A Non-Rigid Feature Extraction Method for Shape Recognition

Jon Almazán, Alicia Fornés, Ernest Valveny
Computer Vision Center – Dept. Ciències de la Computació
Universitat Autònoma de Barcelona
Bellaterra, Barcelona, Spain
{*almazan,afornes,ernest*}*@cvc.uab.es*

***Abstract*—This paper presents a methodology for shape recognition that focuses on dealing with the difficult problem of large deformations. The proposed methodology consists in a novel feature extraction technique, which uses a non-rigid representation adaptable to the shape. This technique employs a deformable grid based on the computation of geometrical centroids that follows a region partitioning algorithm. Then, a feature vector is extracted by computing pixel density measures around these geometrical centroids. The result is a shape descriptor that adapts its representation to the given shape and encodes the pixel density distribution. The validity of the method when dealing with large deformations has been experimentally shown over datasets composed of handwritten shapes. It has been applied to signature verification and shape recognition tasks demonstrating high accuracy and low computational cost.**

***Keywords*-Shape recognition; Non-rigid representation; Geometrical centroids; Pixel density**

I. Introduction

Shape recognition is one of the classic problems in Computer Vision. It has been tackled from different points of view and it is in constant evolution. This is due to the fact that shape recognition is used in a multitude of applications with important challenges such as noise, degradations or elastic deformations. Therefore, shape descriptors should be robust enough in order to guarantee intra-class compactness and inter-class separability in the presence of distortions.

In our case, we are interested in shape descriptors that could be applied to handwriting recognition. This is one of the applications where a large variability poses a big challenge to shape descriptors. Several descriptors have been applied to this field (1). For instance, the curvature scale space (CSS) descriptor (2), which successively blurs the shape contour by convolving it with a Gaussian kernel, where the scale is increased at each level of blurring. It is tolerant to deformations but it can only be used for closed contours. Zernike moments (3) introduces a set of rotation-invariant features based on the magnitudes of a set of orthogonal complex moments of the image. Another well-known descriptor is Shape Context (4), which selects n points from the contour of the shape, and for each of them, computes the distribution of the distance and angle with respect to the other points. It is tolerant to deformations, and is able of dealing with open regions. In the particular context of hand-drawn symbol recognition, the Blurred Shape Model (BSM) (5) has been introduced as a robust descriptor to classify deformed symbols. It is based on computing the spatial distribution of shape pixels in a set of pre-defined image sub-regions taking into account the influence of neighboring regions. The use of neighborhood information permits to handle a certain degree of deformation. However, due to the rigidity of the model, large deformations cause large differences in the spatial information encoded by the BSM.

We find several shape recognition approaches that are based on non-rigid representations in the literature. These are deformable representations that adapt to the shape to be described. For example, (6; 7) describe shape recognition methods based on the extraction of features from sub-images obtained by iteratively partitioning the original image. These partitions are extracted by dividing sub-images using the geometrical centroid and following a hierarchical structure. Then, a descriptor is obtained either by extracting features related to the density of the resulting sub-images, or by using the coordinates of the geometrical centroids.

This partitioning procedure can be seen as a non-regular distribution of geometrical centroids located over regions of the image with high density, *i.e.*, a non-regular distribution of points that is iteratively adapting to the given shape. Therefore, this procedure can be used to improve the BSM in order to avoid its rigid grid-based representation. The main contribution of our work is the integration of this deformable scheme in the BSM approach in order to build a new method capable of dealing with deformations. The result is a new descriptor that will adapt its representation to the given shape using an iterative region partitioning procedure based on geometrical centroids. Then, using this new distribution of geometrical centroids, that can be seen as a deformable grid, we will extract the BSM-based pixel density measure in order to compute the final shape descriptor.

The ability of our method to deal with deformations has been proved using two applications closely related to shape recognition: off-line signature verification and symbol recognition. For signature verification, we will use as benchmark the GPDS signature corpus (8), and we will compare our results with Vargas *et al.* methods (9; 10), which reports the lowest EER in this dataset. For symbol recognition, we will use the NicIcon dataset (11), composed of handwritten

symbols from different writers. These experiments will show the following facts: first, that our method is able to deal with large deformations, second, that outperforms the original BSM, and finally, that its performance in signature verification is comparable with specific methods in the literature when working with random forgeries.

