A Keyword Spotting Approach Using Blurred Shape Model-based Descriptors

Alicia Fornés
Computer Vision Center - Dept. of Computer Science
Universitat Autònoma de Barcelona, Edifici O, 08193
Bellaterra, Spain
afornes@cvc.uab.es

Volkmar Frinken
Institute of Computer Science and Applied Mathematics
Neubrückstrasse 10
3012 Bern, Switzerland
frinken@iam.unibe.ch

Andreas Fischer
Institute of Computer Science and Applied Mathematics
Neubrückstrasse 10
3012 Bern, Switzerland
afischer@iam.unibe.ch

Jon Almazán
Computer Vision Center - Dept. of Computer Science
Universitat Autònoma de Barcelona, Edifici O, 08193
Bellaterra, Spain
almazan@cvc.uab.es

Gabriel Jackson
Institute of Computer Science and Applied Mathematics
Neubrückstrasse 10
3012 Bern, Switzerland
jax@students.unibe.ch

Horst Bunke
Institute of Computer Science and Applied Mathematics
Neubrückstrasse 10
3012 Bern, Switzerland
bunke@iam.unibe.ch

ABSTRACT

The automatic processing of handwritten historical documents is considered a hard problem in pattern recognition. In addition to the challenges given by modern handwritten data, a lack of training data as well as effects caused by the degradation of documents can be observed. In this scenario, keyword spotting arises to be a viable solution to make documents amenable for searching and browsing. For this task we propose the adaptation of shape descriptors used in symbol recognition. By treating each word image as a shape, it can be represented using the Blurred Shape Model and the Deformable Blurred Shape Model. Experiments on the George Washington database demonstrate that this approach is able to outperform the commonly used Dynamic Time Warping approach.

Categories and Subject Descriptors

I.5.4 [**Pattern Recognition**]: Applications—*Text Processing*; I.5.2 [**Pattern Recognition**]: Design Methodology—*Feature evaluation and selection*

1. INTRODUCTION

The fully automatic recognition of handwritten text is after decades of research still a very challenging field [25]. The large variety of different writing styles as well as degraded documents in case of historical data pose several challenges. Common approaches make use of learning-based systems that require large amounts of annotated images to train a recognition engine. For historical documents, however, such training data may not exist, or its creation may be tedious and costly, since it has to be done manually. Nevertheless, libraries all over the world store huge numbers of handwritten books that are crucial for preserving the world's cultural heritage.

In such a scenario, keyword spotting and retrieval [4, 5] –the task of retrieving all instances of a given word– offers a viable solution to make documents amenable to searching and browsing. Certain efforts have already been put into word spotting for historical data [20, 16]. Another related application is the segmentation of images of historical documents into meaningful regions, which can be improved with keyword spotting. In [12] the keyword "Fig." is spotted in the images to help identifying figures and their corresponding captions. Finally, it is worth mentioning that Google and Yahoo have announced their intention to make handwritten books accessible through their search engines [14]. In this context, keyword spotting will be a valuable tool for users browsing the contents of these books.

The field of keyword spotting can be divided into *query-by-example* (QBE) and *query-by-string* (QBS) approaches. QBS describes the setup in which the user enters an arbitrary character string into the system. Although it allows maximum flexibility as far as the set of keywords is concerned, this approach requires at least a small correctly transcribed training set. If no such data is available, QBE can be applied. The user selects one or a few words by looking at the data set and the system retrieves all words with a similar shape. Hence, this approach can also be seen as a special case of the general image retrieval problem. The most prominent technique for QBE is dynamic time warping [20].

In this paper we propose to use shape descriptors for the task of keyword spotting. The rational is to consider the word image as a shape, and consequently, describe it us-

ing shape descriptors. Among the shape descriptors able to cope with the variabilities of handwriting styles, Shape Context [3], Blurred Shape Model [7] and the recently proposed Deformable Blurred Shape model [2] have shown to be a good choice for recognizing hand-drawn symbols. For a more extensive list on shape descriptors see [26].

The Shape Context descriptor concatenates global histograms for corresponding points to include the context of the shape. This has been been shown to perform well when used for highly distorted hand-drawn symbols. The matching, however, requires a point-to-point alignment of two shapes using a graph-matching algorithm. The high computational costs involved renders this approach unsuitable for word spotting applications.

The Blurred Shape Model (BSM) descriptor encodes the spatial probability of appearance of the shape pixels and their context information, and extracts a feature vector which describes the shape. The Deformable Blurred Shape model (DBSM) is an improved version of the BSM, allowing a higher degree of deformation by the integration of a deformable model into the BSM descriptor. Contrary to Shape Context, both BSM and DBSM descriptors allow the fast comparison of two shapes by directly comparing their feature vectors, e.g., using the Euclidean distance.

