2009 10th International Conference on Document Analysis and Recognition

Handwritten word-image retrieval with synthesized typed queries

José A. Rodríguez-Serrano*
Computer Vision Centre
Universitat Autonoma de Barcelona, Spain
cojar@lboro.ac.uk

Florent Perronnin
Textual and Visual Pattern Analysis
Xerox Research Centre Europe, France
Florent.Perronnin@xrce.xerox.com

Abstract

We propose a new method for handwritten word-spotting which does not require prior training or gathering examples for querying. More precisely, a model is trained "on the fly" with images rendered from the searched words in one or multiple computer fonts. To reduce the mismatch between the typed-text prototypes and the candidate handwritten images, we make use of: (i) local gradient histogram (LGH) features, which were shown to model word shapes robustly, and (ii) semi-continuous hidden Markov models (SC-HMM), in which the typed-text models are constrained to a "vocabulary" of handwritten shapes, thus learning a link between both types of data. Experiments show that the proposed method is effective in retrieving handwritten words, and the comparison to alternative methods reveals that the contribution of both the LGH features and the SC-HMM is crucial. To the best of the authors' knowledge, this is the first work to address this issue in a non-trivial manner.

1. Introduction

There are many applications in document retrieval where a fundamental step is to match a candidate word image to a prototype, where the prototype is a representation of the concept that is queried. The prototype can be an exemplary image containing the search word [4] or a model built using one or several images [7]. In both cases, one or more images are necessary and must be manually collected from documents.

For matching typed-text words, a strategy has been proposed that does not require the manual collection of prototypes. The idea is to generate the prototypes automatically. If the font type is uniform and known, it is straightforward to render a word image with that very font and to use the synthesized word image for matching [3]. The straightforward application of this idea to handwritten words is difficult because, despite some efforts in the synthesis of handwritten words (see e.g. [10]), it is still an open problem.

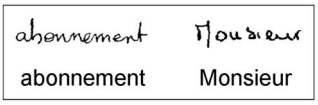

Figure 1. Handwritten and typed versions of the same word

We therefore propose to synthesize *typed* text images using computer fonts to match *handwritten* images. At first thought, one could think that such a method would yield unpractical results because there is a significant mismatch in the shapes of typed-text and handwritten letters. Especially, the variability of writing styles is much higher than that of typed text. However, if we consider the example in Fig. 1, there are reasons to hope. Indeed, several typed/handwritten letters share a common shape. In this article, we show that robust image descriptors – such as the LGH features [8] – and statistical models – such as semi-continuous Hidden Markov models (SC-HMM) [2] – might be able to cope with the observed variations and reduce the mismatch. Also, there is a wide range of available fonts to mimic different writing styles and allographs. To the best of our knowledge, this is the first time that typed-text words are employed to match handwritten words using a non-trivial method.

The rest of the article is structured as follows. §2 summarizes the proposed method. §3 reviews the LGH features. §4 explains the use of SC-HMM in the proposed system. In §5 the experimental validation is reported. Finally, §6 contains the conclusions and future work.

2. Proposed solution

To robustly match handwritten words using typed-text templates, we propose two ingredients, one at the feature

*He is now with the Computer Science Department, Loughborough University, United Kingdom.

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.201

level and one at the modeling level:

- *LGH features [8] for robust description of word shapes*: it was shown that LGH features encode word shapes robustly. Our experimental results show that LGH features are more robust than other state-of-the-art features to handwritten / typed variations.
- *Use of semi-continuous hidden Markov models (SC-HMM) [2] for word modeling*: in a SC-HMM first the feature space is clustered using a Gaussian mixture model (GMM), and then the model parameters are constrained to that GMM. In this work we use SC-HMM for training models using typed images but the key point is to use a GMM estimated from handwritten images. Then, the constraints of the SC-HMM make that the model learns a "link" between handwritten and typed data. An additional advantage of the SC-HMM is that several examples can be combined into a single model so that scoring is performed once. This is to be contrasted with image matching approaches such as DTW where one should perform one DTW comparison per query image.