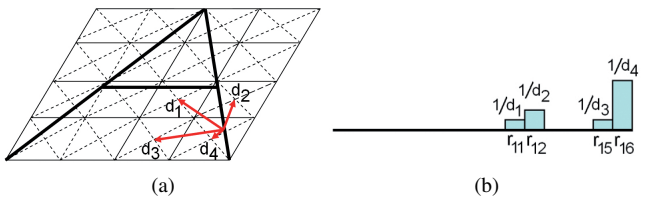
The rest of the paper is organized as follows: Section II is devoted to explain the proposed method, while Section III explains the classification technique used. The explanation of the experiments, which includes the experimental protocols and the datasets used, is conducted in Section IV. Then, performance results, as well as the comparison with the original approach and the state of the art, are shown in Section V. Finally, Section VI concludes the paper.

II. Methodology

As we stated in the introduction, the proposed model is based on the Blurred Shape Model (BSM). So first, we will give a brief introduction about this method. Then, we will detail the new representation proposed to deal with deformations, and the computation of the new descriptor.

A. Blurred Shape Model

The main idea behind the BSM descriptor (5) is to describe a given shape by a probability density function encoding the probability of pixel densities of a certain number of image sub-regions. Given a set of points forming the shape of a particular symbol, each point contributes to compute the BSM descriptor. This is done by dividing the given image in a $n \times n$ grid with equal-sized sub-regions (cells). Then, each cell receives votes both from the *shape pixels* located inside its corresponding cell, and also from those located in the adjacent cells. Thereby, every pixel contributes to the density measure of its sub-region cell, and its neighboring ones. This contribution is weighted according to the distance between the point and the centroid of the cell receiving the vote. In Fig. 1 an example of the contribution for a given pixel is shown. The output is a vector histogram, where each position contains the accumulated value of each sub-region, and contains the spatial distribution in the context of the sub-region and its neighbors.



(a) (b)

Figure 1. BSM density estimation example. (a) Distances of a given shape pixel to the neighboring centroids. (b) Vector descriptor update using distances of (a).

B. Non-rigid feature extraction

The pixel density distribution encoded by the BSM has been shown to obtain good results when dealing with handwritten symbol recognition (5). However, the rigidity of the model, due to the regular grid-based representation, is not suitable when dealing with large elastic deformations. We propose a model to capture the pixel density distribution but using a deformable grid representation. Instead of the regular grid of size $k \times k$ we will place over the image a set of $k \times k$ points following a non-regular distribution. These points, denoted as *focuses*, will accumulate votes of the neighboring pixels weighted by their distance as it is done in the original approach. Moreover, instead of defining the neighborhood as a set of fixed cells of the grid, it will be defined as an arbitrary $h \times w$ *influence area* centered on the focus, which will provide more flexibility. The size of this area is the same for all the focuses.

Focus distribution. The distribution of focuses in the image will be driven by the idea of maximizing the pixel density around the focus. Consequently, focuses will be distributed over the shape to be represented, *i.e.*, the *grid* will be adapted to the shape of the object. Our method is based on the *region partitioning* procedure of the Adaptive Hierarchical Density Histogram (AHDH) (6), which consists in iteratively producing regions of the image using the geometrical centroid estimation. The coordinates of the *focuses* will be the position of these geometrical centroids.

First, we consider the binary image as a distribution of *shape pixels* in a two-dimensional space-background (Figure 2a). The set of shape pixels is defined as S and their number as N. Furthermore, we define as R_i^l, $i = \{1, 2, \ldots, 4^l\}$ the i-th rectangular region obtained in the iteration (or 'level') l of the partitioning algorithm, and as $F^l \in \mathbb{R}^2$ the set of geometrical centroids of the regions in R^l. For each level l, the ***region partitioning*** procedure estimates the geometric centroid of all regions R_i^l and then splits each region into four sub-regions using as a center the geometric centroid. The new sub-regions generated will form the new set of regions R^{l+1} .The initial region, R^0, is the whole image, and F^0 would contain the geometrical centroid of this region (Figure 2b). Considering a separate cartesian coordinates system for each region R_i^l, the geometrical centroid F_i^l is computed using equations