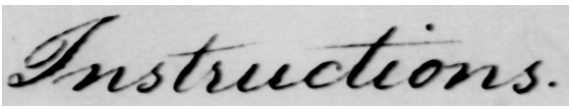
Both descriptors, BSM and DBSM are designed for symbol recognition and are therefore scale invariant. For words, however, the length of the words is an important property for computing their similarity. Consequently, we present a modification of the BSM and DBSM descriptors for keyword spotting and further describe the keyword spotting approach based on these descriptors. We demonstrate the superior performance compared to a common, state-of-the-art DTW reference system.

The rest of the paper is structured as follows. After the preprocessing step described in Section 2, the BSM and DBSM descriptors are introduced in Sections 3 and 4. The adaptation of BSM and DBSM to word spotting is described in Section 5. In Section 6 the DTW reference system is reviewed. The experimental evaluation of the proposed system is presented in Section 7 and, finally, in Section 8, conclusions are drawn.

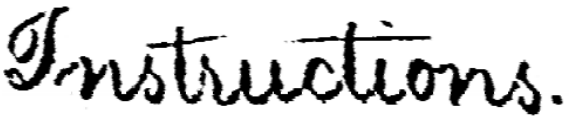
2. PREPROCESSING

The segmented text lines are normalized prior to recognition in order to cope with different writing styles. First, the skew angle is determined by a regression analysis based on the bottom-most black pixel of each pixel column. Then, the skew of the text line is removed by rotation. Afterwards the slant is corrected in order to normalize the direction of long vertical strokes found in characters like 't' or 'l'. After estimating the slant angle based on a histogram analysis, a shear transformation is applied to the image. Next, a vertical scaling is applied to obtain three writing zones of the same height, i.e., lower, middle, and upper zone, separated by the lower and upper baseline. To determine the lower baseline, the regression result from the skew correction is used, and the upper baseline is found by vertical histogram analysis. For more details on the text line normalization operations, we refer to [17].

Finally the width of the text is normalized. For this purpose, the average distance of black/white transitions along a horizontal straight line through the middle zone is determined and adjusted by horizontal scaling. An example of a preprocessed word image is shown in Figure 1.



(a)



(b)

Figure 1: Preprocessing. (a) Original word (b) Preprocessed word.

3. BLURRED SHAPE MODEL

The Blurred Shape Model (BSM) descriptor [7] is proposed for the recognition of hand-drawn symbols in historic documents. Similar to the recognition of handwritten words, hand-drawn symbol recognition is a hard task due to the high variability of symbol appearance encountered within different writing styles. Furthermore, historic documents are more prone to artifacts such as noise, torn paper, degraded ink, or bleed-through.

BSM encodes the spatial probability of appearance of the shape pixels and their context information in the following way. The image is divided into a grid of $n \times n$ equally-sized subregions. For each grid element a weighted sum of all pixels in the surrounding grid elements is computed. Thus, each foreground pixel contributes to a density measure of grid element it lies in and its neighboring ones.

In Figure 2, a letter shape parametrization is shown. Figure 2(a) shows the distances estimation of a shape point respect to the nearest centroids. To give the same importance to each shape point, all the distances to the neighbors centroids $\{d_1, d_2, d_3, d_4\}$ are normalized so that $\frac{1}{d_1} + \frac{1}{d_2} + \frac{1}{d_3} + \frac{1}{d_4} = 1$. The output descriptor is a vector where each entry corresponds to the number of pixels in a sub-region. Afterwards it is normalized, so that all entries are in the range $[0, 1]$ and sum up to 1. Hence, a vector can be considered as a probability density function (pdf) of $n \times n$ bins. This way, the output descriptor represents a probability distribution of the object shape. Since a shape is represented as a vector, two shapes can be compared efficiently by computing their (Euclidean) distance.

BSM is scaling and (x, y)-stretching invariant because of the fixed size of the $n \times n$ grid. The selection of the grid size is an important parameter, which defines the region of activity of the symbol's pixels (see Fig. 3), and consequently, the resolution by which the shape is sampled (also denoted as the blurring degree). The optimum grid size depends on the dataset and must be able to reflect the difference between inter-class and intra-class variability. The algorithm is summarized in table 1. For further details, see [7].

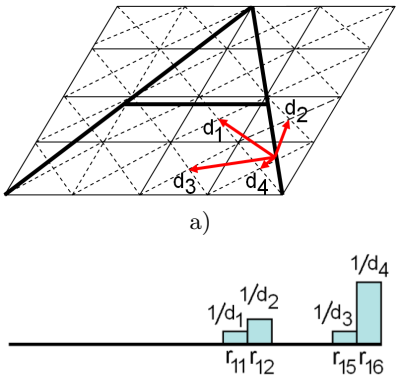


Figure 2: BSM density estimation example. (a) Distances $d_1..d_4$ of a given shape pixel to the neighboring centroids. The regions are numbered r_1 to r_{16}. (b) The contribution of the pixel indicated above to the corresponding grid elements.

