwritten documents, for the sake of generality, we also tested a typewritten corpus (LB dataset) consisting of a set of 20 pages from a 1825 book on Lord Byron's life [21]. In that case we have 4988 segmented words corresponding to 1569 different transcriptions. We can see an example of the four datasets in Figure 3. In terms of the document degradation, the LB collection is the most well-preserved and, since it is typewritten, it is expected to be the less challenging dataset. Between the GW20 and the BCN collections, the handwriting style in the GW20 images is less variable and the image quality is quite good, whereas the BCN collection is the most challenging one since the images present severe degradations and the variability in handwriting style is highly noticeable.

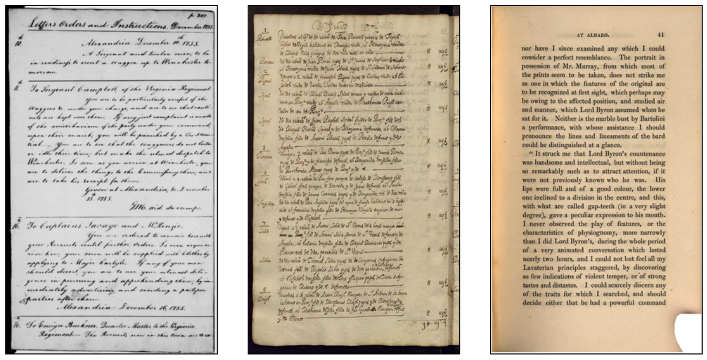


Fig. 3. Example of pages from the a) George Washington, b) Barcelona Cathedral and c) Lord Byron's collections.

In order to evaluate the performance of the spotting method we have chosen to report the mean average precision mAP and recall measures. In our case, a returned region from the documents will be considered as relevant when it overlaps at least a 50% of the sought word in the ground-truth.

Table 1
Local patch and feature geometries parameters used at each database.

		GW20	BCN	LB
Line Height		80	70	60
Feature Size	Small	40	36	32
	Medium	60	56	48
	Large	80	72	60
Patch Grid		27 × 27	24 × 24	20 × 20
Patch Geometry	Tiny	80 × 80	70 × 70	60 × 60
	Small	160 × 80	140 × 70	120 × 60
	Medium	240 × 80	210 × 70	180 × 60
	Large	320 × 80	280 × 70	240 × 60

For each database, we need to calculate the line height parameter H in order to define the geometry parameters of the local features and local patches. Table 1 summarizes the line height and the inferred feature and patch geometries that we have used in the following experiments. Such line height H has been automatically estimated by means of a projection profile analysis [12] over a subset of pages of the document collection. The text line height is obtained by calculating the median separation between peaks of the projection profile. This allows to obtain an accurate H parameter despite the possible errors of the line detection algorithm. Note that the H parameter has to roughly match the text-line height but does not need to be extremely accurate to yield good performances. Although different font sizes might be used in the documents (e.g. GW20 collection), the proposed approach remains stable.

4.2 Results

In this section, we analyze the performance of the proposed system. We organize the different carried experiments as follows. First, we will present some qualitative results to assess the effectiveness of the method to retrieve visually similar words. Then, we provide an exhaustive study of the effect of the different parameter configurations of the proposed method. Subsequently, we analyze the system's behavior in a large-scale scenario. We finally compare the obtained results with other state-of-the-art methods.

4.2.1 Qualitative Results

We present in Figure 4 some qualitative results for the four databases with an SPM-BoVW patch descriptor with a codebook of 2^{15} visual words. In a word spotting application, the chosen word descriptor should agree with the human perception when considering that two words are similar. We report here some queries where the system yields some false positive words in the first ten results. These results show that gradient-based descriptors fulfill the visual requirements in both typewritten and handwritten scenarios since the false positives (framed in red) are visually similar to the queried word. In addition, it is worth to note that even if the method does not entail any segmentation step, the retrieved regions are usually well centered over the text lines.