$$\mathbf{x_c} = \frac{\sum_{(x,y)\in S_i^l} \mathbf{x}}{\mathbf{N_i^l}}, \; \mathbf{y_c} = \frac{\sum_{(x,y)\in S_i^l} \mathbf{y}}{\mathbf{N_i^l}}, \qquad (1)$$

where N_i^l denotes the number of shape pixels set S_i^l in the processed region R_i^l, and $(\mathbf{x}, \mathbf{y})$ are the pixel coordinates. This iterative procedure finishes when a termination level L is reached. Then, the final coordinates of the *focuses* will be the geometrical centroids computed in the level L, that is F^L. Thus, the number of focuses to represent the shape (4^L) can be determined using this termination level L. An

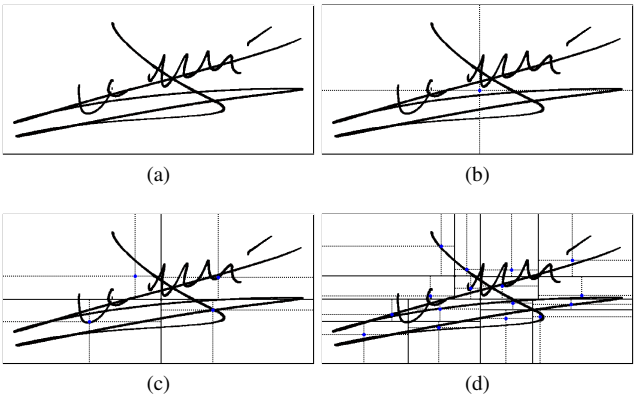


Figure 2. Focuses distribution computation based on the region partitioning algorithm: (a) original image, (b), (c) and (d) focuses (in blue) at level 0, 1 and 2 respectively.

example of the distribution of focuses for different levels is shown in Figure 2.

Features computation. Once the position of the focuses for a given shape is obtained, we compute two different output features: a vector histogram $\mathbf{f} \in \mathbb{R}^{4^L}$, which contains the density measure of nearby pixels of each focus, and $\mathbf{p} \in \mathbb{R}^{2\times 4^L}$ containing the coordinates of the focuses position. Feature vector $\mathbf{f}$ uses the previously mentioned $h \times w$ *influence area* to calculate the pixel density around every focus. Focuses will receive votes from the neighboring shape pixels, which are those inside this *influence area* defined around the focus. Based on the BSM (5), this vote is weighted according to the distance between the pixel and the focus. Feature vector $\mathbf{p}$ contains x and y coordinates of the position of the focus, normalized by the width and height of the image, respectively.

III. Classification

For classification a Support Vector Machine classifier is trained using the feature vectors obtained with our method. The decision function of a SVM for a test sample with feature vector $\vec{F'}$ has the form

$$g(\vec{F'}) = \sum_{\vec{F}\in trainingset} \alpha_{\vec{F}} y_{\vec{F}} k(\vec{F}, \vec{F'}) - \beta, \qquad (2)$$

where $y_{\vec{F}}$ is the class label of $\vec{F}$, $\alpha_{\vec{F}}$ is the learned weight of train sample $\vec{F}$, β is a learned threshold and $k(\vec{F}, \vec{F'})$ is the value of a kernel function based on a given distance d. We use the χ^2 distance, which has shown good results in recognition (12)

$$d(\vec{F}, \vec{F'}) = \frac{1}{2}\sum_{i=1}^{n} \frac{(\vec{F_i} - \vec{F'_i})^2}{\vec{F_i} + \vec{F'_i}}, \qquad (3)$$

where n is the dimension of the feature vector. Then, kernel function is computed using equation

$$k(\vec{F}, \vec{F'}) = e^{-\frac{1}{D}d(\vec{F},\vec{F'})}, \qquad (4)$$

with D a scalar which normalizes the distances. We set D to the average χ^2 distance between all elements of the train set (12). Furthermore, when combining features, the kernel may be extended in a weighted fusion for m features using

$$A = \sum_{j=1}^{m} \frac{w_j}{D_j} d(\vec{F_j}, \vec{F'}_j) \qquad (5)$$

$$k(\{\vec{F}_1, \ldots, \vec{F}_m\}, \{\vec{F'}_1, \ldots, \vec{F'}_m\}) = e^{-\frac{A}{W}}, \qquad (6)$$

with w_j the weight of the j-th feature, D_j the normalization factor for the j-th feature, $\vec{F_j}$ the j-th feature vector, and $W = \sum_{j=1}^{m} w_j$. We use the LibSVM implementation (13) to train the classifier.

