A sliding window framework for word spotting based on word attributes

Suman K. Ghosh and Ernest Valveny

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

Abstract. In this paper we propose a segmentation-free approach to word spotting. Word images are first encoded into feature vectors using Fisher Vector. Then, these feature vectors are used together with pyramidal histogram of characters labels (PHOC) to learn SVM-based attribute models. Documents are represented by these PHOC based word attributes. To efficiently compute the word attributes over a sliding window, we propose to use an integral image representation of the document using a simplified version of the attribute model. Finally we re-rank the top word candidates using the more discriminative full version of the word attributes. We show state-of-the-art results for segmentation-free query-by-example word spotting in single-writer and multi-writer standard datasets.

Keywords: Word Spotting, Sliding window, Word attributes

1 Introduction

Due to recent development of image databases of handwritten and historic manuscripts, the demand for algorithms to make these databases accessible for browsing and indexing are in rise. The state of the art OCR technologies are not directly applicable to these type of documents due to challenges like the diversity of the handwriting style, the presence of noise and distortions in historical manuscripts, etc.

To overcome this one can perform an image based search in the form of query by example. The goal of query by example word spotting can be defined as identifying and retrieving all those regions in a dataset of document images that contain an instance of a query word image. In a multi-writer collections, where handwriting can differ significantly from document to document, this task can be quite challenging. In the literature, word spotting appears under two distinct trends wherein the fundamental difference concerns the search space which could be either a set of segmented word images (segmentation-based approaches) or the complete document image (segmentation-free approaches). In this work, we address the query by example word spotting problem in a segmentation-free multi-writer scenario.

Initial works in word spotting followed a traditional path of OCR technologies, starting with a binarization followed by layout analysis to perform a word

level segmentation. Popular matching techniques like Hidden Markov Model (HMM) and Dynamic Time Warping (DTW) were used to match query words with these extracted word candidates by representing both query and word candidates as sequence of features. Example of this type of framework are the works of [1–3]. The main drawbacks of these methods come from the dependence on the segmentation step, which can be very sensible to handwriting distortions, and the computational cost of the sequence-based comparison.

More recently, word spotting methods which do not use a precise segmentation step have been reported [4–8]. Some of these methods [4–6] are based on the extraction of local keypoints that are encoded using descriptors based on gradient information [4, 5] or Heat Kernel Signature [6]. Word spotting is then performed by locating zones of the document images with similar interest points and, in some cases with the same spatial configuration as the query model [4]. In general, they use a costly distance computation, which is not scalable to large datasets. The work of Rusiñol et al.[7] avoids segmentation by representing regions with a fixed-length descriptor based on the well-known bag of visual words (BoW) framework [9] . In this case, comparison of regions is much faster since a dot-product or Euclidean distance can be used, making a sliding window over the whole image feasible. In addition, Latent Semantic Indexing (LSI) is used to learn a latent space where the distance between word representations is more meaningful than in the original space. Rothacker *et al.* [8] also makes use of the BoW representation to feed a HMM obtaining a robust representation of the query and avoiding segmentation using a patch-based framework. Comparison of regions is slower than in the BoW-based approach of [7], so it could not be directly applied in a large-scale scenario. In [10] Almazán *et al.* proposed to use a HOG based framework in combination with an exemplar-SVM framework to learn a better representation of the query from a single example. Compression of the descriptors by means of product quantization permits a very efficient computation over a large dataset in combination with a sliding window-based search. In [10] the authors also proposed to use a reranking step to further improve the accuracy using a more costly Fisher Vector based representation over the top results retrieved using HOG descriptors.

Though all these methods perform well in the case of single writer documents, the representations used are not capable of handling the variation imposed by documents written by multiple writers. More powerful representation and learning techniques are needed to deal with this problem. In this sense, in [11], Almazán *et al.* used a fixed length attribute representation which gives an efficient way of performing word spotting in both QBE (Query By Example) and QBS (Query By String) scenarios using the same framework. They achieved good results in a segmentation-based framework in both single and multi-writer datasets. The attribute representation encodes the spatial position of characters in the word image through a Pyramidal Histogram of Characters (PHOC) and is learned using the more powerful Fisher Vector representation of the images. Once word images are represented in this attribute space spotting is reduced to a Nearest Neighbour problem. Though this framework has achieved high accuracy

in case of segmented words it can not be applied directly in a segmentation-free approach as it involves computation of costly Fisher Vector representation, which is unfeasible at query time.