4.2.2 Baseline

We can see in Figure 5 the evolution of the mean average precision and recall indicators for codebook sizes from 2^8 to 2^{15} visual words and amount of topics from 2^6 to 2^9 for the three collections. The system tends to perform better with large codebooks both in terms of ranking and recall abilities. However, we can appreciate an asymptotic behavior that will indicate that a sparser

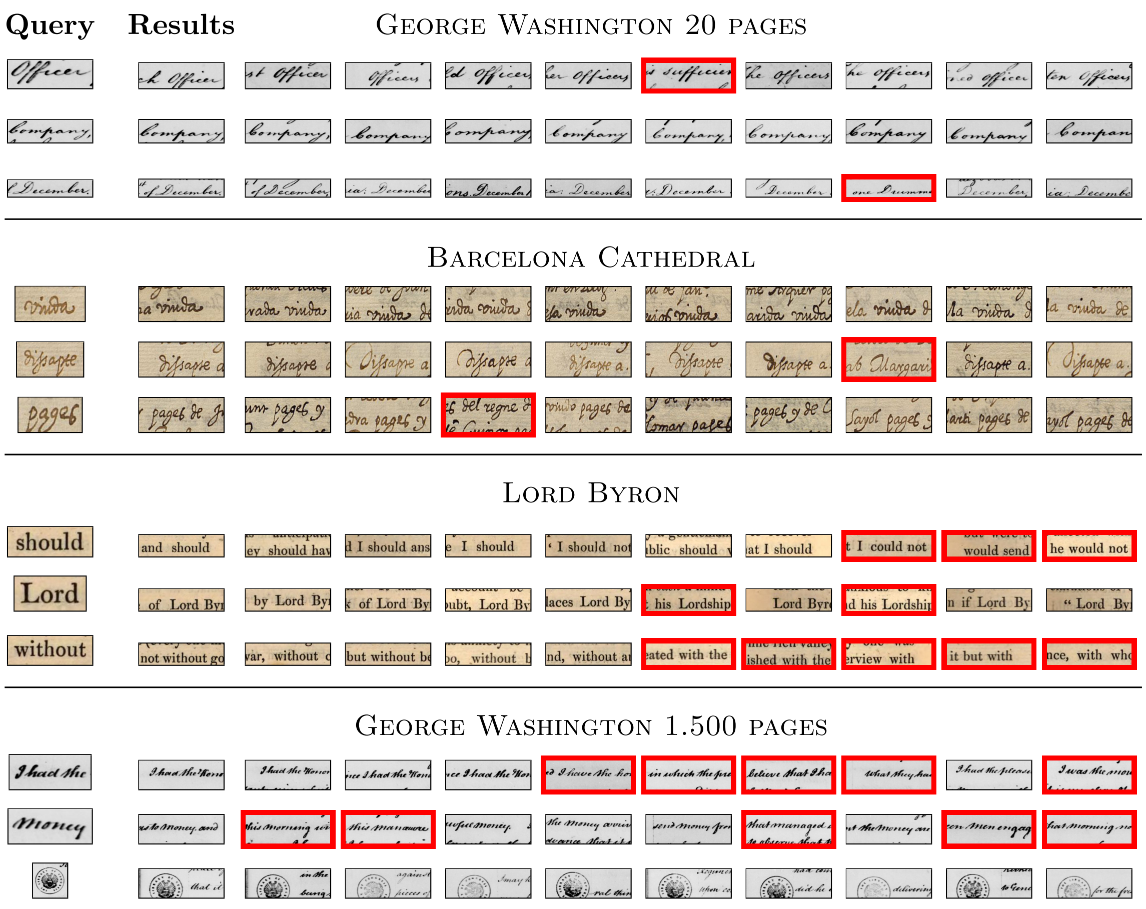


Fig. 4. 10 top-most retrieved images for some queries in the four evaluated collections.

patch representation, due to the use of too large codebooks, might hinder the system's performance. This fact is emphasized in the BCN collection, which presents more noise. Here, when using medium-sized codebooks, the system generalizes better and is able to absorb the noise whereas when we increase the vocabulary size the patch descriptor becomes more noisy and the recall is affected.

