Bag-of-Features HMMs for segmentation-free word spotting in handwritten documents

Leonard Rothacker
Department of Computer Science
TU Dortmund University
44221, Dortmund, Germany
Email: leonard.rothacker@udo.edu

Marçal Rusiñol
Computer Vision Center,
Department of Computer Science
Edifici O, Universitat Autònoma de Barcelona
08193, Bellaterra, Spain
Email: marcal@cvc.uab.es

Gernot A. Fink
Department of Computer Science
TU Dortmund University
44221, Dortmund, Germany
Email: gernot.fink@udo.edu

***Abstract*—Recent HMM-based approaches to handwritten word spotting require large amounts of learning samples and mostly rely on a prior segmentation of the document. We propose to use Bag-of-Features HMMs in a patch-based segmentation-free framework that are estimated by a single sample. Bag-of-Features HMMs use statistics of local image feature representatives. Therefore they can be considered as a variant of discrete HMMs allowing to model the observation of a number of features at a point in time. The discrete nature enables us to estimate a query model with only a single example of the query provided by the user. This makes our method very flexible with respect to the availability of training data. Furthermore, we are able to outperform state-of-the-art results on the George Washington dataset.**

I. Introduction

Word spotting methods aim at providing efficient access to textual information without the need to exhaustively process and recognize the whole document contents. They are especially suitable for handwritten documents, for which text recognizers do not yet achieve satisfactory transcription results. Huge amounts of cultural information are still locked in image libraries, because they cannot be efficiently exploited. This has motivated the document analysis community to deal with the handwritten word spotting problem for more than fifteen years.

Inspired by the handwriting recognition field, many handwritten word spotting approaches take advantage of the sequential nature of handwritten text. They model word images as a sequence of column-wise extracted features that are finally matched by a sequence alignment technique. The most widely used features in the literature are word profiles [1] or features combining global and local information. Very prominent representations have been proposed by Marti and Bunke [2] and Vinciarelli et al. [3]. Regarding the feature alignment, techniques such as dynamic time warping [1], Hidden Markov models (HMM) [4], [5] or neural networks [6] have been used for spotting purposes.

However, such techniques rely on preprocessing methods. In the layout analysis step the document is usually segmented at word or line level. In the normalization step unwanted variabilities are removed. The major problem is that failures in preprocessing regularly lead to failures in the recognition. Especially, accurate word segmentations are difficult to obtain in handwritten and degraded documents. For that reason, several segmentation-free word spotting techniques have emerged. The most common approach [7], [8], [9] is to use a patch-based framework in which a window slides over the whole document. In such a framework perfect segmentations are not expected and elements from surrounding words will appear within a patch. With respect to the normalization the most common methods [2], [5] include binarization, skew and slant correction, baseline height and word width normalizations. Representations, like profile-based features, tend to be very sensitive with respect to the successful application of these preprocessing methods. Artifacts from the segmentation introduce further variabilities. Therefore, they are hardly applicable in the scenario considered here. In previous publications like [10], [8], [11] it was proposed to use features from the object recognition field as a basis for representing both handwritten and typewritten words. For instance, the work of Rodríguez-Serrano and Perronnin [10] was one of the first contributions that proposed to use local gradient histogram features (SIFT-like, cf. [12]) for word spotting. The results obtained clearly outperformed profile-based features. A major advantage of using these gradient statistics is that they usually do not require extensive preprocessing. In [8] neither normalization nor segmentation techniques have been applied.

In this paper we propose to use a Bag-of-Features representation powered with SIFT descriptors [12] to feed an HMM as previously proposed in [11]. The Bag-of-Features HMM is a generative finite state machine. Its distinctive characteristic is its output modeling. At each point in time it generates a Bag-of-Features representation. In comparison to a discrete HMM this can be interpreted as generating multiple instead of only a single symbol with respective probabilities. These HMMs are used in a patch-based framework in order to bypass the segmentation step. The descriptors are directly computed on the gray-scale document image, omitting any prior normalization. Since HMMs encode the sequential information of the extracted features as well, we obtain better results than just applying a Bag-of-Features model as in [8].