With these two elements, we propose a system which is able to find a query string in a handwritten document collection. Training the query model for string S requires the following steps:

1. Train off-line a GMM with a large unordered set of LGH features extracted from many *handwritten* word images. This GMM is independent of the query string and can be learned once and for all.
2. For query string S generate one or more typed-word images using selected computer fonts.
3. For each synthesized word image, extract a sequence of LGH features.
4. Train a SC-HMM using these sequences.

We note that the choice of the fonts at step 2 is a problem of paramount importance as fonts which look more "handwritten-like" will certainly lead to better models. We will get back to this point in the experimental section.

At runtime, the spotting process works as follows. For each candidate handwritten word-image:

1. Extract a sequence of LGH features.
2. Score the sequence on the SC-HMM query model and take a decision.

Details will follow in the next paragraphs.

It should be remarked that the intention of the proposed solution is not to compete with handwriting recognition

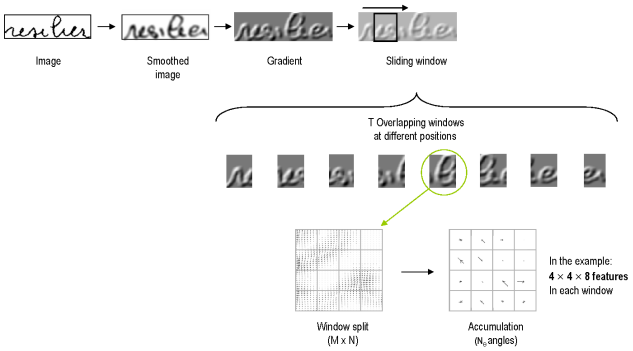

Figure 2. Computation of the LGH features

models that could obtain better performance in the same task when trained with an adequate and sufficient amount of data. In contrast, the proposed idea represents a solution at a reduced cost for *spotting without handwritten prototypes* when a recognition system is not available.

3. LGH features for robust shape encoding

LGH features [8] follow the *sliding window* approach in which a window traverses the word image from left to right. At each position of the window, a set of features are computed using only those pixels contained in the window. The feature extraction in a given window consists of three steps:

1. Adjust the upper and lower bounds of the sliding window to the area actually containing pixels
2. Split the reduced window into a 4x4 grid
3. At each of the obtained cells, compute the gradient and accumulate a histogram of 8 angle orientations.

For further details, we refer the reader to [8]. Fig. 2 provides a schematic overview of the process.

4. Semi-continuous HMM for modeling the link between typed and handwritten

The hidden Markov model [6] is a state-of-the-art tool for modeling handwritten words [5, 9]. A hidden Markov model has three types of parameter: the initial occupancy probabilities, the transition probabilities and the emission probabilities. In the case of continuous variables, the emission probabilities are generally modeled with Gaussian Mixture Models (GMM) whose parameters can be subdivided into mixture weights and Gaussian mean vectors and covariance matrices.

A special case is the semi-continuous HMM (SC-HMM). This model assumes that the feature space has been partitioned using a GMM-based clustering, where $\{\mu_k, \Sigma_k\}_{k=1}^K$ denotes the set of means and covariances of each Gaussian component. The kth component is also referred to as the kth Gaussian codeword. The obtained means and covariances are then shared in all the states of the SC-HMM. In other words, the emission probability at the ith state of the SC-HMM takes the form

$$p_i(x) = \sum_{k=1}^{K} w_{ik} \mathcal{N}(x|\mu_k, \Sigma_k), \quad (1)$$

where $\mathcal{N}(x|\mu, \Sigma)$ denotes a Gaussian with mean μ and covariance Σ. Therefore, the only state-dependent parameters to estimate in a SC-HMM are the weights w_{ik}.