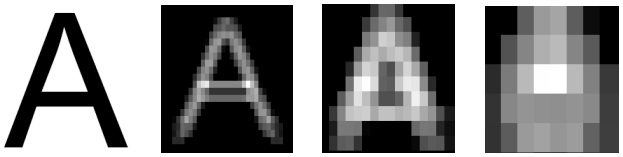


Figure 3: BSM with different grid sizes. (a) Input image. (b) 32x32 regions. (c) 16x16 regions. (d) 8x8 regions.

Given an image I:

1. Obtain the *shape* S contained in I.

2. Divide I in $n \times n$ equal size sub-regions $R = \{r_1, ..., r_{n^2}\}$, with c_i the center of points for each region r_i, $i \in [1, .., n^2]$.

3. Let $N(r_i)$ be the neighbor regions of region r_i, defined as $N(r_i) = \{r_k | r_k \in R, ||c_k - c_i|| < 2|g|\}$, where g is the cell size.

4. Let $r_i^{\mathbf{x}}$ be the region which contains the point x.

5. Initialize the probability vector v as $v(i) = 0$, $\forall\ i \in [1, .., n^2]$.

6.

For each point $\mathbf{x} \in S$,

$D = 0$

For each $r_i \in N(r_k^{\mathbf{x}})$,

$d_i = d(\mathbf{x}, r_i) = ||\mathbf{x} - c_i||^2$

$D = D + \frac{1}{d_i}$

End_For

Update the probability vector v as $v(r_i) = v(r_i) + \frac{1}{d_i D}$

End_For

7. Normalize v as: $v(i) = \frac{v(i)}{\sum_{j=1}^{n^2} v(j)}\ \forall\ i \in [1, ..., n^2]$

Table 1: Blurred Shape Model description algorithm.

4. DEFORMABLE BSM

The Blurred Shape Model (BSM) has shown to be tolerant to distorted shapes. However, it is not robust enough in case of large deformations. In this case, the integration of the BSM with a deformable model arises as an appealing alternative. The Deformable Blurred Shape Model (DBSM) described in [2] integrates the BSM descriptor with non-linear deformable model, the Image Distortion Model (IDM) [11]. This new model is based on deforming the grid structure of the BSM in order to adapt it to the given shape.

First of all, and in order to allow deformations of the grid, instead of the BSM regular grid of size $k \times k$, a set of $k \times k$ points (named focuses) are equidistantly distributed over the image. These *focuses* correspond to the centroids of the original regular grid and, as in the BSM approach, accumulate votes of the neighboring pixels weighted by their distance. Instead of defining the neighborhood as a set of fixed cells of the grid, it is defined as an arbitrary *influence area* centered on the focus, in order to provide flexibility. The deformation of the grid is obtained by moving independently each of the focuses along with their respective influence area. In order to limit the amount of deformation, each focus is allowed to move only inside a pre-defined *deformation area*. Figure 4 shows an example of the focus representation and their influence and deformation areas. This resulting representation provides more flexibility and allows the focus deformation tracking.

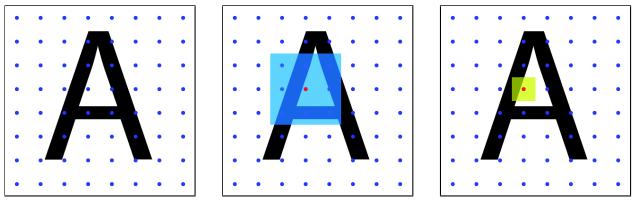
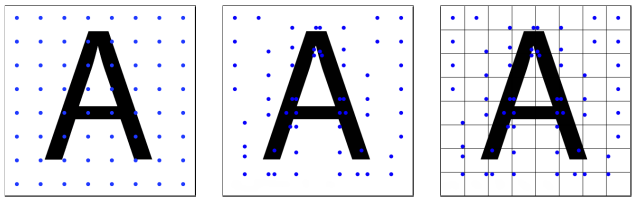


Figure 4: Deformable Blurred Shape Model representation (extracted from [2]). (a) Focuses representation. (b) Influence area. (c) Deformation area.

Afterwards, every focus is moved independently inside the deformation area to maximize its accumulated BSM value. Therefore, the final position of each focus is the local maximum of the density measure within its deformation area. Figure 5 shows an example of this process. As a result, every image is represented with two output descriptors: a vector v which contains the BSM value of each focus, and the vector p containing the (x, y) coordinates of each focus.