As expected, when using the LSA encoding, the greater the number of topics is, the better the system performs. Looking at the GW20 and LB experiments, the dimensionality reduction produces a small drop-off in terms of mean average precision for small codebooks that is counteracted as we increase the codebook size. However, if we look at the experiments carried with the BCN collection, an interesting phenomenon can be observed. Here, the drastic dimensionality reduction not only does not hinder the performance but provokes a significant improvement in recall against the raw descriptors. This recall increase can be attributed to the original idea of the LSA algorithm, which not only reduces the dimensionality of the descriptors but also finds relationships between different visual words corresponding to the same keyword. In noisy

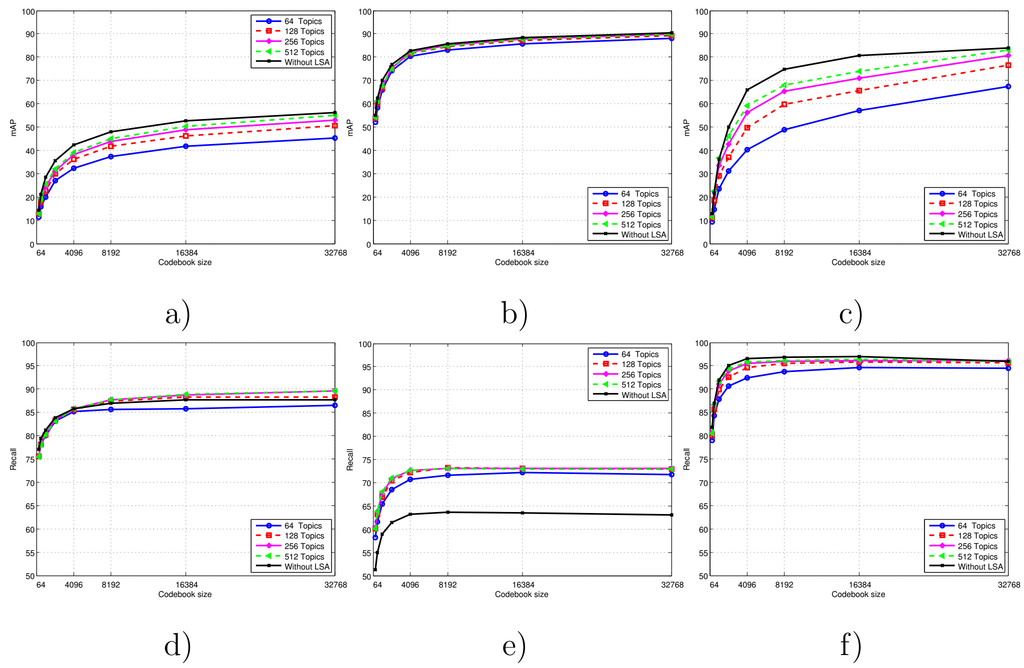


Fig. 5. Mean average precision and recall at different codebook sizes for the a) and d) George Washington, b) and e) Barcelona Cathedral and c) and f) Lord Byron's collections.

environments the use of LSA results in a more compact representation that in addition generalizes better and thus ameliorates the final performance.

The results of our baseline system with $K = 2^{15}$, $T = 512$ and using the SPM scheme are summarized in the first row of the Table 3. The obtained results clearly outperform our previous approach presented in [21] in both the GW20 and LB collections, mainly due to the increase in the codebook size and the amount of topics in the LSA encoding.

4.2.3 Compressing with Product Quantization

Each patch from the documents in the baseline system is described by a 1536-dimensional double-valued feature vector, thus occupying 12.288 bytes in memory. Each page having in average more than 10.000 patches, we need approximately 120MB to store each page from the collection in memory. Since managing such amount of data makes the system not scalable, we have compressed the patch descriptors with the PQ method. We can see in Table 2 the details in terms of memory usage per patch and the compression ratios reached for different values of the c and m parameters. We have achieved a lossy patch representation that reduces its size with respect to the baseline by a factor that ranges from 96 to 2048 times. This means that in a Gb of RAM memory, we can fit between 900 to 18.000 pages.