There is strong evidence that the Gaussian codewords encode prior information about the problem of interest. For instance, in [7] it was shown that a SC-HMM can be trained with a single sample and still outperform a DTW-based matching.

In this article, we show that these constraints of the SC-HMM can help reducing the mismatch between typed and handwritten data. We build SC-HMMs to represent words with synthesized typed samples, but we will estimate the Gaussian codewords from *handwritten data*. Therefore, even if the free parameters of the SC-HMM are estimated from typed data, the constrained parameters (means and covariances) convey handwritten information. Such a model is likely to perform better when confronted with handwritten samples, as will be the case.

5. Experimental validation

This section reports the experimental validation. First, the experimental setting is described. Second, the performance of the system is evaluated as a function of the font type. The system is compared to alternative methods to prove that LGH features and SC-HMM are actually crucial for making the method work. Third, we show how combining multiple typed fonts might lead to improved accuracy. Fourth, we compare the proposed typed-queries with handwritten queries.

5.1. Experimental setting

To validate the proposed system we carried out a set of experiments on a database of 105 real scanned letters written in French provided by the customer department of a large corporation. This database is particularly challenging owing to the variability of writers, styles, artifacts and other anomalies such as spelling mistakes. The occurrences of 10 of the words (Monsieur, Madame, contrat, résiliation, salutation, résilier, demande, abonnement, company name and veuillez) are labelled for evaluation purposes.

Standard segmentation techniques are employed to obtain a set of word image hypotheses. Over-segmentation is employed to produce a large set of hypotheses. About 250 candidate word-images are generated per document image. Each candidate word-image is described as a sequence of 128-dimensional LGH features.

A GMM with 512 Gaussians is trained using approximately 1,000,000 feature vectors randomly extracted from a separate set of letters. All the SC-HMMs involved in the experiments below are trained on top of this GMM and use 10 states per character.

The performance of the detection task is evaluated in terms of the average precision (AP), which represents the average of the precision value in a precision/recall plot. We perform experiments for the 10 different keywords and report the mean across the 10 keywords (mean average precision or mAP).

Comparison to alternative methods In order to assess the role of both the LGH features and the SC-HMM in the proposed approach, we will repeat the retrieval experiments using (i) an alternative, standard set of features, and (ii) an alternative, standard image matching approach. This would correspond to the "trivial" solution to the problem of typed-to-handwritten matching. Regarding the features, we chose the zoning features proposed by Vinciarelli et al. [9]. This feature set is a standard one for word modeling and consists in counting the pixels of a 4x4 split of the window. As for the image matching approach we use the standard DTW.

5.2. Models using a single font

In the first round of experiments, we use a single synthesized sample per query. Training a SC-HMM with a single sample does not lead to over-fitting because of the a priori information encoded in the universal vocabulary (i.e. GMM). Indeed, we show below that training a SC-HMM with a single sample is more effective than a template matching approach using DTW.

In this case, for a desired word, we generate a single word image using a computer font. We evaluate the performance of the retrieval task as a function of the employed font face, where we have experimented with the most usual computer fonts, shown in Fig. 3. Fig. 4 shows the mAP for each font.

It can be observed that, for 19 out of the 25 fonts the approach using LGH features and SC-HMM outperforms the standard approaches. In particular, for 18 out of the 25 fonts the proposed approach obtains a relative increase of over 20% with respect to the best alternative approach; in 13 out of the 25 the relative increase is over 50%; and in 7