In this paper, we use the second matching technique proposed in [2], where the focuses move to maximize its own BSM value (note that this process is independent of the training image). Firstly, it allows the independent computation of the feature vectors for all images in the database, and secondly, it allows a fast comparison among the different feature vectors using Euclidean distance. Given two shapes I and J, let the vectors $\mathbf{v_I}$ and $\mathbf{v_J}$ contain the BSM values of the focuses of I and J, and the vectors $\mathbf{p_I}$ and $\mathbf{p_J}$ contain the position coordinates of the focuses. After normalizing the vectors, the distance between two shapes can be computed via



(a) (b) (c)

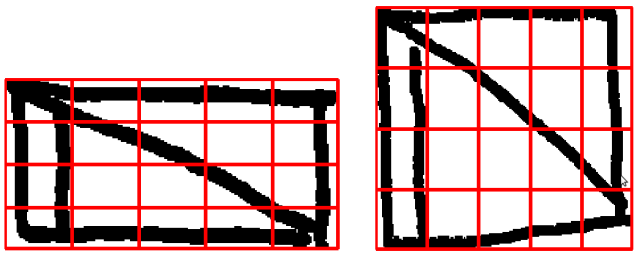
Figure 5: Example of the focuses deformation (extracted from [2]). (a) Initial position of the focuses. (b) Final position of the focuses after the maximization of their values. (c) Deformation area used.

$$distance(I, J) = d(\mathbf{v_I}, \mathbf{v_J}) \cdot \alpha + d(\mathbf{p_I}, \mathbf{p_J}) \cdot (1 - \alpha) \quad (1)$$

where d is the Euclidean distance between two vectors, and α is a weighting factor. For further details, see [2].

5. ADAPTATION TO WORD SPOTTING

As it has been mentioned in the introduction, we need to adapt the BSM and DBSM descriptors for keyword spotting. For this purpose, the descriptors should also reflect the length of a word. The length of a word plays a crucial role in estimating similarity. This is contrary to symbol recognition, where size invariance is usually desired because it allows to cope with differences in symbol size. In Figure 6 we can see how the BSM grid is correctly adapted to the two segmented symbols, resulting in very similar feature vectors although their size is different.

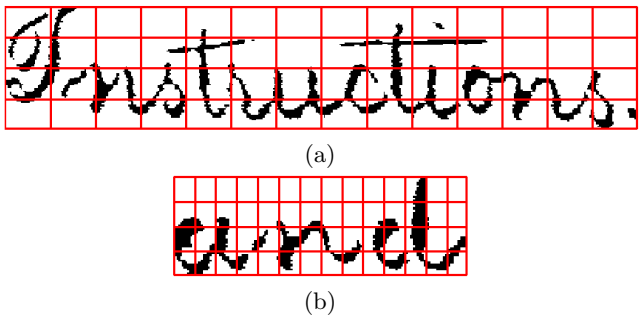


(a) (b)

Figure 6: Two examples of the same architectural symbol, and their BSM grid.

The following example illustrates the problem that arises when using the BSM or DBSM descriptors without any modifications. Given a grid size of 4×14, the word *Instructions* (Fig.7(a)) is distributed along $4 \times 14 = 56$ cells. This means that about 1-2 columns of the cells are used for encoding each character. However, if the word *and* is described using the same grid size of 4×14 (Fig.7(b)), instead of 1-2 columns, 4-5 columns of cells are used to encode each character. Consequently, it may occur that two words with different lengths could obtain similar feature vectors.

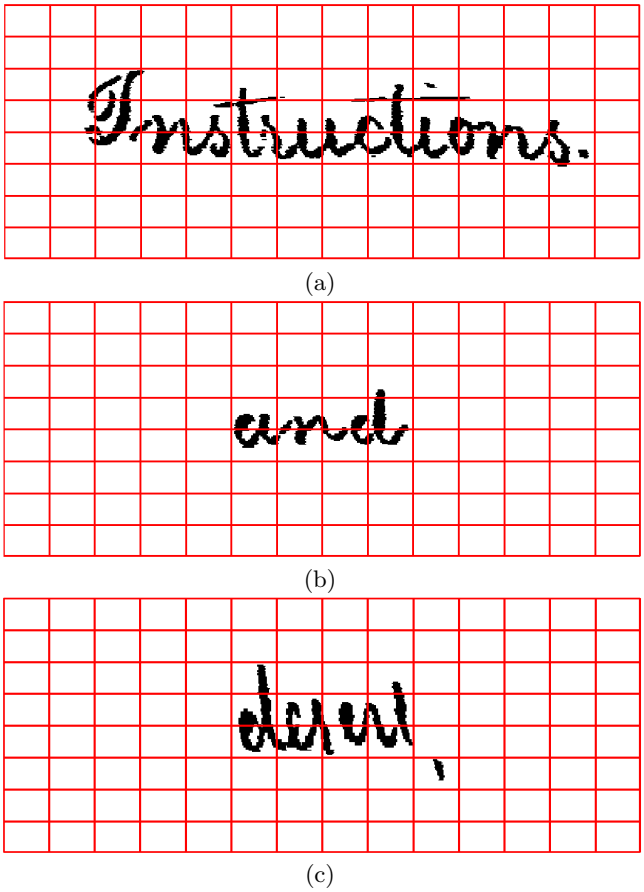
The straight-forward approach of using a different grid size for each word results in feature vectors of different size and sophisticated matching techniques are needed. Hence, the



(a) (b)

Figure 7: Two words with the same grid size. Here, the number of columns used for describing each character is different.

comparison between different words in the database can not be performed using Euclidean distance.