Bag-of-Features and HMMs have not been used in an integrated methodology for word spotting before. In our approach we are able to combine the advantages and avoid the disadvantages of both techniques. In [8] it was shown that Bag-of-Features representations can successfully be applied in a segmentation-free word spotting scenario. As for visual recognition in general (cf. [13]), these representations tend to be very robust with respect to different variabilities. This is a

consequence of the properties of the SIFT descriptor (cf. [12]) and the clustering step necessary to obtain the *features* that a Bag-of-*Features* model is built upon. Furthermore, they do not encode any specific spatial information what makes them also robust against larger transformations. However, it has been shown that for visual recognition tasks the complete lack of spatial information is not helpful. Capturing loose spatial context is improving the results significantly (cf. [14], [8]). For the majority of word spotting methods, on the other hand, it is a standard approach to model spatial information by sequences of feature vectors. As described before, these features are usually very sensitive with respect to segmentation errors and writing styles. Often this requires complex models with a high number of free parameters that have to be estimated from training samples. Given the sequential structure and high variability of handwritten script we propose a method that deals with both aspects by combining a statistical sequence model with a statistical model of unordered local features. Our HMMs are initialized and trained with standard algorithms (Baum-Welch) although we only use a single sample. Therefore, our method will profit immediately in scenarios where more training material is available.

In summary, the main contribution of this paper is the proposal of a segmentation-free word spotting method that integrates the Bag-of-Features principle with a statistical sequence model. In addition, the statistical model is estimated only from the query itself. In that sense the method is completely unsupervised as it does not need any prior word transcription.

II. Bag-of-Features HMMs for segmentation-free word spotting

In a query-by-example word spotting scenario we have to query a document collection with a sample word image. Our proposed method can be divided into three parts. First, the document images must be represented by local feature representatives (Section II-A). In the second step a query model is created by estimating a Bag-of-Features HMM. Only the provided sample is used for that purpose. The model is initialized and trained on the sequence of Bag-of-Features representations obtained from the bounding box around the query word image (Section II-B). In the last step the model is decoded on the complete document collection. In order to bypass the segmentation step, patches are densely sampled from all pages. For each patch the maximum probability of generating the respective Bag-of-Features sequence is computed (Section II-C). For the final word retrieval, we order the patches with respect to their scores and evaluate the method's performance. Figure 1 shows an overview of the depicted process by visualizing the respective steps on a small section from a document of the George Washington dataset (cf. [1]).

A. Document image representation

A Bag-of-Features representation is a histogram of features that are representative for the problem domain. The first step for obtaining those features is to densely extract SIFT descriptors on each page of the document collection. The descriptors are located on a regular grid of 5×5 pixels and have a uniform scale of 40×40 pixels. Please note that no scale space is involved. Given the nature of handwritten script, it is also important to set the descriptor orientation to zero (cf. [8], [11]), since we want to discriminate between horizontal and vertical pen strokes. Figure 1 (1, 2) shows a section of a document and the dense grid of SIFT descriptors. In the next step, we can estimate features that are representative for the document corpus. Therefore, the Generalized Lloyd [15] clustering algorithm is applied on randomly sampled 5% of all the extracted descriptors in order to build the codebook. Afterwards, we quantize each descriptor in the document collection with respect to the most similar feature representative. In our experiments we choose a codebook size of 4096. Figure 1 (3) exemplarily visualizes the quantization result. Each dot corresponds to a descriptor location. Their colors indicate the respective representatives from the codebook. Please note the similar color patterns in visually similar parts of the document. The choice of the parameters is derived from prior experiments based on [8].