IV. Experiments

This new descriptor has been experimentally evaluated for two different purposes: signature verification and symbol recognition. Following, we will introduce the public datasets selected for each case, as well as the evaluation protocol and the performance measure used.

A. Signature verification

Signature verification consists in, given a signature, decide whether it belongs to a concrete individual or not. That is, classify the signature as genuine or forgery. We have evaluated our methodology for off-line signature verification using the **GPDS signature corpus** (8) (Figure 3a), which is divided in two subsets. The GPDS-100 and GPDS-750 signature corpus contains 24 genuine signatures and 24 forgeries of 100 and 750 different individuals, respectively. Samples have been obtained using different writing ink pens, and have been digitized with a scanner at 600 dpi. All signatures are binary images with different sizes.

Two different **evaluation protocols** are commonly used in signature verification: *skilled forgeries* and *random forgeries*. In both protocols we use the same training set configuration. For each individual, we randomly choose 5 genuine signatures as positive samples, and, as negative samples, we select a genuine signature of each one of the rest of users in the database (99 for both GPDS-100 and GPDS-750). For testing, first, the rest of genuine samples (19 signatures) are selected. Then we choose, in the case of the *skilled*

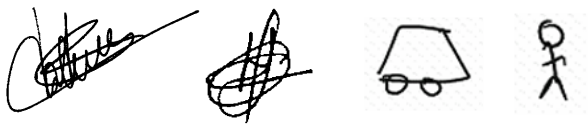


Figure 3. Image samples of the datasets. (a) GPDS. (b) NicIcon.

forgeries configuration, all available forgeries, and in the case of *random forgeries*, another genuine sample of each one of the rest of users in the database (different sample to the one used for training). Training and testing procedure were repeated 10 times with different subsets in order to obtain reliable results. As evaluation measure, two types of error were considered: False Rejection Rate (FRR), when genuine signatures are rejected, and False Acceptance Rate (FAR), when forgery signatures are accepted. The Equal Error Rate (EER), which is the value when the FAR and FRR are equal, is also considered.

B. Symbol recognition

We have also evaluated the new descriptor for the task of symbol recognition. We will measure the performance using the accuracy rate in classification over the **NicIcon** dataset (11) (Figure 3b). This dataset is composed of 26163 handwritten symbols of 14 classes from 34 different writers and it is commonly used for on-line symbol recognition, but off-line data is also available. The dataset is divided in three subsets (training, validation and test) for two different settings: *writer dependent* and *writer independent*. We have selected the off-line data with both configurations as benchmark to test our method. Every symbol has been binarized and cropped in an image of 240×240 pixels.

V. Results and analysis

We now show the benefits of the proposed method using the experiments in Section IV. We have trained the SVM classifier introduced in Section III with two different configurations of the features explained in Section II: using only the feature vector $\mathbf{f}$ containing the value of the pixel density around every focus (**nrBSM$_\mathbf{f}$**), and using a fusion in kernel (Equation (6)) for both pixel density $\mathbf{f}$ and coordinates position $\mathbf{p}$ feature vectors (**nrBSM$_\mathbf{f+p}$**).

First, we report results on signature verification in Tables I and II for skilled forgeries and random forgeries protocols, respectively. We compare our results with the original BSM, and also with Vargas *et al.* methods (9; 10), which report the lowest EER in the GPDS dataset. We can appreciate that our non-rigid feature extraction method outperforms the original grid-based BSM in both subsets with both protocols proving that the inclusion of a deformable grid is more capable to deal with deformations. Compared to Vargas *et al.* approaches (9; 10), whilst our method is surpassed in the case of the skilled forgeries protocol, it achieves better results with random forgeries. These results demonstrate the good properties of the proposed descriptor to represent the shape of the signature. In skilled forgeries, specific features of the signatures, not necessarily related to shape, can be decisive to achieve a low FAR. In this case (Table I), our method keeps a low FRR, but lacks in the capacity to extract the necessary features to differentiate a genuine and a skilled forgery signature, leading to a high FAR. However, in random forgeries, where signatures from different writers are compared, the capacity to capture the basic shape structure of the signature is the key element to obtain a low FAR, as is the case with our method. Thus, we demonstrate that our method achieves a good performance when recognizing the general structure of the shape is required. Furthermore, for the specific case of signature verification, our approach may be improved to deal with skilled forgeries combining it with specific features not based on shape.