(a) (b) (c)

Figure 8: Words located in a blank template image. Here, the number of columns used for describing each character is similar.

In order to cope with this problem, we propose to create a template blank image, where every word image of the database will be located in the center of this template image, according to its own centroid (see Fig.8).

Using the template, we obtain three important advantages:

- The same number of cells will be used for describing the characters of each word.
- The feature vector of a short word (see Fig.8(a)) will be completely different (e.g. more cells containing 0 values) from that of a long word (see Fig.8(b)).
- Using the center of gravity the approach is robust to noise. Even incorrectly segmented word images are correctly located in the center of the template, as shown in Fig.8(b), where some noisy pixels or punctuation marks can be seen.

6. DTW REFERENCE SYSTEM

Dynamic Time Warping (DTW) is a dynamic programming approach that finds an optimal alignment between two sequences by a pairwise comparison of elements of the first sequence to elements of the second sequence. Each element in the one sequence can be assigned to several consecutive elements in the other sequence. In [6], DTW was proposed for word spotting in speech recognition, and also the first approaches to word spotting for handwritten text used DTW representing text as a sequence of features vectors. Various features have been proposed in conjunction with DTW [24, 21, 19]. We use the nine features proposed in [17], extracted via a sliding window moving from left to right over the image. At each of the N positions of the sliding window, n features are extracted. The sliding window has a width of one pixel. It is moved in steps of one pixel, i.e., N equals the width of the text line. From each window, $n = 9$ geometric features are extracted, three global and six local ones. The global features are the 0^{th}, 1^{st} and 2^{nd} moment of the black pixels' distribution within the window. The local features are the position of the top-most and that of the bottom-most black pixel, the inclination of the top and bottom contour of the word at the actual window position, the number of vertical black/white transitions, and the average gray scale value between the top-most and bottom-most black pixel. To compute the inclination of the top and bottom contour, the sliding window to the left of the actual one is considered.

Our DTW implementation, similarly to the one described in [20], makes use of a Sakoe-Chiba band [23] to speed-up the computation. The only pruning criterion we used was the length of the word, i.e. one word image must not be more than twice as long as the other.

In order to spot a certain keyword, all instances of that word occurring in the training set are compared to all words in each text line. In this paper, we consider a perfect, manually corrected word segmentation in order to rule out the influence of segmentation errors on the word spotting performance. The minimum of all these *DTW* distances serves as a distance function of the keyword's word class to the text line. If the DTW distance of a keyword to the text line is below a given threshold, the text line and the word having the minimum distance is returned as a positive match.

7. EXPERIMENTAL EVALUATION

In this section we will describe the database, metrics, comparatives and the experiments performed.

7.1 Database

In order to validate the proposed approach, we use the George Washington Dataset (GW DB) because it is frequently used for keyword spotting [18, 20, 22, 24, 15, 10, 13, 1, 8]. It consists of 20 pages of letters, orders and instructions of George Washington written in the year 1755. One example of these letters is shown in Figure 9. The pages originate from a large collection with a variety of images, the quality of which ranges from clean to very difficult to read. In our experiments, we have used the same pages as used by Rath and Manmatha in [20], which are: George Washington Papers at the Library of Congress from 1741-1799, Series 2, Letterbook 1, pages 270-279 and 300-309 (to be found at http://memory.loc.gov/ammem/gwhtml/gwseries2.html).

The selected pages we use are relatively clean. The text is part of a larger corpus, written not only by George Washington but also by some of his associates. It inhibits some variations in writing style. However, the writing on the pages we consider is fairly similar. The considered pages include 4,894 words on 675 text lines. The GW DB contains the same pages as the one in [13], but we found the automatically segmented and extracted words to be too erroneous. Hence, we decided to use the already preprocessed (see Section 2) and manually segmented and labelled word images used in [9].