Table 2
Bytes and compression ratio per patch for each PQ setup.

c sub-quantizers	m sub-vectors				
	8	16	32	64	128
64	6 (1:2048)	12 (1:1024)	24 (1:512)	48 (1:256)	96 (1:128)
128	7 (1:1755)	14 (1:877)	28 (1:438)	56 (1:219)	112 (1:109)
256	8 (1:1536)	16 (1:768)	32 (1:384)	64 (1:192)	128 (1:96)

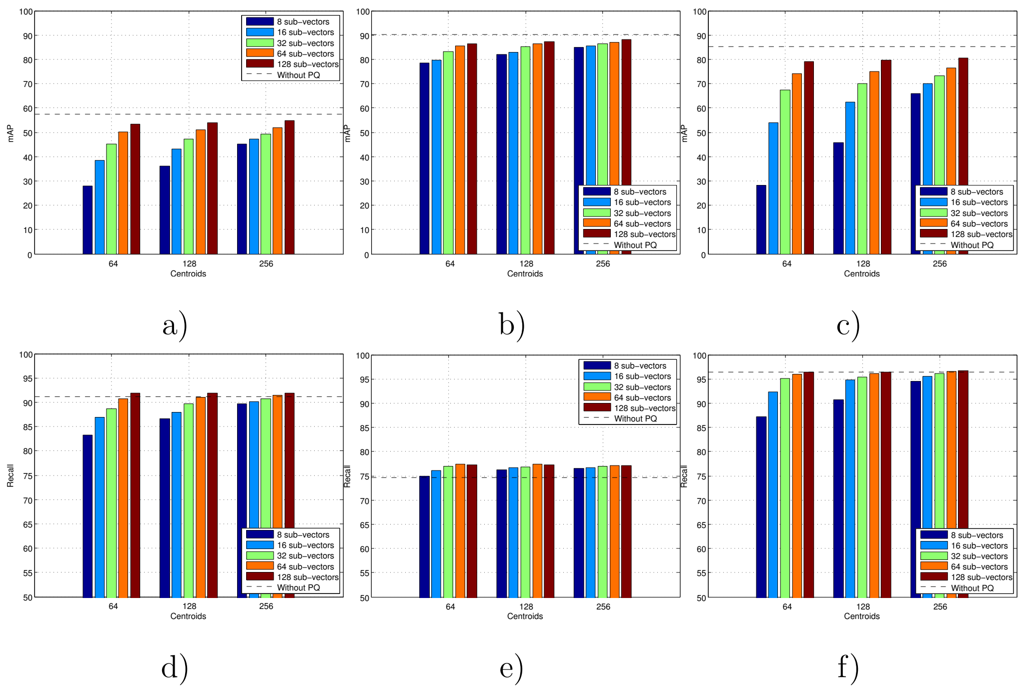


Fig. 6. Mean average precision and recall for different c and m values when using PQ for the a) and d) George Washington, b) and d) Barcelona Cathedral and c) and f) Lord Byron's collections.

In Figure 6 we present the mAP and recall measures obtained after compressing the patch descriptors with different values of the c and m parameters. As we can appreciate, concerning the mAP, the increase of sub-vectors m enhances the performance of the system whereas no significant improvement is observed when increasing the amount of sub-quantizers c. Regarding the recall indicator, we appreciate the same phenomenon, but in that case the recall values even slightly outperform the baseline system.

These results can be attributed to the quantization step when local patch descriptors are converted to PQ codes. This quantization reduces the discriminative power of the local patch descriptors, resulting in a reduction of the mAP. However, it also reduces the effects of noise, leading to the observed moderate increase of the recall in all databases.

Table 3
Performance of the proposed baseline method with PQ, multi-length configuration.

	Collection					
	GW20		BCN		LB	
	mAP(%)	Recall(%)	mAP(%)	Recall(%)	mAP(%)	Recall(%)
Baseline ($K = 2^{15}$,$T = 512$)	57.51	91.22	90.17	74.64	85.16	96.48
With PQ ($c = 256$, $m = 128$)	54.68	91.88	88.07	77.07	80.61	96.71
With PQ and Multi-length	61.35	95.43	88.93	83.21	90.38	97.34

4.2.4 Using Mutli-length Patch Indexation

So far in the presented experiments the size of the query word has not been taken into account. In our previous work [21], we found that short queries performed worse than larger ones when using a fixed size of the patch. Therefore, we have used a multi-length patch representation in order to fix this shortcoming despite the increase in memory requirements. We can see in the third row of Table 3 the important gain in both mAP and recall when the patch indexation is adapted to the query width. By looking at the individual performances attained at each patch geometry, we have observed that only the tiny patch configuration performs slightly worse than the fixed length approach. All other scales outperform the fixed approach. For instance in the GW20 dataset we obtained a 48.94%, 54.55%, 67.03% and 79.17% mAP for the tiny, small, medium and large patch configurations respectively. While the fixed approach reach a 54.68% mAP. A similar phenomenon is observed for the recall value. The increase in performance as the patch geometry grows can be explained due to the perceptual aliasing. Short queries are more likely to obtain false positives since matching to a sub-strings is not penalized in our method. Even though this behavior penalizes the performance of the method, this is not an undesired conduct in the query-by-example setup.