In this work we propose to use a similar representation over a sliding window protocol for segmentation-free word spotting. As the computation of such a costly representation at query time is not feasible, we propose to pre-compute an integral image representation of the attributes. However, it is not possible to encode exactly the same original attribute representation in an integral image. Some simplifications have to be done which makes the attribute representation a bit less discriminative. To overcome this we propose an additional re-ranking step at the end of the pipeline which uses the same attribute representation as of [11] for final ranking of the top candidate windows. Our main contributions can be summarized as: i) We propose an efficient computation of the attribute word representation over a whole document using an integral image ii) We combine an initial ranking based on a sliding window search with a re-ranking step on the top candidate windows using a more powerful attribute representation iii) With this combined approach we are able to perform segmentation-free query by example in the challenging multi-writer scenario, where we are not aware of any previous reported results.

The rest of the paper is organised as follows: in section 2 we briefly describe the computation of the attribute model and its extension in the sliding window protocol followed by the explanation of re-ranking step. In section 3, we discuss about the various experiments carried out to compare our method with other state of the art methods. Finally we conclude the paper with possible extensions and improvements.

2 Method Description

The approach proposed is illustrated in Figure 1, the query and document images are first converted to its PHOC representation. Then, the retrieval step is simplified to a nearest neighbour problem, computing cosine distance of the query image to all of the candidates given by the sliding window and ranking them in order of similarity. Finally, we compute the more discriminative attributes for the top $N\%$ candidates and re-rank these to give final ranked list as result. In the following subsections we first give a summary of the attribute word representation proposed by Almazán *et al.* in [11], next we describe how it can be adapted to compute the integral image. Finally we explain the combination of both representations to obtain the final spotting pipeline.

2.1 Attribute-based word representation

The main idea of the approach proposed by Almazán *et al.* [11] is to learn a common low dimensional representation for word images and text strings, that permits to address retrieval as a simple nearest neighbor problem. Though this

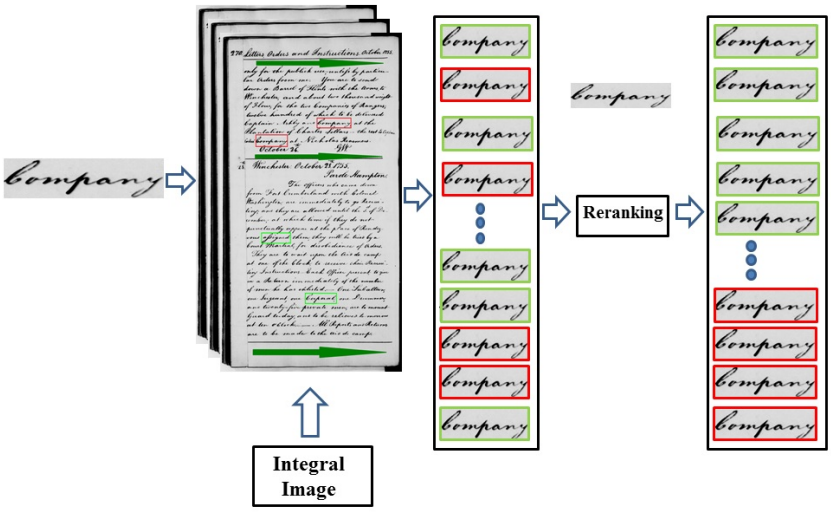

Fig. 1. General overview of the proposed pipeline

representation can be utilized to accomplish both QBE and QBS, here our focus is on QBE word spotting.

To learn the attribute representation first, text strings are embedded into a $d-$dimensional binary space, in a way similar to the bag of characters string kernels [14, 15]. This embedding – called Pyramidal Histogram Of Characters (PHOC) – encodes if a particular character appears in a particular spatial region of the string. The basic representation is just a binary histogram of characters, encoding which characters appear in the string. In order to add more discriminative power new levels are added to this histogram in a pyramidal way. At each level of the pyramid the word is further split and a new histogram of characters is added for each new division to account for characters at different parts of the word. At the end, 5 levels are used leading to a word representation of 604 dimensions.

Then, this embedding is used as a source for learning character attributes from word images. Each word image is projected into a $d-$dimensional space (same dimension as the PHOC representation) where each dimension is an attribute encoding the probability of appearance of a given character in a particular region of the image, using the same pyramidal decomposition as in the PHOC representation. Each attribute is independently learned using an SVM classifier on a Fisher Vector description of the word image, enriched with the x and y coordinates and the scale of the SIFT descriptor.