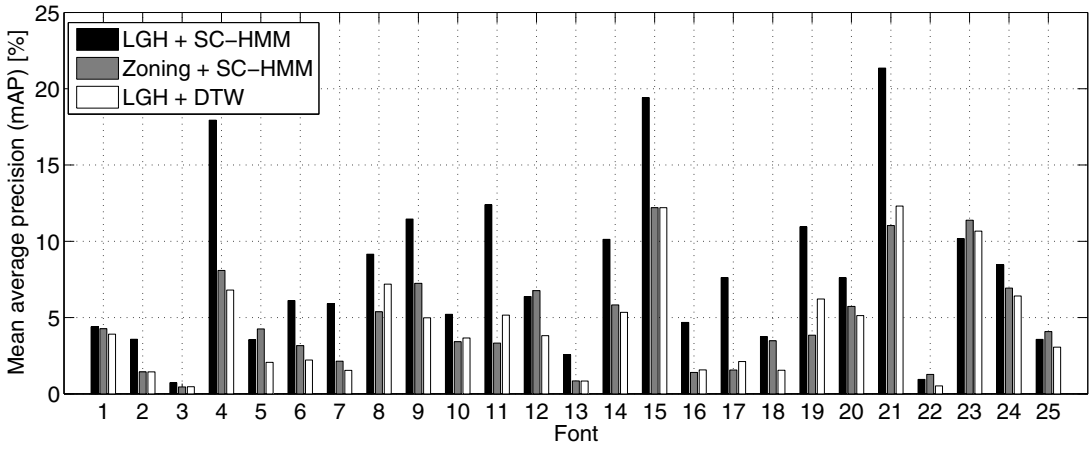

Figure 4. Results (mAP) with single-font models, comparing the proposed method (left) to alternative methods.

1 Times New Roman
2 Arial
3 Courier New
4 Lucida handwriting
5 Bradley hand
6 Viner Hand
7 Brush Script
8 Rage Italic
9 Comic Sans
10 Monotype corsiva
11 Harlow Solid
12 Tempus Sans
13 Matura MT Script
14 Freestyle Script
15 French Script
16 Ravie
17 Forte
18 Verdana
19 Gigi
20 Lucida Caligraphy
21 Kunstler Script
22 Juice ITC
23 Papyrus
24 Poor Richard
25 OCR A Extended

Figure 3. The 25 font faces used in the experiments

.

out of 25 it is over 100%. We conclude that both the LGH features and the SC-HMM are crucial for matching typed words against handwritten words.

The importance of the handwritten UBM for learning the link between typed and handwritten images is evidenced by the following fact: when repeating the experiment but using a UBM computed from typed text images, the mAP for 23 out of the 25 fonts is less than 3%. This poor result is due to the fact that no prior information about handwritten shapes is considered in this case, which confirms our choice.

Another interesting observation is that the best ranked fonts (e.g. Kunstler Script, French Script, Lucida Handwriting) are very handwritten-like, while the classical typed fonts (e.g. Times, Arial, Courier, OCR) rank low.

5.3. Models using multiple fonts

In the next experiment, we generate word images using different fonts and use *several* images to train the SC-HMM. The question is whether the retrieval accuracy can be improved by using multiple fonts.

Based on the ranking of fonts in Fig. 4, we trained a model using the N-best fonts, with $N = 1, 2, 3, \ldots, 25$. Fig. 5 shows the mAP as a function of the number N of fonts. The best performance is obtained by considering the best 9 fonts ($> 32\%$), compared to the 21 % obtained when using the best single font. This is a significant improvement of the retrieval accuracy. Of course, this set of 9 fonts might not be the optimal one among all possible combination of fonts. Also, the best combination will be writer-dependent as different fonts model more appropriately the writing styles of different persons. This will be considered in future work.

5.4. Comparison to handwritten prototypes

To understand whether the obtained mAP values are reasonable, we compare the performance of the proposed system based on typed-queries with that of a standard system based on handwritten queries. Hence, we repeat similar ex-

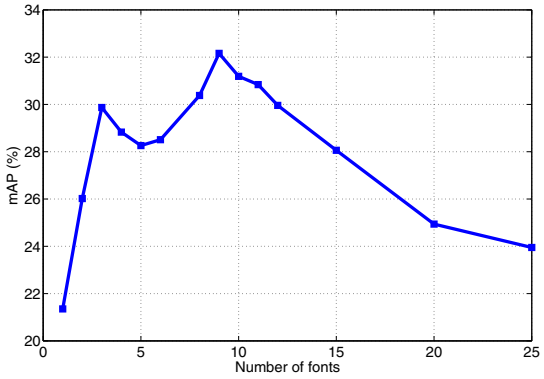

Figure 5. Results using the best N fonts

.

periments but using handwritten images instead of synthesized images to train the SC-HMM.