7.2 Metrics

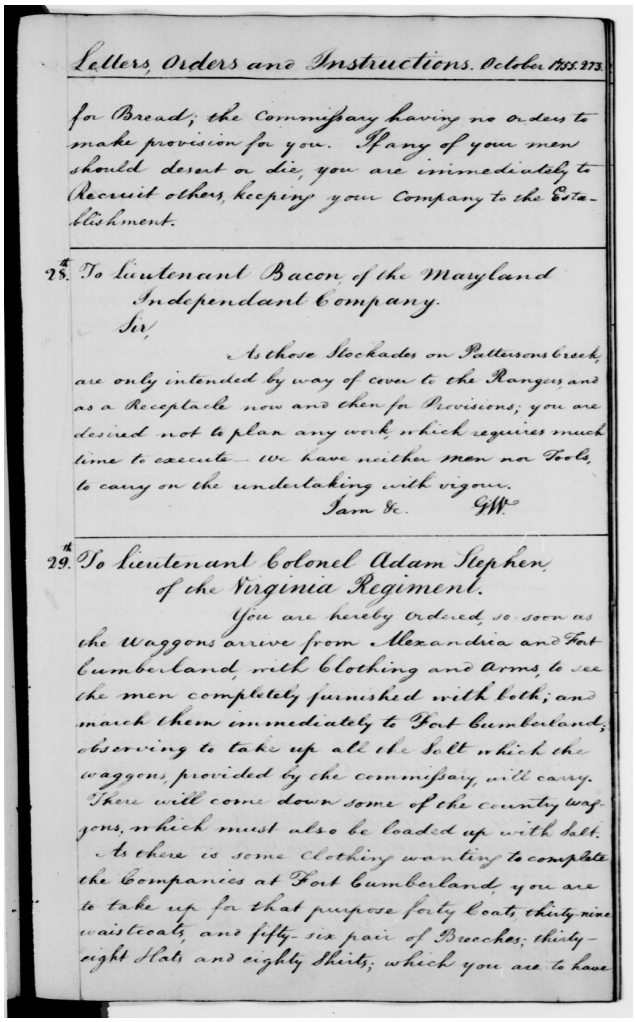
A retrieval system returns for each document a similarity measure that indicates how relevant the document is. Given a query word image, the BSM, DBSM and DTW approaches return the distance to the nearest word image among of the reference word images. All of these scores can be transformed into a relevance measure $r \in [0; 1]$ where a higher r value indicates that a document is more relevant.

The performance of a single system can be measured by applying a threshold $\theta \in [0, 1]$ to the document score. Those with $r > \theta$ are returned as positive matches, the rest are rejected as negative matches. For various different thresholds θ we can hence compute the number of *true positives* (TP), *true negatives* (TN), *false positives* (FP), and *false negatives* (FN). Given these values, *precision* and *recall* of a system can be estimated. Precision is defined as the number of relevant objects found by the algorithm divided by the number of all objects found, while recall is defined as the number of relevant objects found divided by the number of all relevant objects in the test set:

$$\text{precision} = \frac{TP}{TP + FP}$$
$$\text{recall} = \frac{TP}{TP + FN}$$

To compare several systems, a system's performance should be expressed by a single value. The *average precision* (ap) is the average over all recall values, and the evaluation measure used in this paper is the mean of all average precisions over all queries, called *Mean average Precision* (MaP).

7.3 Comparatives


Letters, Orders and Instructions. October 1755. 273

for Bread; the Commissary having no orders to make provision for you. If any of your men should desert or die, you are immediately to Recruit others, keeping your Company to the Establishment.

28th To Lieutenant Bacon of the Maryland Independant Company.

Sir,

As those Stockades on Pattersons Creek are only intended by way of cover to the Rangers and as a Receptacle now and then for Provisions; you are desired not to plan any work which requires much time to execute — We have neither men nor Tools to carry on the undertaking with vigour.

I am &c. GW.

29th To Lieutenant Colonel Adam Stephen of the Virginia Regiment.

You are hereby ordered, so soon as the Waggons arrive from Alexandria and Fort Cumberland, with Clothing and Arms, to see the men completely furnished with both; and march them immediately to Fort Cumberland, observing to take up all the Salt which the waggons, provided by the commissary, will carry. There will come down some of the country waggons, which must also be loaded up with Salt. As there is some Clothing wanting to complete the Companies at Fort Cumberland, you are to take up, for that purpose forty Coats, thirty-nine waistcoats, and fifty-six pair of Breeches; thirty-eight Hats and eighty Shirts; which you are to have

Figure 9: George Washington example page.

As it has been mentioned before, the methods to be compared are BSM, DBSM and DTW. The details of the descriptors used in the comparison are the following. For the BSM we have used two different grid sizes, where in the first one each cell is about 4×4 pixels, and in the second case, each cell is about 5×5 pixels. For a fair comparison with DBSM, we have located a focus each 4×4 pixels, and in the second configuration, one every 5×5 pixels. With these two configurations, we tested the performance of the approach with two different blurring degrees.

Concerning the DBSM approach, four different values of α are proposed for weighting the differences in the BSM values (also called *intensity values*) and the changes in the positions of focuses. In case of $\alpha = 0$, only the differences in the position of the focuses are used for computing the similarity between two words. The value $\alpha = 0.3$ means that the differences in BSM values account for 30% of the final distance and the focuses positions 70%, whereas $\alpha = 0.7$ means that the differences in BSM values account 70% and 30% the positions. Finally, when $\alpha = 1$, only the BSM values are taken into account for computing the similarity between two words.

7.4 Results and Discussion

Figure 10 shows the mean average precision (MaP) of the BSM and DBSM with a cell size of 5×5, whereas Figure 11 shows the mean average precision with a cell size of 4×4 pixels. The BSM with a cell size of 5×5 pixels obtains a MaP of 54.79%, whereas it increases to 56.41% in case of a cell size of 4×4. Similarly, the DBSM results are higher when the grid resolution is higher, increasing from 56.15% (5-pixels distance) to 58.16% (4-pixels distance). These results show that, on this specific dataset, the higher the sampling resolution, the better the performance. Notice that in other databases, the optimal resolution could be different.