B. Model estimation

A query is defined by a bounding box around a word in the document image. In Figure 1 (4) this is exemplarily shown for the word *the*. The objective is now to estimate a Bag-of-Features HMM that encodes its sequential visual appearance. The description of the visual appearance is already given by the features that are located within the query bounding box. In order to reduce noise caused by the adjacent upper and lower text lines, only features not extending farther than the upper and lower boundaries of the query are used. On the other hand, features overlapping with the left and right query boundaries are kept. This way the beginning and ending of the word can be modeled. The sequential visual appearance is then created by sliding a window over the feature grid. For each position, a Bag-of-Features histogram is computed from the features within the window, in the following referred to as term vector. Additionally, the term vector is normalized to unit length. This is important for the integration with a probabilistic model.

For estimating the HMM, only the query's term vector sequence is used. We use a linear topology which only allows transitions to the same and the successive state. The first step in the initialization process determines the number of states with respect to the length of the sequence. Then, the term vectors are linearly mapped to the states. Transition and output probabilities are defined for each state i. Let N_i be the number of associated term vectors. The probabilities for self-transition and transition to the successive state j are given by

$$a_{ii} = \frac{N_i - 1}{N_i}, \qquad a_{ij} = \frac{1}{N_i} \tag{1}$$

Thus, a probabilistic process depending on N_i activations of state i is described. Finally, for N_i associated term vectors $\boldsymbol{f}_n$ the output probability vector $\boldsymbol{c}_i$ for feature representatives in state i is given by

$$\boldsymbol{c}_i = \frac{1}{N_i} \sum_{n=0}^{N_i - 1} \boldsymbol{f}_n \tag{2}$$

The output coefficients $\boldsymbol{c}_i$ are, therefore, obtained by averaging the observed term vectors. After initializing the model, it can be refined with Baum-Welch training [16]. In this process the model's probability of generating the query's term vector sequence is optimized by an iterative maximum-likelihood