4.2.5 Large scale evaluation

The vectorial representation of our spotting method allows to efficiently index large collection of pages. Unfortunately, publicly available databases only have tens of annotated pages since creating the ground-truth for large collections is a tedious tasks. Therefore, in our large scale experiments using the GW1500 dataset, we had to calculate the retrieval score manually. Query images were generated by randomly selecting 50 word images and the retrieval score was obtained by manually annotating the correct matches for the 50 top-most results. We have used up to 1.500 document images which required up to 60 million patches to represent the whole collection. Therefore, when using the multi-length representation, the whole document collection needed about 7.5 Gb of memory with the best PQ configuration.

Table 4
Performance of the system in the GW1500 scenario.

Num. of pages	mAP (%)	time (ms.)
100	57.73	3.01
500	56.83	2.85
1000	55.98	2.84
1500	55.69	2.87

The mAP and time needed to process a single page for different sized collections is shown in Table 4. As expected, the mAP score of the system slowly decreases as more pages are indexed because of the amount of distractors and false positive visually similar words also increases. Regarding the computational cost, the time required to process a page remains constant as more pages are added. The PQ framework allows to retrieve approximate nearest neighbors sub-linearly so that the candidate search time actually decreases when adding more pages to the collection. However, the more pages we add to the collection, the more the voting scheme from the candidate localization step increases its computational cost. Leading to a nearly constant time as more pages are added since both steps compensate each other. In addition, these results do not take into account that pages can be processed independently. Therefore, an straightforward modification to speed up retrieval speed is to process document pages concurrently.

4.2.6 Comparison with related literature

The George Washington collection has been used in many word spotting works and has become a de-facto dataset used to benchmark different systems. However, the lack of a standard evaluation protocol and the different taxonomies of word spotting methods provokes that achieving a direct comparison among methods is not straightforward. Not all the authors use the same set of pages, query words and even evaluation measures. We present in Table 5 a review of the achieved performances of several state-of-the-art methods. Only Almazan et al. [7] used the same evaluation methodology than us. For the sake of comparison, we have evaluated our method using each of the different experimental setups and evaluation measures proposed by the authors in the original papers. We can see that in equal conditions, the proposed method outperforms all state-of-the-art but the method proposed by Frinken et al. in [4]. It is also worth to mention that some segmentation-based methods present their results by using a manual segmentation of the images avoiding thus the problems derived from any erroneous segmentation artifacts. Lets now discuss the different results obtained using the different configurations.

In some methods we have to take into account the reported recall measure. In Rath and Manmatha [10] and Rothfeder et al. [9], they use a pruning step to remove unlikely correspondences and also to speed up the retrieval process.

Table 5. Review of performances of the state-of-the-art methods using the GW20 dataset.