More formally, given a training image I, we can compute its Fisher Vector representation [18] $f(I)$, where $f(I)$ is a function of the form $f : I \rightarrow R^D$, being D the dimension of the Fisher Vector representation. Now, to project Fisher Vector representations into the PHOC attribute space, we learn an embedding function ϕ_I of the form $\phi_I : I \rightarrow R^d$ such that

$$\phi_I(I) = \mathbb{W}^{\mathbb{T}} f(I) \tag{1}$$

where W is a matrix with an SVM-based classifier for each attribute learned using the PHOC labels of all the training words.

In a query by example setting both the query and word images are described with this attribute representation, which is very discriminative as each attribute is giving the probability of a certain character in a specific position within the word. Retrieval simply translates into finding the word candidates whose attribute representation is close to that of the query image.

To make direct comparison between binary PHOCs and real valued attribute representations feasible Almazán *et al.* in [11], proposed an additional step to learn a common subspace between strings and images. A final calibration step is added, using Canonical Correlation Analysis, that aims at maximizing the correlation among both representations. In our case, although we are not concerned about comparing the PHOCs from text strings with attributes from images(typical for recognition tasks), we still use this low dimensional subspace as it provides an elegant way to reduce the dimensionality while not affecting the discriminativeness of the representations.

This final calibration and dimensionality reduction step can be represented with an additional embedding function ψ represented as $\psi_I : I \rightarrow R^{d'}$ and can be given as:

$$\psi_I(I) = U^T \phi_I(I) \tag{2}$$

being U the transformation matrix obtained with Canonical Correlation Analysis.

2.2 Representation of Word Attributes and Ranking

The main bottleneck of using word attributes as basic representation over a sliding window protocol is that it involves the costly computation of SIFT descriptors and Fisher Vector representation at run time – it takes around 110ms for a single candidate window –. Moreover, note that to compute attributes for every window given by a sliding window protocol, one have to compute SIFT descriptors and the Fisher Vector for a large number of overlapping windows redundantly over the same image. To alleviate these problems we propose to pre-compute off-line the attribute representation for every pixel of the image and store it in an efficient integral image [19] that can be used to compute very fast the representation of any candidate window at query time.

To describe the computation of the integral image of the attribute representation, let us denote the document images of the dataset as $I^k, k = 1...n$ where n is the total number of images. For a given image I^k, we first compute the set of dense SIFT descriptors $d^k_{i,j}$ at every location (i, j). Then, we can define the embedding function into the attribute space ϕ_I for every pixel location as:

$$\phi_I(i, j) = \mathbb{W}^{\mathbb{T}} f(d_{i,j}) \tag{3}$$

where $f(d_{i,j})$ is the Fisher Vector representation for the (i, j) pixel of image I^k and W is a matrix encoding the attribute classifiers as in previous section. Finally this attribute representation for every pixel is projected to the lower dimensional subspace obtained through Canonical Correlation Analysis using the same transformation matrix U introduced in previous section:

$$\psi_I(i,j) = \mathbb{U}^{\mathbb{T}}\phi_I(i,j) \tag{4}$$

Once we have the final attribute representation for every pixel, it can be easily aggregated into an integral image $\Psi_{i,j}$:

$$\Psi_{i,j} = \sum_{i'<=i,j'<=j} \psi_{i,j} \tag{5}$$

The time and memory requirements for computing the attribute representation representation can be further reduced if we arrange the image into $N \times N$ dimensional blocks and instead of computing Fisher Vector representation for every pixel, we only compute one Fisher Vector for each block.

Finally, given a query image and the integral image representation we have to generate candidate windows using a sliding window and rank the list of candidate windows according to the similarity with the query image. We first compute the attribute representation for the query and compute the attribute representation for all the candidate windows. Given a window $w = (X_1, Y_1, X_2, Y_2)$, where (X_1, Y_1) are the co-ordinates of the top left corner and (X_2, Y_2) are the co-ordinates of the bottom right corner of the w, we can compute the attribute representation Ψ_w in a very simple way with just 4 vector additions as:

$$\Psi_w = \Psi(X_2, Y_2) + \Psi(X_1, Y_1) - \Psi(X_1, Y_2) - \Psi(X_2, Y_1) \tag{6}$$

Now to compute the similarity between the query and the candidates we use cosine similarity by taking dot products.