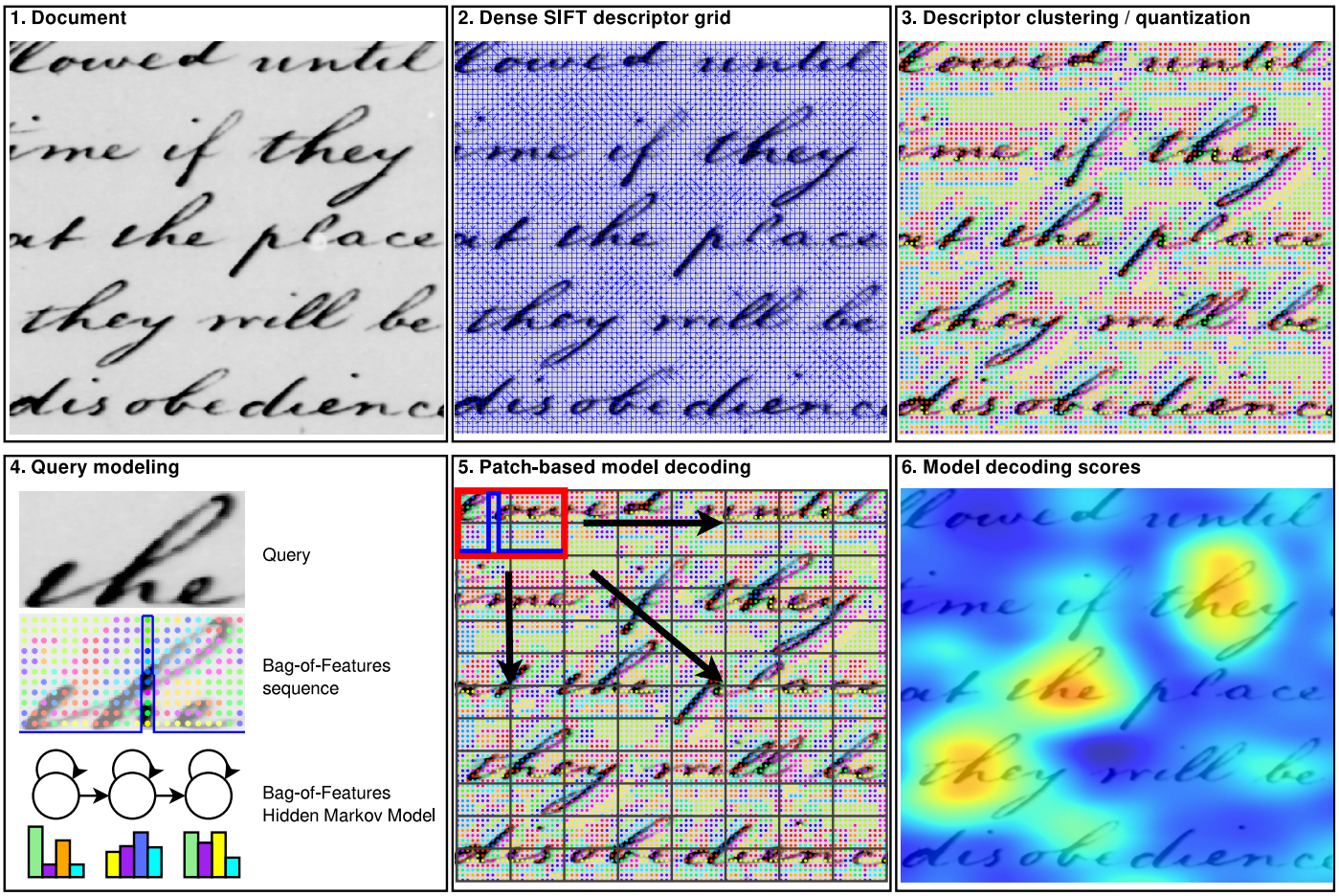


Figure 1. Overview of the proposed segmentation-free word spotting method. In the upper part of the Figure the document image representation is visualized. In the lower part the estimation of a query model and the patch-based decoding with respect to that model are shown. Patch-based representations are evaluated at each grid point. The scores obtained are visualized by interpolating them over the document image indicating low to high responses with blue to red colors. Best viewed in color. Based on image from the George Washington dataset [1].

procedure. Figure 1 (4) visualizes the estimated model exemplarily. The color distributions indicate the feature probabilities in each state. Please note that the described estimation process can directly be adapted to cases where multiple training samples are available. The number of states is then calculated with respect to the shortest observed term vector sequence. After initially associating each sample's term vectors with the respective states, the transition and output probabilities are computed analogously.

In the model estimation a number of parameters have to be defined. The sliding window width is set to 5 pixels and the window shift to 2 pixels based on the feature grid resolution and prior experiments from [11]. The effect of pruning boundary features in the query bounding box, the number of HMM states and the number of Baum-Welch training iterations are thoroughly studied in Section III.

C. Model decoding

After the model is estimated, the document collection can be queried in a patch-based approach. From each page in the document collection patches of the same size as the query are sampled. For efficiency reasons the patch shift is 50% of the patch size. Figure 1 (5) illustrates this for the example document section. In order to decode the query model with the Viterbi algorithm, a term vector sequence is obtained from each patch. As these vector sequences have always the same length, scores can directly be compared with each other when computing the optimal path probability. For further detail on how to obtain the output probability given an observed term vector, refer to [11]. An interpolated map of probabilistic scores is shown in Figure 1 (6). Given the probabilistic scores, the query results can be retrieved. For that purpose overlapping patches are eliminated by non-maximum-suppression. Finally, the remaining top 200 patches per page are returned.