Method	Reference	Segmentation	Learning	Dataset	Original Results	Proposed method
Semi-continuous HMMs	Rodríguez-Serrano and Perronnin [2]	Words	Yes, 5-fold cross validation: 1 fold train, 1 fold validation, 3 folds test	20 pages, all words in train set as queries	53.1% mAP	**57.9% mAP**
Boosted decision trees	Howe et al. [8]	Words	Yes: 20-fold cross validation: 19 folds train, 1 fold test	20 pages, all words in train set as queries	79.5% mAP	**87.4% mAP**
Dynamic time warping	Rath and Manmatha [10]	Words	No	10 good quality pages, 2381 queries	40.9% mAP	**64.7% mAP**
Corner feature correspondences	Rothfeder et al. [9]	Words	No	10 good quality pages, 2381 queries	36.2% mAP	**64.7% mAP**
Synthesized words	Liang et al. [11]	Words	Yes: 5-fold cross validation: 4 folds train, 1 fold test	20 pages, 38 queries	67% mAP at rank 10	**79.2% mAP at rank 10**
Recurrent Neural Networks	Frinken et al. [4]	Lines	Yes, 4-fold cross validation: 2 folds train, 1 fold validation, 1 fold test	20 pages, all words of the training set appearing in all 4 folds as queries	**71% mean prec.**	67.1% mean prec.
Character HMMs	Fischer et al. [3]	Lines	Yes, 4-fold cross validation: 2 folds train, 1 fold validation, 1 fold test	20 pages, all words of the training set appearing in all 4 folds as queries	62% mean prec.	**67.1% mean prec.**
Slit style HOG features	Terasawa and Tanaka [6]	Lines	No	20 pages, 15 queries	79.1% mAP	**87% mAP**
Gradient features with elastic distance	Leydier et al. [14]	None	No	20 pages, 15 queries	60% P=R	**71.2% P=R**
Exemplar SVM	Almazán et al. [7]	None	No	20 pages, all 4856 words as queries	54.5% mAP	**61.35% mAP**

Likewise, we revoked retrieved images with low score until a similar recall value is attained. This pruning increases the mAP score as we can observe in Table 5.

Other methods use a selected set of queries [11,6,14] avoiding stop words. Since stop words have few characters and are difficult to distinguish visually, by not considering them our method increased its mAP score. Similarly, methods which use cross-validation only cast the query words appearing in all fold sets. The configuration used by Rodriguez-Serrano and Perronnin in [2] divides the database in 5 folds: a fold is used to create the queries, another fold is used for validation purposes and the last 3 folds are used as test collection. Since the GW dietaries explain facts temporally, some words just appearing in specific page ranges are not considered using this configuration. However, stop words are kept as queries, resulting in a mAP score decrease. However, the leave-one-out configuration used in [8], just use a single page as test leading to drastic reduction both in the number of queries and the possible retrieved results facilitating a steep increase of the mAP.

Finally, Fischer et al. [3] and Frinken et al. [4] use cross-validation with an evaluation framework performed at line level, i.e. a whole line is assessed as relevant when it contains a single instance of the query word. In order to compare their results with our method, we followed a procedure similar to the one defined in both papers when comparing to DTW. The retrieved images are first projected to the closest text line. The score of a whole text line corresponds to the highest score of the projected words. By following such evaluation procedure, we obtain 100 line results for a given query, since each page contains about 20 lines and only 5 pages are indexed per fold. Consequently, the results obtained using this evaluation procedure have to be taken cautiously when compared with word-level evaluations. When using this configuration the mAP tends to increase since the effect of false-positives and lowly ranked true-positives is lessened.

Still, Frinken et al. [4] outperforms our method. This is not surprising since their method is learning-based. Besides using statistical machine learning models that cope with the handwritten word variations, these methods also integrate language models to further reduce the effects of visual ambiguities. However, example-based system are more flexible as they can be used to spot any kind of word or symbol present in the document image. For instance, our system was also able to spot the graphical stamp of the Library of the Congress as shown in the last row of Figure 4.

Finally, another important aspect when evaluating spotting systems is the computational cost. From all the methods reviewed in Table 5, just Almazan et al. [7] present an efficient implementation that can be scalable to large environmnents, requiring 15 ms. per indexed page. Approaches based on slidingwindows [4,6] are not scalable, since they have to process the whole document corpus each time that a query is cast. Other holistic word signatures like [10,9] require a complex alignment processes which can not be effectively indexed. By contrast, the vector representation used by our method can naturally handle larger amount of data by efficiently storing indexed information and obtaining results in sub-linear complexity.

5 Conclusions

In this paper we have presented an efficient keyword spotting method for historical collections that does not involve any segmentation stage. Thus the proposed method presents a clear advantage over segmentation-based methods which are likely to fail in challenging scenarios.The proposed method yields a very compact, efficient and discriminative representation thanks to the LSA technique and the PQ compression step. Such representation is able to efficiently index the document information both in terms of memory and computational cost, resulting in a suitable method for large-scale scenarios. By introducing a multi-length patch representation, we have increased the retrieval performance when querying small words. In addition, we have presented a thorough analysis and evaluation of all the involved parameters of the method in order to assess the configuration maximizing the retrieval performance.