2.3 Reranking

The integral image of attributes while being fast can not exploit the full discriminative power of the original attribute representation due to certain simplifications that are required to be able to compute the integral image offline before the query time. In particular: i) as we are computing Fisher Vector on a per pixel basis, we can not have, at the time of computing the integral image, the relative position of the key-points inside a given candidate box. Therefore, SIFT descriptors cannot be enriched using the relative positional information x, y coordinates, as explained in section 2.1. ii) Also, as we cannot know the size of the underlying window,we can not apply the window size normalization performed in the original approach.

These limitations result in a significant loss of accuracy that can be partially alleviated by introducing a re-ranking step, as it is usual in other applications of image retrieval [16, 17]. Basically it consists of applying more discriminative

and costly features to the best windows retrieved by the first ranking step in order to obtain the final ranking list. In the context of word spotting example of re-ranking can be seen in [10],where they re-rank top windows by Fisher Vector after selecting them using a HOG based representation.

In this work we use the same strategy: the top $N\%$ candidates from the ranked list given by the initial ranking obtained with the sliding window search are re-ranked using the more discriminative original attribute representation described in section 2.1.

3 Experimental Results

To evaluate our method and compare with other state of the art methods we use three different datasets. First, we briefly describe each of these datasets before moving to the results section.

3.1 Datasets

The George Washington (GW) dataset [12] contains 5000 words annotated at word level. The dataset comprises 20 handwritten letters written by George Washington and his associates in 18th century. The writing styles present only small variations and it can be considered a single-writer dataset.

The Lord Byron (LB) dataset similar to the GW dataset it also contains approximately 5000 words spread over 20 pages annotated at word level. However the nature of the data is completely different as it consists of typewritten text.

To evaluate our method in a multi-writer setting we used the **IAM Offline Dataset**[13]. It is a large dataset comprised of 1539 pages of modern handwritten English text written by 657 different writers. The document images are annotated at word and line level and contain the transcriptions of more than 13000 lines and 115000 words. We follow the official partition for writer independent text line recognition task.

3.2 Results

To evaluate and compare to state-of-the-art approaches, we follow standard protocols as in [10, 7] in case of GW and LB datasets. Every word in the dataset is considered as a query and after ranking, a candidate window is considered as a true positive if it overlaps by more then 50% with any of the ground truth annotated boxes. We measure accuracy in terms of Mean Average Precision. In case of IAM dataset, however, we follow a different strategy as in line spotting instead of word spotting, *i.e.* the whole lines are retrieved if they contain the query word. Each query word is searched inside all annotated text lines using sliding window approach. The distance between query and text line is defined as the distance between query and the closest candidate word of that line. A similar strategy has been followed by Almazán *et al.* in [11].

Table 1 summarizes the results of our method, compared to other state-of-the-art methods. We provide results for three settings of our method: the first one without reranking, and then reranking with two different choices of N. From the results it can be observed that our base line system without any reranking can give better results than [10] without reranking in the GW dataset and only a bit lower in the LB dataset. Using reranking we obtain the best results among all systems in the GW dataset and very close to the best system in the LB dataset. For the IAM dataset none of the existing approaches reporting results for word spotting can be directly compared to our approach as they either work in a segmentation based framework [11] or in Query By String [4, 21]. Up to our knowledge our results are the first ones reported for IAM dataset in a segmentation-free query by example framework. Results, specially with the reranking step, compare pretty well with the 52.61 MAP reported in [11] in the much easier task of segmentation-based word spotting.

Table 1. Result of our word spotting method in comparison with state-of-the-art. (1) Proposed method without the reranking step. (2) Re-ranking with top 80% of candidates. (3) Re-ranking with top 60% of the candidates from first step.

	GW	LB	IAM
Almazán *et al.* [10]	51.88	84.34	-
Almazán *et al.* [10] (with RR)	57.46	84.51	-
Russiñol *et al.* [7]	30.42	42.83	-
Kovalchuk *et al.* [20]	50.1	90.7	-
Proposed (1)	56.27	84.45	35.68
Proposed (2) with RR (80%)	67.7	90.45	42.08
Proposed (3) with RR (60%)	63.87	87.85	39.94

We also show the average computational time to evaluate each query in Table 2. In this table we kept only one variant of re-ranking with top 80% of the candidates. It can be observed that the proposed method without the re-ranking step is quite fast to be used in a real time environment. In comparison with Almazán *et al.* [10] it is marginally slow while achieving a higher accuracy. The re-ranking step significantly increases the computational time as it must compute SIFT and Fisher Vector for each candidate window.