III. Evaluation

As in [8] and [9] we evaluate our method on the George Washington dataset [1]. The document collection contains 20 handwritten pages with available annotations for 4860 words. All these words are used as queries without any further ground truth modifications, like stemming or stopword filtering. Our results are reported in terms of mean average precision (mAP) and mean recall (mR). For a single query we, therefore, order all retrieved patches from all pages with respect to their Viterbi scores. A patch is considered relevant if it overlaps

with a positive patch from the ground truth by more than 20%. The overlap is measured as fraction of the intersecting area and the enclosing area spanned by the two patches. The considered overlap threshold is the only difference in the evaluation protocol compared to [8], [9]. This is due to the larger patch shifts for larger query words (cf. Section II-C). For a comparable threshold (of 50%) the ground truth is often not hit as precisely as necessary. As this can be seen as a limitation on the one hand, it emphasizes the robustness of our method on the other hand. Achieving high mean average precision scores is more difficult if relevant patches do not overlap as much with the actual word in the document image. Please note that as in [8], [9] the query itself might be among the retrieved patches. No filtering as e.g. in [1] is applied.

In our experiments we focus on the evaluation of parameters for modeling queries with HMMs. As discussed in Section II-B, features have to be selected in the bounding box of the query word image. The term vector sequence obtained with a sliding window is then modeled by a number of HMM states. Finally, the model is optimized with Baum-Welch training. In the evaluation only one parameter is varied at once. In Tables I, II and III the respective best observed reference scores and their parameters are marked in bold. In the following we will discuss the effect of different parameters on the mean average precision. Effects on the mean recall are not as evident and behave consistently over the experiments.

Table I shows results for different feature selection schemes. *No feature pruning* refers to the scheme where all features located within the query's bounding box are used in the term vector computation. *Vertical pruning* refers to selecting only those features whose local image descriptors do not extend beyond the lower or upper query boundary. *Horizontal and vertical feature pruning* finally refers to the scheme where feature descriptors are not allowed to extend beyond any query boundary. The results clearly show the benefit of pruning features at the top and bottom. This way the noise resulting from the adjacent text lines can be be reduced. The mean average precision drops by an absolute 4.7% when additionally pruning features at the sides. Figure 2a shows that this is due to the decreased performance of shorter words. These are very likely to appear within longer words. For instance, *the* can be found in *they, them, other* etc. It is, therefore, especially important to model those words' contexts in the sentence. For longer words Figure 2a shows the importance of feature pruning. The more characters a query contains the more context from the upper and lower text line is encoded in the model, otherwise.

Table II shows the influence of different model lengths on the retrieval performance. As described in Section II-B, the number of states in the HMM is estimated based on the number of frames, i.e. different sliding window positions, within the query's bounding box. The respective parameter is given as a percental value of the frame number. As can be seen in Figure 2b, a 3% scaling factor works best for long queries and a scaling factor of 17% works best for short queries. The values have been determined in prior experiments. However, the best overall performance was achieved with a linear combination of the two variants. The considerable difference in mean average precision between a uniform and a query length dependent scaling theme clearly demonstrates the importance of this

Table I. INFLUENCE OF QUERY FEATURE PRUNING

Query feature pruning	mAP	mR
No feature pruning	58.2%	94.5%
Vertical feature pruning	$\mathbf{61.1\%}$	$\mathbf{95.5\%}$
Horizontal and vertical feature pruning	56.4%	95.2%

Table II. INFLUENCE OF THE NUMBER OF MODEL STATES

Model states scaling	mAP	mR
3% of query frame number	52.9%	92.9%
17% of query frame number	49.1%	93.8%
Linear combination $(\mathbf{17\% - 3\%})$	$\mathbf{61.1\%}$	$\mathbf{95.5\%}$