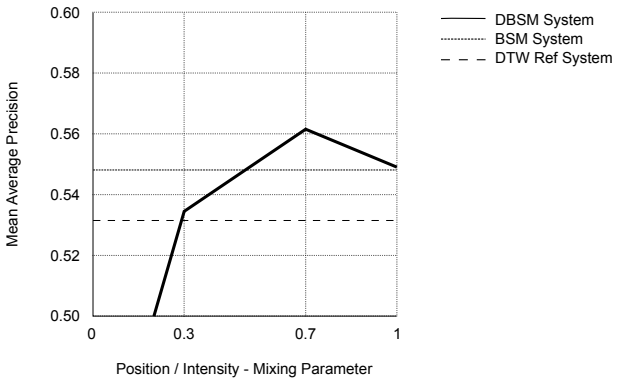


Figure 10: Results for a cell size of 5x5 pixels: Mean average precision for different values of α = Position/Intensity-Mixing parameter.

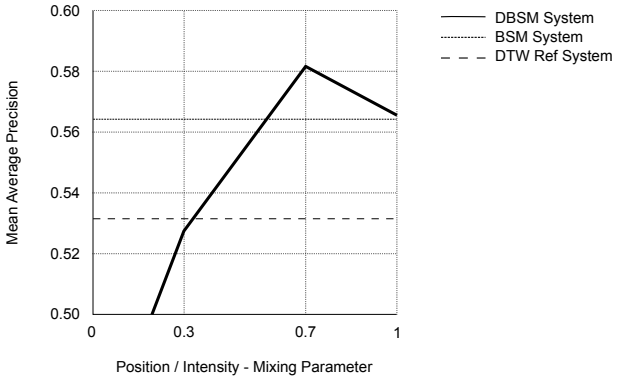


Figure 11: Results for a cell size of 4x4 pixels: Mean average precision for different values of α = Position/Intensity-Mixing parameter.

Concerning the weighting of the DBSM, the *intensity values* are more discriminant than the positions of the focuses, obtaining the best results with $\alpha = 0.7$. When only the focuses positions are taken into account, the MaP dramatically decreases to 43%. Not surprisingly, if only the BSM values are taken into account, the DBSM performance (56.5%) is very similar to the BSM approach (56.41%).

It is important to remark that both BSM and DBSM approaches outperform the DTW method, with a mean average precision of 53.14%. In addition, one important advantage of the BSM and DBSM approaches compared to the DTW method is the lower computational cost. The DTW

method requires the computation of a matrix for comparing each pair of words, which has a complexity order of $O(n^2)$. In contrast, in the BSM and DBSM approaches, the feature vectors can be compared using Euclidean distance, which has a complexity order of $O(n)$, and consequently, allowing faster keyword spotting systems. Note that all descriptors can be computed off-line in a preprocessing step.

Referring to the comparison between the BSM and DBSM approaches, results show that DBSM obtains slightly better results (58.16% compared to 56.41%), especially when both position and intensity values are taken into account. However, the computation of the BSM descriptor is much faster than the DBSM descriptor and thus, the BSM approach seems to offer a good trade-off between complexity and performance increase.

Concerning the comparison between words, however, the BSM approach only requires to compare the BSM vectors, whereas the DBSM approach requires to compare more than one vector; viz. the BSM values and the focuses position (x,y coordinates). Therefore, the final decision should be made depending on the database at hand.

Although both proposed systems, BSM and DBSM, have a mean average precision of below 60%, this is the best currently reported performance for the given setup. Other experiments have demonstrated that it is possible to obtain higher performance on this dataset for training based systems, e.g. in [9]. In contrast, the BSM, DBSM and DTW proposals do not require any training data, which renders them suitable for searching in databases when no ground truth is available.

8. CONCLUSIONS

In this paper we have proposed a shape-based keyword spotting approach, which makes use of the Blurred Shape Model (BSM) and the Deformable Blurred Shape Model (DBSM). We have also described the adaptation required for dealing with word images instead of symbols.

Experimental results show that both BSM and DBSM approaches outperform the DTW method, and also, they are faster to compute. When compared to BSM, it can be concluded that the slightly higher performance of the DBSM comes at the cost of an increased computational complexity. Thus, the final choice should be made depending on the size of the database and the performance requirements of the final user.

Further work will be focused on the exploration of other shape descriptors suitable for handwritten text. Especially a fast way to compute Shape Context descriptor or approximations thereof are promising research directions for keyword spotting.

9. ACKNOWLEDGMENTS

This work has been partially supported by the Swiss National Science Foundation (CRSI22_125220), by the European project FP7-PEOPLE-2008-IAPP: 230653 as well as by the Spanish projects TIN2008-04998, TIN2009-14633-C03-03, CONSOLIDER-INGENIO 2010 (CSD2007-00018), the research grant of the Universitat Autonòma de Barcelona (471-01-8/09) and the José Castillejo mobility research grant JC2010-0112.