Finally, we have presented an exhaustive comparison with state-of-the-art word-spotting methods. We evaluated our method using the experimental setup of each compared method, concluding that our method outperforms all them but Frinken et al. [4]. However, our example-based method is more flexible since it does not rely on any segmentation method nor any image pre-processing step and it does not take advantage of a language model, so that, it is not limited to search words in the document but it can retrieve any kind of symbol present in the images. This is a feature which can be useful for historical documents where symbols used to abbreviate common words and illustrations commonly appear. Nonetheless, adding either a language model or some statistical machine learning steps to the proposed architecture is straightforward. For instance in [30], we added a model of the character distribution over words in a query-by-string framework.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Education and Science under projects TIN2011-25606 (SiMeVé), and TIN2012-37475-

C02-02, by the European project ERC-2010-AdG-20100407-269796, by the People Programme (Marie Curie Actions) of the Seventh Framework Programme of the European Union (FP7/2007-2013) under REA grant agreement no. 600388, and by the Agency of Competitiveness for Companies of the Government of Catalonia, ACCIÓ.

References

[1] R. Plamondon, S. Srihari, On-line and off-line handwriting recognition: A comprehensive survey, IEEE Trans. on Pattern Analysis and Machine Intelligence 22 (1) (2000) 63–84.

[2] J. Rodríguez-Serrano, F. Perronnin, A model-based sequence similarity with application to handwritten word-spotting, IEEE Trans. on Pattern Analysis and Machine Intelligence (2012) 2108–2120doi:10.1109/TPAMI.2012.25.

[3] A. Fischer, A. Keller, V. Frinken, H. Bunke, Lexicon-free handwritten word spotting using character HMMs, Pattern Recognition Letters 33 (7) (2012) 934–942. doi:10.1016/j.patrec.2011.09.009.

[4] V. Frinken, A. Fischer, R. Manmatha, , H. Bunke, A novel word spotting method based on recurrent neural networks, IEEE Trans. on Pattern Analysis and Machine Intelligence 34 (2) (2012) 211–224. doi:10.1109/TPAMI.2011.113.

[5] T. Rath, R. Manmatha, Word spotting for historical documents, International Journal on Document Analysis and Recognition 9 (2–4) (2007) 139–152. doi:10.1007/s10032-006-0027-8.

[6] K. Terasawa, Y. Tanaka, Slit style HOG feature for document image word spotting, in: Proc. of the International Conference on Document Analysis and Recognition, 2009, pp. 116–120. doi:0.1109/ICDAR.2009.118.

[7] J. Almazán, A. Gordo, A. Fornés, E. Valveny, Efficient exemplar word spotting, in: Proc. of the British Machine Vision Conference, 2012, pp. 67.1–67.11. doi:10.5244/C.26.67.

[8] N. Howe, T. Rath, R. Manmatha, Boosted decision trees for word recognition in handwritten document retrieval, in: Proc. of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2005, pp. 377–383. doi:10.1145/1076034.1076099.

[9] J. Rothfeder, S. Feng, T. Rath, Using corner feature correspondences to rank word images by similarity, in: Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshop, Vol. 3, 2003, pp. 30–36. doi:10.1109/CVPRW.2003.10021.

[10] T. Rath, R. Manmatha, Word image matching using dynamic time warping, in: Proc. of the IEEE Computer Society Conference on

Computer Vision and Pattern Recognition, Vol. II, 2003, pp. 521–527. doi:10.1109/CVPR.2003.1211511.

[11] Y. Liang, M. Fairhurst, R. Guest, A synthesised word approach to word retrieval in handwritten documents, Pattern Recognition 45 (12) (2012) 4225–4236. doi:10.1016/j.patcog.2012.05.024.

[12] L. Likforman-Sulem, A. Zahour, B. Taconet, Text line segmentation of historical documents: A survey, International Journal on Document Analysis and Recognition 9 (2–4) (2007) 123–138. doi:10.1007/s10032-006-0023-z.

[13] G. Louloudis, B. Gatos, I. Pratikakis, C. Halatsis, Text line and word segmentation of handwritten documents, Pattern Recognition 42 (12) (2009) 3169–3183. doi:10.1016/j.patcog.2008.12.016.