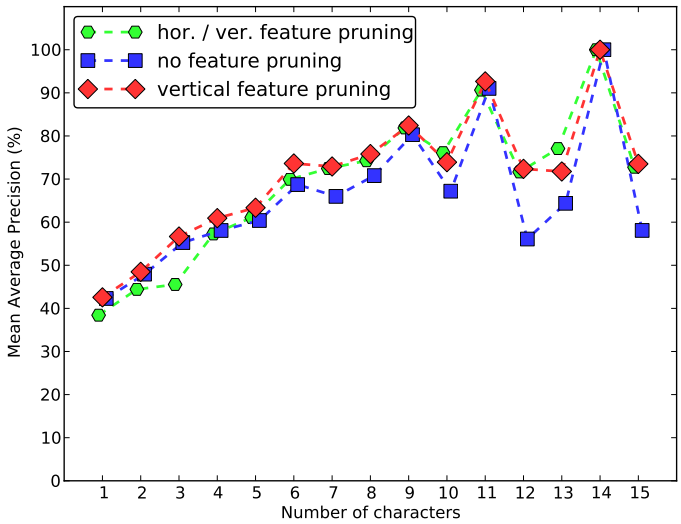
parameter. It controls how much of the query word is modeled by a single state. The smaller the modeled segments, the more accurate the HMM representation. Generally, a trade-off between specificity and generality must be found. In our given experimental scenario, we can observe (see Figure 2b) that a higher scaling factor works well for short queries and that performance decreases once the queries get longer. This can be explained with our patch-based configuration (cf. Section II-C). For efficiency reasons the patch shift is proportional to the patch size. Longer words are, therefore, often not aligned with the patches. By reducing the number of states the representation does not enforce a strict sequence of term vectors. We rather model the possible occurrence of different term vectors in fewer states. This gives us the flexibility of generating high scores even though only a part of a queried word is observed. Adapting the state number with respect to the query length allows us to address both requirements. Figure 2b shows how the scores obtained with the linear combination match the performances for short and long query words obtained with the respective uniform scaling approaches.

Table III shows the effect of Baum-Welch training. In the training process the probability of observing the query's term vector sequence is optimized. Similarly, as for the number of model states a compromise between specificity and generality must be found. For the considered benchmark on the George Washington dataset it is found after three iterations.

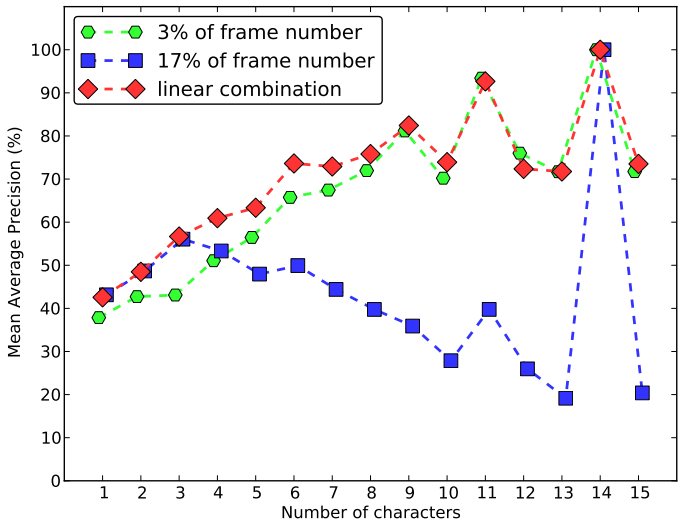
Finally, Table IV shows a performance overview of other state-of-the-art word spotting methods. The work of Rath and Manmatha [1] is often considered as a baseline experiment. However, we cannot directly compare it to our method due to the different prerequisites and the different dataset configurations. A comparison against the other two methods [9], [8] is possible, because here these requirements are met. Regarding the evaluation, we almost use the same protocol. Relative improvements of 12.3% [9] and 101.0% [8] clearly show how our proposed method outperforms the state-of-the-art results. The difference is especially noticeable for shorter words. The proposed strategy shares the Bag-of-Features application with [8] and the query-size dependent patch-based framework with [9]. The results are therefore suited to demonstrate the benefit of using Bag-of-Features HMMs in a segmentation-free word spotting scenario.