Again, we run the retrieval experiments in the case of a single query image and multiple query images. To be as independent as possible from the particular choice of the training samples, we repeat each experiment 10 times using different handwritten training sets and average the results.

Using a single handwritten image to query the system, we obtain a mAP of 17.6%. This is worse than the result obtained with the best individual typed font (21%), and almost half the result obtained when combining several fonts ($>$ 32%). *Thus with the proposed approach we can get better performance than querying with a single handwritten font, and without the effort of manually collecting the query sample*.

When querying with 25 handwritten images, we obtain a mAP of 64.0%. This value is significantly higher than in the typed case but the manual collection of 25 samples comes at a cost, especially in the case of rare words. What we would like to make clear is that the proposed system based on synthesized queries leads to a reasonable performance, even if not comparable to the handwritten queries, but allows searching for any keyword without the need for collecting samples.

6. Conclusions and perspectives

This article proposes a new method for handwritten word spotting without prototypes. In our approach, the prototypes are automatically generated using typed-text fonts, and two mechanisms contribute to a robust matching of typed-to-handwritten words: (i) the LGH features, and (ii) the use of SC-HMMs. Experimental results show that the approach has a competitive performance and that the two mentioned factors are the cause of the improvement.

Although we would obtain better results of models are trained using handwritten samples directly, the current method enables quick access to digital libraries by completely eliminating the cost of collecting prototypes. Therefore, one interesting perspective of the system is to use it for querying a certain word, and then using the output handwritten samples to train a more accurate handwritten model, e.g. with active learning techniques [1].

Acknowledgments

When this work was conducted, the J.A. Rodríguez-Serrano was a visitor at the Xerox Research Centre Europe. His work has been partially supported by the Spanish projects TIN2006-15694-C02-02, TIN2008-04998 and CONSOLIDER-INGENIO 2010 (CSD2007-00018).

References

[1] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, pages 129–145, 1996.

[2] X. D. Huang and M. A. Jack. Semi-continuous hidden Markov models for speech signals. In *Readings in speech recognition*, pages 340–346, 1990.

[3] T. Konidaris, B. Gatos, K. Ntzios, I. Pratikakis, S. Theodoridis, and S. J. Perantonis. Keyword-guided word spotting in historical printed documents using synthetic data and user feedback. *Int. J. on Document Analysis and Recognition*, 9(2-4):167–177, 2007.

[4] R. Manmatha, C. Han, and E. M. Riseman. Word spotting: A new approach to indexing handwriting. In *CVPR*, page 631, 1996.

[5] U.-V. Marti and H. Bunke. Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system. *Int. J. of Pattern Recognition and Artificial Intelligence*, 15:65–90, 2001.

[6] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of the IEEE*, 77:257–286, 1989.

[7] J. A. Rodríguez and F. Perronnin. Handwritten word-spotting using hidden Markov models and universal vocabularies. *Pattern Regognition*.

[8] J. A. Rodríguez and F. Perronnin. Local gradient histogram features for word spotting in unconstrained handwritten documents. 2008. ICFHR.

[9] A. Vinciarelli, S. Bengio, and H. Bunke. Offline recognition of unconstrained handwritten texts using HMMs and statistical language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):709–720, 2004.

[10] J. Wang, C. Wu, Y.-Q. Xu, and H.-Y. Shum. Combining shape and physical modelsfor online cursive handwriting synthesis. *Int. J. Doc. Anal. Recognit.*, 7(4):219–227, 2005.