4 Conclusion

This paper proposes a segmentation-free approach to word spotting in document images. We have shown an efficient way to represent PHOC based word attributes in an integral image format, which can be computed offline and used efficiently in query time. The results of our method shows significant improvements over the current state-of-the-art. In addition we are able to apply our

Table 2. Result with respect to the computation time in per query basis

	GW	LB
Almazán *et al.* [10]	1.04s	0.83s
Kovalchuk *et al.* [20]	0.033	0.009
Proposed	3.45s	2.87s
Proposed with RR	15.6s	11.7s

method to the multi-write IAM dataset where we are not aware of other published results in the context of segmentation-free word spotting. Computational time could be further improved integrating our approach with a compression technique such as product quantization as done in [10].

The proposed method is based on a simplification of the original attribute word representation. Context information around a pixel can be exploited in the future in order to compensate for the poorer Fisher Vector representation that we use in our method.

References

1. U.V. Marti, H. Bunke, Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition systems, IJPRAI (2001), 65-90.
2. A. Vinciarelli, S. Bengio, H. Bunke, Offline recognition of unconstrained handwritten texts using HMMs and statistical language models, IEEE Transactions on PAMI, 26(2004), 709-720.
3. J. Rodríguez-Serrano, F. Perronnin, Local gradient histogram features for word spotting in unconstrained hand-written documents, International Conference on Frontiers in Handwriting Recognition, 2008.
4. V. Frinken, A. Fischer, R. Manmatha, H. Bunke, A novel word spotting method based on recurrent neural networks, IEEE Transactions on PAMI, 34(2012), 211-224.
5. Y. Leydier, A. Ouji, F. Lebourgeois, H. Emptoz, Towards an omnilingual word retrieval system for ancient manuscripts, Pattern Recognition 42(2009), 2089-2105.
6. X. Zhang, C. L. Tan, Segmentation-free Keyword Spotting for Handwritten Documents based on Heat Kernel Signature, International Conference on Document Analysis and Recognition, 2013, 827-831.
7. M. Rusiñol, D. Aldavert, R. Toledo, J. Lladós, Browsing heterogeneous document collections by a segmentation-free word spotting method, International Conference on Document Analysis and Recognition, 2011, 63-67.
8. L. Rothacker, M. Rusiñol, G. A. Fink, Bag-of-Features HMMs for Segmentation-free Word Spotting in Handwritten Documents, International Conference on Document Analysis and Recognition, 2013, 1305-1309.
9. G. Csurka, C. R. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: Workshop on Statistical Learning in Computer Vision, European Conference on Computer Vision, 2004, 1-22.

10. J. Almazán, A. Gordo and A. Fornés and E. Valveny, Segmentation-free Word Spotting with Exemplar SVMs, Pattern Recognition, 2014.
11. J. Almazán, A. Gordo and A. Fornés and E. Valveny, Word Spotting and Recognition with Embedded Attributes, IEEE Transactions on PAMI (2014).
12. T. Rath and R. Manmatha, Word spotting for historical documents, IJDAR (2007).
13. U.V. Marti and H. Bunke, The IAM-database: An english sentence database for off-line handwriting recognition, IJDAR (2002).
14. C. Leslie, E. Eskin, and W. Noble, The spectrum kernel: A string kernel for SVM protein classification, in Pacific Symposium on Biocomputing, 2002.
15. H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, Text classification using string kernels, JMLR (2002)
16. O. Chum, J. Philbin, J. Sivic, M. Isard, A. Zisserman, Total recall: Automatic query expansion with a generative feature model for object retrieval, International Conference on Computer Vision, 2007, 1-8.
17. R. Arandjelović, A. Zisserman, Three things everyone should know to improve object retrieval, IEEE Conference on Computer Vision and Pattern Recognition, 2012, 2911-2918.
18. F. Perronnin, J. Sánchez, and T. Mensink, Improving the Fisher kernel for large-scale image classification,in European Conference on Computer Vision, 2010.
19. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. in IEEE Conference on Computer Vision and Pattern Recognition, 2005.
20. Alon Kovalchuk, Lior Wolf, and Nachum Dershowitz A Simple and Fast Word Spotting Method, International Conference on Frontiers in Handwriting Recognition, 2014
21. A. Fischer, A. Keller, V. Frinken, and H. Bunke, HMM-based word spotting in handwritten documents using subword models, in International Conference on Pattern Recognition, 2010