IV. CONCLUSION

In this paper we proposed a novel method for segmentation-free word spotting. With our Bag-of-Features HMMs we were able to outperform state-of-the-art methods on the George



(a) Influence of feature pruning for different word lengths



(b) Influence of the number of HMM states for different word lengths

Figure 2. Plots 2a and 2b show the effect of feature pruning and the number of HMM states with respect to different word lengths. Markers are used to visualize mean average precision scores for different parameterizations and discrete numbers of characters. Dashed lines are only drawn for an easier visual comparison. The red diamond markers indicate the configuration for the overall optimal scores found in the evaluation. Please note that the 100% mean average precision for 14 characters word length can be considered an artifact. Only a single word exists in this category. Generally, the average character number is 4.4 leading to a respective influence of words in that order of length when computing the mean average precision over all queries. Best viewed in color.

Table III. INFLUENCE OF BAUM-WELCH TRAINING

Baum-Welch iterations	mAP	mR
0	59.5%	95.1%
1	60.3%	95.3%
3	**61.1%**	**95.5%**
5	60.5%	95.5%
7	59.0%	95.3%

Table IV. COMPARISON OF STATE-OF-THE-ART WORD SPOTTING RESULTS FOR THE GEORGE WASHINGTON DATASET

Method	GW dataset configuration	mAP
Rath and Manmatha [1]	10 pages, 2381 queries	40.9%
Proposed	20 pages, 4860 queries	61.1%
Almazán et al. [9]	20 pages, 4856 queries	54.4%
Rusiñol et al. [8]	20 pages, 4860 queries	30.4%

Washington dataset. In comparison we could demonstrate that the retrieval performance for shorter words has been improved. This can be regarded as especially challenging. In addition, we only need a single sample for estimating our query model. Continuous HMMs, in contrast, require substantial amounts of training data. Future research could investigate to which extent word spotting performance can be improved when estimating Bag-of-Features HMMs based on a set of training samples.

REFERENCES

[1] T. Rath and R. Manmatha, "Word spotting for historical documents," *Int. Journal on Document Analysis and Recognition*, vol. 9, no. 2–4, pp. 139–152, April 2007.

[2] U. Marti and H. Bunke, "Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system," *Int. Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, no. 1, pp. 65–90, February 2001.

[3] A. Vinciarelli, S. Bengio, and H. Bunke, "Offline recognition of unconstrained handwritten texts using HMMs and statistical language models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 709–720, June 2004.

[4] J. Rodríguez-Serrano and F. Perronnin, "Handwritten word-spotting using hidden Markov models and universal vocabularies," *Pattern Recognition*, vol. 42, no. 9, pp. 2106–2116, September 2009.

[5] A. Fischer, A. Keller, V. Frinken, and H. Bunke, "Lexicon-free handwritten word spotting using character HMMs," *Pattern Recognition Letters*, vol. 33, no. 7, pp. 934–942, May 2012.

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

[7] B. Gatos and I. Pratikakis, "Segmentation-free word spotting in historical printed documents," in *Proc. of the Int. Conf. on Document Analysis and Recognition*, 2009, pp. 271–275.

[8] M. Rusiñol, D. Aldavert, R. Toledo, and J. Lladós, "Browsing heterogeneous document collections by a segmentation-free word spotting method," in *Proc. of the Int. Conf. on Document Analysis and Recognition*, 2011, pp. 63–67.

[9] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, "Efficient exemplar word spotting," in *Proc. of the British Machine Vision Conf.*, 2012, pp. 67.1–67.11.

[10] J. Rodríguez-Serrano and F. Perronnin, "Local gradient histogram features for word spotting in unconstrained handwritten documents," in *Proc. of the Int. Conf. on Frontiers in Handwriting Recognition*, 2008.

[11] L. Rothacker, S. Vajda, and G. Fink, "Bag-of-features representations for offline handwriting recognition applied to Arabic script," in *Proc. of the Int. Conf. on Frontiers in Handwriting Recognition*, 2012.

[12] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[13] S. O'Hara and B. A. Draper, "Introduction to the bag of features paradigm for image classification and retrieval," *Computing Research Repository*, vol. arXiv:1101.3354v1, 2011.

[14] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," in *IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 2169–2178.

[15] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[16] L. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *The Ann. of Math. Stat.*, vol. 41, pp. 164–171, 1970.