Table I
Signature verification results for skilled forgeries

Dataset	Method	FAR(%)	FRR(%)	EER(%)
GPDS-100	Vargas (9)	**5.13**	20.82	**12.06**
	BSM (5)	17.85	20.71	19.11
	nrBSM$_\mathbf{f}$	16.27	16.38	16.29
	nrBSM$_\mathbf{f+p}$	18.67	**14.79**	16.93
GPDS-750	Vargas (10)	**9.26**	27.59	**17.36**
	BSM (5)	32.42	21.15	27.44
	nrBSM$_\mathbf{f}$	28.78	**19.37**	24.16
	nrBSM$_\mathbf{f+p}$	27.72	21.82	25.11

Table II
Signature verification results for random forgeries

Dataset	Method	FAR(%)	FRR(%)	EER(%)
GPDS-100	Vargas (9)	0.27	21.87	3.75
	BSM (5)	0.11	24.45	4.72
	nrBSM$_\mathbf{f}$	**0.06**	16.82	2.76
	nrBSM$_\mathbf{f+p}$	**0.07**	**14.22**	**2.35**
GPDS-750	Vargas (10)	0.09	19.92	3.28
	BSM (5)	0.22	21.44	3.64
	nrBSM$_\mathbf{f}$	0.07	18.16	2.98
	nrBSM$_\mathbf{f+p}$	**0.04**	**15.63**	**2.56**

Table III
Results in the NicIcon dataset on writer independent (WI) and writer dependent (WD)

Configuration	Method	Accuracy (%)
WI	BSM (5)	75.09
	nrBSM$_\mathbf{f}$	90.18
	nrBSM$_\mathbf{f+p}$	**90.62**
WD	BSM (5)	85.97
	nrBSM$_\mathbf{f}$	93.55
	nrBSM$_\mathbf{f+p}$	**94.38**

In Table III we show results on symbol recognition over the NicIcon dataset. We see that our method considerably outperforms BSM. Moreover, note that we obtain a high accuracy in the difficult WI configuration, where the training set does not contain samples from writers that appear in the test set and vice versa. These facts reinforce the idea that the non-rigid representation that we introduce leads to a better representation of the shape, tolerant to large deformations and different writing styles. State of the art, which only exists for the on-line data, achieves 92.63% and 98.57% of accuracy rate (11) in classification with a SVM for WI and

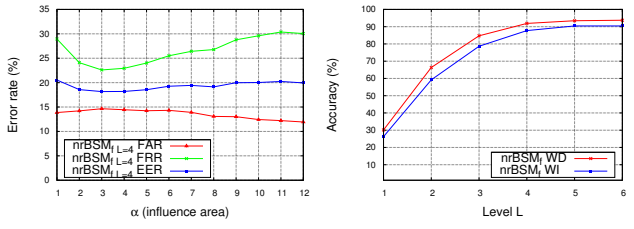
WD, respectively. Comparatively, we see that our approach is slightly below these recognition rates, but using only off-line data, which makes the problem much more difficult. Finally, note that in this case, like in random forgeries, the combination with the vector of focus position $\mathbf{p}$ entails a slightly increase in the accuracy rate.