[14] Y. Leydier, F. Lebourgeois, H. Emptoz, Text search for medieval manuscript images, Pattern Recognition 40 (12) (2007) 3552–3567. doi:10.1016/j.patcog.2007.04.024.

[15] Y. Leydier, A. Ouji, F. LeBourgeois, H. Emptoz, Towards an omnilingual word retrieval system for ancient manuscripts, Pattern Recognition 42 (9) (2009) 2089–2105. doi:10.1016/j.patcog.2009.01.026.

[16] M. Rusiñol, J. Lladós, Word and symbol spotting using spatial organization of local descriptors, in: Proc. of the of the Eighth IAPR Workshop on Document Analysis Systems, 2008, pp. 489–496. doi:10.1109/DAS.2008.24.

[17] P. Roy, J. Ramel, N. Ragot, Word retrieval in historical document using character-primitives, in: Proc. of the International Conference on Document Analysis and Recognition, 2011, pp. 678–682. doi:10.1109/ICDAR.2011.142.

[18] B. Gatos, I. Pratikakis, Segmentation-free word spotting in historical printed documents, in: Proc. of the International Conference on Document Analysis and Recognition, 2009, pp. 271–275. doi:10.1109/ICDAR.2009.236.

[19] L. Rothacker, M. Rusiñol, G. Fink, Bag-of-features hmms for segmentation-free word spotting in handwritten documents, in: Proc. of 12th International Conference on Document Analysis and Recognition, 2013, pp. 1305–1309. doi:10.1109/ICDAR.2013.264.

[20] N. Howe, Part-structured inkball models for one-shot handwritten word spotting, in: Proc. of 12th International Conference on Document Analysis and Recognition, 2013, pp. 582–586. doi:10.1109/ICDAR.2013.121.

[21] M. Rusiñol, D. Aldavert, R. Toledo, J. Lladós, Browsing heterogeneous document collections by a segmentation-free word spotting method, in: Proc. of the International Conference on Document Analysis and Recognition, 2011, pp. 63–67. doi:10.1109/ICDAR.2011.22.

[22] B. Fulkerson, A. Vedaldi, S. Soatto, Localizing objects with smart dictionaries, in: European Conference on Computer Vision, Vol. 5302 of Lecture Notes on Computer Science, 2008, pp. 179–192. doi:10.1007/978-3-540-88682-2_15.

[23] E. Nowak, F. Jurie, B. Triggs, Sampling strategies for bag-of-features image classification, in: European Conference on Computer Vision, Vol. 3954 of Lecture Notes in Computer Science (LNCS), 2006, pp. 490–503. doi:10.1007/11744085_38.

[24] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in: Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006, pp. 2169–2178. doi:10.1109/CVPR.2006.68.

[25] G. Salton, C. Buckley, Term-weighting approaches in automatic text retrieval, Information Processing & Management 24 (5) (1988) 513–523. doi:10.1016/0306-4573(88)90021-0.

[26] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, R. Harshman, Indexing by latent semantic analysis, Journal of the American Society for Information Science and Technology 41 (6) (1990) 391–407. doi:10.1002/(SICI)1097-4571(199009)41:6¡391::AID-ASI1¿3.0.CO;2-9.

[27] A. Levey, M. Lindenbaum, Sequential karhunen-loeve basis extraction and its application to images, IEEE Trans. on Image Processing 9 (8) (2000) 1371–1374. doi:10.1109/83.855432.

[28] H. Jégou, M. Douze, C. Schmid, Product quantization for nearest neighbor search, IEEE Trans. on Pattern Analysis and Machine Intelligence 33 (1) (2011) 117–128. doi:10.1109/TPAMI.2010.57.

[29] J. Almazán, D. Fernández, A. Fornés, J. Lladós, E. Valveny, A coarse-to-fine approach for handwritten word spotting in large scale historical documents collection, in: Proc. of the International Conference on Frontiers in Handwriting Recognition, 2012, pp. 455–460. doi:10.1109/ICFHR.2012.151.

[30] D. Aldavert, M. Rusiñol, R. Toledo, J. Lladós, Integrating visual and textual cues for query-by-string word spotting, in: Proc. of 12th International Conference on Document Analysis and Recognition, 2013, pp. 511–515. doi:10.1109/ICDAR.2013.108.