10. REFERENCES

[1] T. Adamek, N. E. Connor, and A. F. Smeaton. Word matching using Single Closed Contours for Indexing Historical Documents. *Journal on Document Analysis and Recognition*, 9(2):153–165, 2007.

[2] J. Almazán, E. Valveny, and A. Fornés. Deforming the Blurred Shape Model for Shape Description and Recognition. *Pattern Recognition and Image Analysis, Lecture Notes in Computer Science*, 6669:1–8, 2011.

[3] S. Belongie, J. Malik, and J. Puzicha. Shape Matching and Object Recognition Using Shape Contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[4] H. Cao, A. Bhardwaj, and V. Govindaraju. A probabilistic method for keyword retrieval in handwritten document images. *Pattern Recognition*, 42(12):3374–3382, 2009.

[5] H. Cao, V. Govindaraju, and A. Bhardwaj. Unconstrained handwritten document retrieval. *International Journal on Document Analysis and Recognition*, pages 1–13, 2011.

[6] A. E. R. Cory S. Myers, Lawrence R. Rabiner. An Investigation of the Use of Dynamic Time Warping for Word Spotting and Connected Speech Recognition. In *IEEE Int'l Conf. on Acoustics, Speech, and Signal Processing*, pages 173–177, 1980.

[7] S. Escalera, A. Fornés, O. Pujol, P. Radeva, G. Sánchez, and J. Lladós. Blurred Shape Model for binary and grey-level symbol recognition. *Pattern Recognition Letters*, 30(15):1424–1433, 2009.

[8] S. Feng. *Statistical Models for Text Query-Based Image Retrieval*. PhD thesis, University of Massachusetts, Amherst, May 2008.

[9] 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*, accepted for publication.

[10] N. R. Howe, T. M. Rath, and R. Manmatha. Boosted Decision Trees for Word Recognition in Handwritten Document Retrieval. In *28th Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 377–383, 2005.

[11] D. Keysers, T. Deselaers, and C. Gollan. Deformation models for image recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8):1422–1435, 2007.

[12] K. Khurshid, C. Faure, and N. Vincent. Fusion of Word Spotting and Spartial Information for Figure Caption Retrival in Historical Document Images. In *10th Int'l Conf. on Document Analysis and Recognition*, volume 1, pages 266–270, 2009.

[13] V. Lavrenko, T. M. Rath, and R. Manmatha. Holistic Word Recognition for Handwritten Historical Documents. In *Int'l Workshop on Document Image Analysis for Libraries*, pages 278–287, 2004.

[14] S. Levy. Google's Two Revolutions. *Newsweek Dec. 27 / Jan. 3*, 2004.

[15] Y. Leydier, F. Lebourgeois, and H. Emptoz. Text

Search for Medieval Manuscript Images. *Pattern Recognition*, 40:3552–3567, 2007.

[16] Y. Leydier, A. Ouji, LeBourgeois, and H. Emptoz. Towards an Omnilingual Word Retrieval System for Ancient Manuscripts. *Pattern Recognition*, 42(9):2089–2105, 2009.

[17] U.-V. Marti and H. Bunke. Using a Statistical Language Model to Improve the Performance of an HMM-Based Cursive Handwriting Recognition System. *Int'l Journal of Pattern Recognition and Artificial Intelligence*, 15:65–90, 2001.

[18] T. M. Rath, V. Lavrenko, and R. Manmatha. A Statistical Approach to Retrieving Historical Manuscript Images without Recognition. Technical Report MM-42, Center for Intelligent Information Retrival, 2003.

[19] T. M. Rath and R. Manmatha. Features for word spotting in historical manuscripts. In *7th Int'l Conf. Document Analysis and Recognition*, pages 218–222, 2003.

[20] T. M. Rath and R. Manmatha. Word Spotting for Historical Documents. *Int'l Journal of Document Analysis and Recognition*, 9:139–152, 2007.

[21] J. A. Rodríguez and F. Perronnin. Local Gradient Histrogram Features For Word Spotting in Unconstrained Handwritten Documents. In *11th Int'l Conf. Frontiers in Handwriting Recognition*, pages 7–12, 2008.

[22] J. Rothfeder, S. Feng, and T. M. Rath. Using Corner Feature Correspondences to Rank Word Images by Similarity. In *Workshop on Document Image Analysis and Retrieval*, page 30, 2003.

[23] H. Sakoe and S. Chiba. Dynamic Programming Algorithm Optimization for Spoken Word Recognition. *Trans. on Acoustics, Speech, & Signal Processing*, 26:43–49, 1978.

[24] K. Terasawa and Y. Tanaka. Slit Style HOG Features for Document Image Word Spotting. In *10th Int'l Conf. on Document Analysis and Recognition*, volume 1, pages 116–120, 2009.

[25] A. Vinciarelli. A Survey On Off-Line Cursive Word Recognition. *Pattern Recognition*, 35(7):1433–1446, 2002.

[26] D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37(1):1–19, 2004.