A. Parameter selection

Our method has two parameters to be adjusted: termination level L for the region partitioning procedure, and size $h \times w$ of the *influence area* around focuses. Termination level determines the number of focuses $M = 4^L$. This level also determines the number of horizontal and vertical partitions of the image, $K = 2^L$. For example, if we set L to 3, K is equal to 8 and the number of focuses M is 64. The influence area is defined as a rectangular region where height h and width w are adjusted wrt K and the height and width of the image using following equations

$$h = \alpha * \frac{H}{K},\ w = \alpha * \frac{W}{K}. \tag{7}$$

In order to select the best α, which controls the size of the influence area, we need to reach a trade-off between the *locality* and the *globality* of the encoded information. With large influence areas, each focus captures more global information than using small influence areas. This affects directly the FAR and FRR, as we can see in Figure 4a. While EER is stable, we can use α to adjust the FAR and the FRR. Regarding the termination level L, it depends on the size of the image, and its adjustment is a compromise between accuracy and dimensionality. Experimentally, we see (Figure 4b) that accuracy becomes stable for a certain level, and deeper levels does not contribute to a significant improvement in the performance. Tables I, II and III show best results after validation of these two parameters. L is equal to 5 in both applications. Thus, the length of vectors $\mathbf{f}$ and $\mathbf{p}$ is equal to 1024 and 2048, respectively.



(a) (b)

Figure 4. Influence of (a) α in signature verification for the skilled forgeries protocol where L is set to 4, and (b) L in shape recognition.

VI. Conclusions

In this paper a novel feature extraction method for shape description and recognition is described. Experimental study, including performance evaluation and comparison with existing methods, shows the ability of our non-rigid descriptor to capture the structure of the shape and deal with large deformations, achieving efficiency and satisfactory performance. Furthermore, the region partitioning procedure provides us a hierarchical scheme where features are extracted at different levels of description. Besides the pixel density around focuses used in our method, we may consider other shape-based characteristics related to the focuses as well as to the resulting sub-regions. A combination of features extracted from different levels, through machine learning techniques, would include local and global information, and might increase the performance of our method. Our recent work is conducted in this way.

Acknowledgment

This work has been supported by the Spanish projects TIN2008-04998, TIN2009-14633-C03-03 and CONSOLIDER-INGENIO 2010(CSD2007-00018), and by a research grant of the UAB (471-01-8/09).

References

[1] D. Zhang and G. Lu, "Review of shape representation and description techniques," *PR*, vol. 37, no. 1, pp. 1–19, 2004.

[2] F. Mokhtarian and A. Mackworth, "Scale-based description and recognition of planar curves and two-dimensional shapes," *TPAMI*, vol. 8, no. 1, pp. 34–43, 1986.

[3] A. Khotanzad and Y. Hong, "Invariant image recognition by zernike moments," *TPAMI*, vol. 12, no. 5, pp. 489–497, 1990.

[4] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *TPAMI*, vol. 24, no. 4, pp. 509–522, 2002.

[5] S. Escalera, A. Fornés, O. Pujol, P. Radeva, and J. Lladós, "Blurred shape model for binary and grey-level symbol recognition," *PRL*, vol. 30, pp. 1424–1433, 2009.

[6] P. Sidiropoulos, S. Vrochidis, and I. Kompatsiaris, "Content-based binary image retrieval using the adaptative hierarchical density histogram," *PR*, vol. 44, no. 4, pp. 739–750, 2010.

[7] G. Vamvakas, B. Gatos, and S. J. Perantonis, "A novel feature extraction and classification methodology for the recognition of historical documents," in *ICDAR*, 2009, pp. 491–495.

[8] M. Blumenstein, M. A. Ferrer, and J. Vargas, "The 4nsigcomp2010 off-line signature verification competition: Scenario 2," in *ICFHR*, 2010, pp. 721–726.

[9] J. F. Vargas, M. A. Ferrer, C. M. Travieso, and J. B. Alonso, "Off-line signature verification based on grey level information using texture features," *PR*, vol. 44, no. 2, pp. 375–385, 2011.

[10] ——, "Off-line signature verification based on grey level information using wavelet transform and texture features," in *ICFHR*, 2010, pp. 587–592.

[11] D. Willems, R. Niels, M. van Gerven, and L. Vuurpijl, "Iconic and multi-stroke gesture recognition." *PR*, vol. 42, no. 12, pp. 3303–3312, 2009.

[12] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *IJCV*, vol. 73, no. 2, pp. 213 – 238, 2007.

[13] C. C. Chang and C. J. Lin. (2001) Libsvm: A library for support vector machine. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm/