# A non-rigid appearance model for shape description and recognition

Jon Almazán, Alicia Fornés, Ernest Valveny

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

**Abstract**

In this paper we describe a framework to learn a model of shape variability in a set of patterns. The framework is based on the Active Appearance Model (AAM) and permits to combine shape deformations with appearance variability. We have used two modifications of the Blurred Shape Model (BSM) descriptor as basic shape and appearance features to learn the model. These modifications permit to overcome the rigidity of the original BSM, adapting it to the deformations of the shape to be represented. We have applied this framework to representation and classification of handwritten digits and symbols. We show that results of the proposed methodology outperform the original BSM approach.

*Keywords:*
Shape recognition, Deformable models, Shape modeling, Hand-drawn recognition

## 1. Introduction

Objects can be easily interpreted by humans, and their concept can be abstracted despite colors, textures, poses or deformations. A lot of effort has been devoted for many years in order to translate this quality to computers. Thus object recognition has become one of the classic problems in Computer Vision. It is commonly divided in different sub-problems that are tackled with different techniques or from different points of view. Some of the most

*Email addresses:* almazan@cvc.uab.es (Jon Almazán), afornes@cvc.uab.es (Alicia Fornés), ernest@cvc.uab.es (Ernest Valveny)

common problems are changes in the viewpoint and the scale, in the appearance of the object and in the illumination of the scene. In this sense, different visual cues can be used to describe and identify objects. Color, texture or shape are some of them, being the last probably one of the most widely considered. Anyway, shape is not exempt from problems, and some difficulties such as noise, degradation, occlusions or deformations can be found. Therefore, shape descriptors should be capable to deal with these problems in order to guarantee intra-class compactness and inter-class separability.

Reviewing the literature, many shape descriptors, capable to deal with some of the problems, have been proposed. A survey on shape recognition can be found in (1). These descriptors can be broadly classified in two kinds of categories: statistical and structural approaches. Statistical approaches use a feature vector derived from the image to describe the shape. Several examples, using different approaches, may be found into this category. For instance, the curvature scale space (CSS) descriptor (2) uses the external contour for coding the shape. It successively blurs the image 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. Scale and rotation invariance are obtained by normalizing the image with respect to these parameters. Another well-known descriptor is Shape Context (4), which is based on the relation between shape pixels. It 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 to deal with open regions. SIFT descriptor (5) uses local information and has been mainly applied for object recognition. It selects local points of interest of the image and describes them in order to provide a “feature description” of the object. The other family of strategies corresponds to structural approaches, which are based on representing the different parts of the shape and also the relation between them using structures, such as strings, grammars or graphs that permit to describe these parts. The comparison between structures is done by means of specific techniques in each case, like graph matching or parsing (6, 7).

In our case, we are interested in shape descriptors that could be applied to Document Analysis applications, mainly in handwritten character recognition and hand-drawn symbol recognition. These are challenging applica-

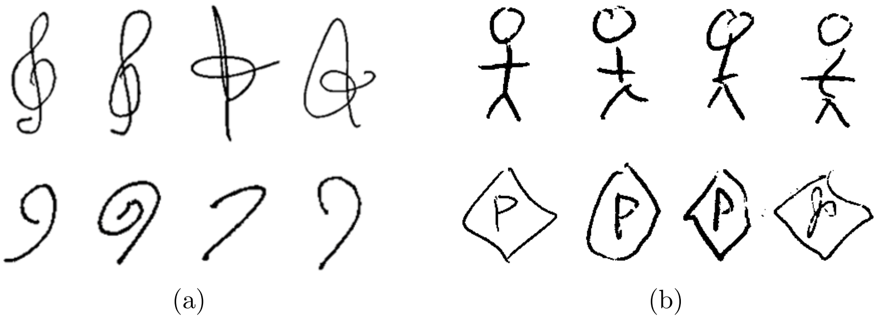

(a) (b)

Figure 1: Example of the variability caused by different writers for (a) two different music clefs and (b) two symbols from the NicIcon (10) dataset.

tions for shape descriptors in terms of intra-class compactness and inter-class separability due to the variability of handwriting. Thus, when selecting or designing a good descriptor, the particular characteristics of handwritten symbols have to be taken into account. Mainly because of many kinds of distortions, such as inaccuracy in junctions, missing parts, elastic deformations, overlapping, gaps or errors like over-tracing. Furthermore, depending on the number of writers, the variability between the symbols appearance, caused by the different writing styles, considerably increases. An example of this variability is shown in Figure 1. Nowadays, although some techniques have been applied with good results, deformations are still an open problem for descriptors. Shape descriptors mentioned above can be applied, but there are others descriptors that are specific for this domain (a survey on symbol recognition methods can be found in (8)). Among them, the Blurred Shape Model descriptor (BSM) (9), which encodes the spatial probability of appearance of the *shape* pixels and their context information, has shown good results in handwritten symbol recognition tasks. However, the tolerance to large shape deformations is still a challenging problem, and it is mainly caused because of the rigidity of the method's representation.

To deal with deformations other methods based on deformable models have been proposed for the recognition of objects and shapes. These methods can be classified in different groups depending on how the deformation procedure is performed. Some models are based on a compromise between forces: external constraint forces and internal forces influenced by the given image. This leads to an energy-minimization problem. The literature about these models is quite large, specially in the image segmentation field. They could be divided in two different sub-groups: *boundary-based* and *region-based* techniques. They are based in different approaches, such as *bottom-up*, *deformable templates*, or *level sets*, and for a better characterization we refer the reader to (11). Probably, the most known example of this first group is the Active Contours Models (ACM) (12), also known as *snakes*, from which some recent works have been based (13, 14). And another energy minimization-based approach, which consists in a non-rigid deformation process, is the *thin-plate splines* (TPS). This method has been mainly applied to image alignment and shape matching (15, 16). The second group of models (17, 18) uses a Bayesian framework in order to combine prior knowledge of the object and its deformation with the data obtained from the image. Finally, methods described by Perronin *et al.* (19), Kuo and Agazzi (20) and Keysers *et al.* (21) are non-linear image deformation models, and are based on pixel matching but applying different constraints on the deformations allowed. Independently of how deformations are modeled, there is a group of methods that try to obtain a model by analyzing the deformations of a shape that are found in the training set. Then, they use this model to match with new samples in order to obtain a distance further used in recognition tasks. The deformable part-based model (22) and the Active Appearance Models (AAM) (23) are some examples of this kind of approaches.

In this paper we propose a method for generating statistical models of shape based on the AAM (23) using an adaptation of the BSM descriptor as the basic appearance features. The BSM descriptor resulted a robust technique when classifying symbols with high variability, and it has been applied with success to problems related with hand-drawn symbols. However, due to the rigidity of its grid-based representation, it has an open problem when large deformations may cause high differences in the spatial pixel distribution. For this reason, we have proposed (24, 25) an extension of this descriptor by integrating it with a deformation model. First, we modify the BSM grid-based representation, to provide more flexibility, and make it deformable. Then, we apply a deformation procedure in order to adapt it to the shape to be described: a non-linear deformation model (24) and a *region partitioning procedure* by computing geometrical centroids (25). This new descriptor is capable to deal with large deformations due to its adaptive representation. Moreover, it allows us to extract information related to the *shape pixels* distribution and the structure of the shape.

Relating the proposed descriptors with the classification by (1), they would directly fall inside the statistical approaches category because we use a single vector to encode the description of the shape. However, as we include information about the structure of the shape (we encode the deformation occurred during the description process), the descriptor will be also structural-related. In this sense, we benefit from some advantages about including structure information without the disadvantages of the structural approaches. Finally, it is worth to mention that both non-linear deformation processes applied are very fast to compute. Other deformation methods, like those based in energy minimization functions or the Bayesian frameworks, could also be applied. Concretely, the TPS would be a perfect candidate to apply because, analog to the BSM, uses a fixed grid. However, their main disadvantage is the computational cost of the deformation to converge and adapt to the shape. For this reason, we have used non-linear deformation methods, which result in a good performance, combined with efficiency and simplicity.

Then, based on this new descriptor, the main contribution of this paper is the proposal of a non-rigid model able to learn patterns of variability. This is performed by combining the modified version of the BSM descriptor with the AAM (23) for learning the variability. It will result in a combined model that matches shape pixel distribution variations and structure variations. Moreover, this model will be integrated in two different classification schemes proposed for shape recognition tasks. Results show that the proposed methodology outperforms the original BSM approach.

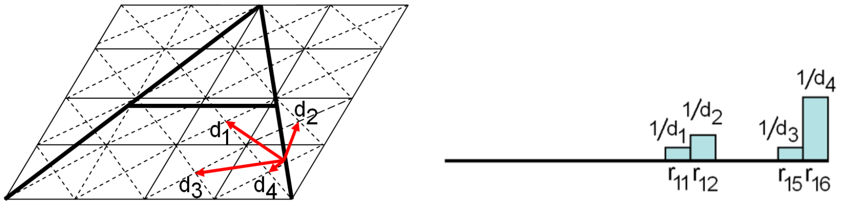
The rest of the paper is organized as follows: Section 2 is devoted to explain the new proposed descriptor, while Section 3 explains the process to build the model of appearance. The explanation of the classification schemes is conducted in Section 4. Then, performance results, as well as the comparison with the original BSM, are shown in Section 5. Finally, Section 6 concludes the paper.

## 2. Adaptive Blurred Shape Model

Our proposed deformable shape descriptor results from the adaptation of the Blurred Shape Model (BSM) (9) with a process of deformation. We use two different approaches: a non-linear deformation model (the Image Distortion Model (IDM) (21)), and a *region partitioning* procedure by geometrical centroid estimation (based on the Adaptive Hierarchical Density Histogram (AHDH) (26)) . Our objective is to encode the pixel distribution of a given image by first adapting the structure of the descriptor to the shape and then computing the pixel density measure using the BSM feature extraction procedure. Therefore, the first step is to modify the original grid-based representation of the BSM (Section 2.1) into a flexible *focus-based* representation (Section 2.2). Then, we will integrate the IDM and the *region partitioning* procedure in order to deform this new structure (Section 2.3 and Section 2.4 respectively).

### *2.1. Blurred Shape Model*

The main idea of the BSM descriptor (9) 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 from the *shape pixels* located inside its corresponding cell, but 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. 2 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 2: BSM density estimation example. (a) Distances of a given shape pixel to the neighboring centroids. (b) Vector descriptor update in regions $r$ using distances of (a).

### *2.2. Focus representation*

As it has been explained, BSM is based on placing a fixed regular grid over the image. Therefore, in order to allow deformations of the grid we must adopt a slightly different representation. Instead of a regular grid of size $k \times k$ we will place over the image a set of $k \times k$ points, equidistantly distributed. These points, denoted as *focuses*, will correspond to the centroids of the original regular grid and, as in the original approach, will accumulate votes of the neighboring pixels weighted by their distance. Concretely, the contribution of a pixel $p$ to a focus $f$ will be equal to $1/d(p, f)$, where $d$ is the euclidean distance. However, instead of defining the neighborhood as a set of fixed cells of the grid, it will be defined as an arbitrary *influence area* centered on the focus, in order to provide flexibility. The deformation of the grid will be obtained by moving independently each of the focuses along with their respective influence area. In order to limit the amount of deformation, each focus will be allowed to move only inside a pre-defined *deformation area*. In Fig. 3 we show an example of the focus representation and their influence and deformation areas. This resulting representation provides more flexibility and allows the focus deformation tracking.

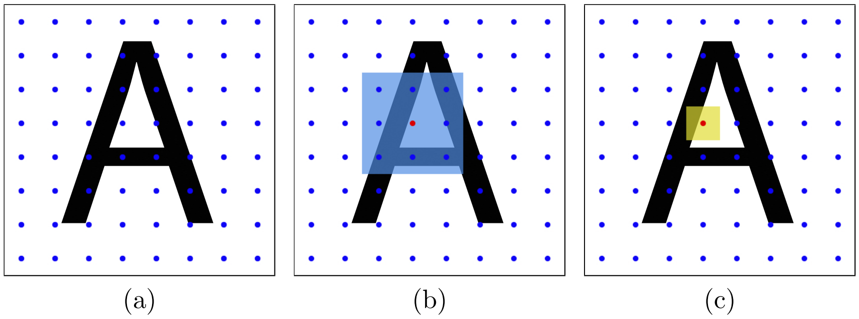


Figure 3: (a) Focuses representation. (b) Influence area. (c) Deformation area.

### *2.3. Focus deformation by non-linear deformation model (DBSM)*

Using this new representation of $k \times k$ focuses, the adaptation of the original Image Distortion Model (IDM) (21) is relatively straightforward. The ***non-linear deformation process*** of the IDM consists in: given a test and a reference images, for each pixel in the test image, determine the best matching pixel (including its context) within a region of size $w \times w$ defined around the corresponding position (i.e. the location of the *test* pixel to be matched) in the reference image. In an analog way, we will move independently every focus inside their own defined *deformation area* following a given criterion. Considering that our objective is to adapt the focuses distribution to the shape to be described, a suitable criterion will be to maximize the density, around the focus, of pixels belonging to the shape. It will be done by maximizing the accumulated value of the BSM, which is only computed from the *shape pixels* inside the *influence area*. This *influence area* moves along with the focus, so the focus will have a different value depending on its position. Thus, for a given image, every focus will be moved independently inside the *deformation area* to maximize the accumulated BSM value. In other words, the final position of each focus will be the local maxima of the density measure of *shape pixels* within its *deformation area*. Figure 4 shows an example of this process. As a result, a given shape will be represented with two output descriptors:

- A vector histogram $\mathbf{t} \in \mathbb{R}^{k^2}$ which contains the density measure of nearby pixels of each focus.
- A vector $\mathbf{s} \in \mathbb{R}^{2k^2}$, which contains $x$ and $y$ coordinates of each focus, which are normalized by the width and height of the image, respectively, in order to be scale invariant.

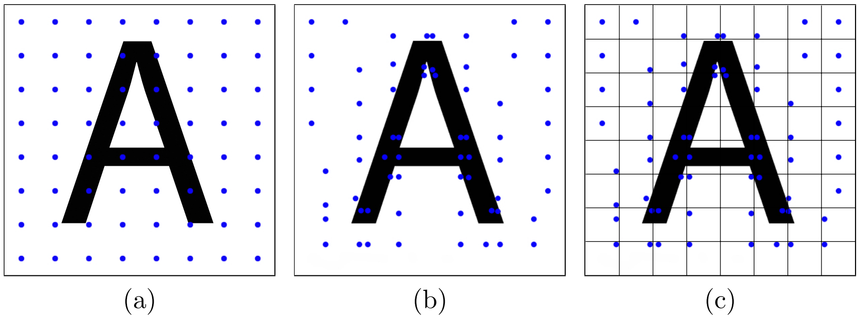

(a)
(b)
(c)

Figure 4: Example of the focuses deformation. (a) Initial position of the focuses. (b) Final position of the focuses after the maximization of their values. (c) Deformation area used.

For now on, we will name this shape descriptor, resulting from the integration of the BSM with the IDM, as the Deformable Blurred Shape Model (DBSM).

### *2.4. Focus deformation by region partitioning (nrBSM)*

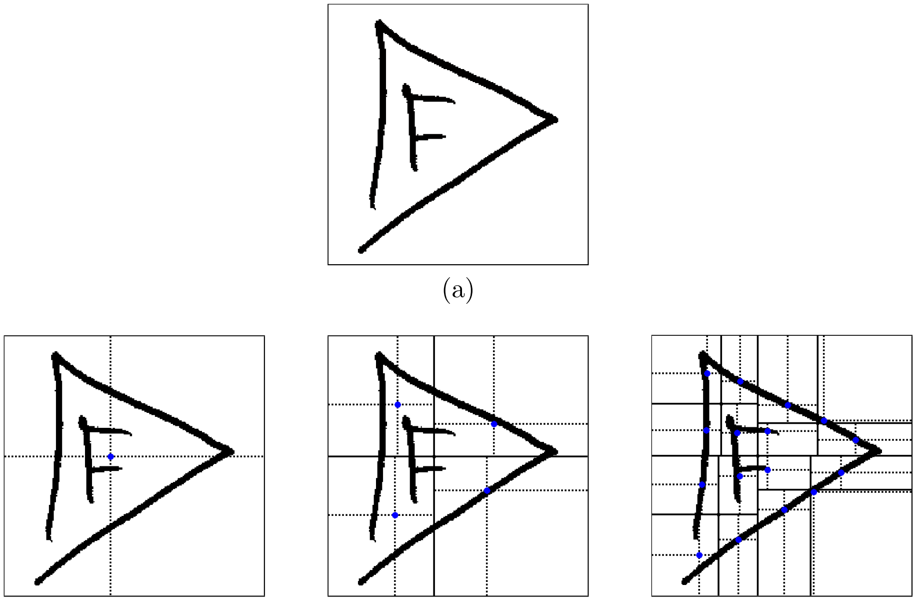
The DBSM descriptor unifies in a single procedure the deformation of the focuses and the computation of the pixel density measure around them. Now, we propose a different approach to compute these two processes in two independent steps. However, in an analog way, this new approach follows the same idea of the DBSM: focuses will be distributed over the image in regions containing a high pixel density in order to adapt them to the structure of the shape. This new approach is based on the ***region partitioning*** procedure of the Adaptive Hierarchical Density Histogram (AHDH) (26), 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 5a). 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 5b). Considering a separate cartesian coordinates system for each region $R_i^l$, the geometrical centroid $F_i^l$ is computed using equations

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

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 example of the distribution of focuses for different levels is shown in Figure 5.

Once the vector $\mathbf{s} \in \mathbb{R}^{2\times 4^L}$ containing the position of the focuses for a given shape is obtained, we compute the vector histogram $\mathbf{t} \in \mathbb{R}^{4^L}$, which contains the density measure of nearby pixels of each focus. It is done in a



(a)

(b)

(c)

(d)

Figure 5: 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.

similar way to the DBSM case: previously mentioned $h \times w$ *influence area* is used to calculate the pixel density around each focus. Focuses will receive votes from neighboring shape pixels, which are those inside this *influence area*. Based on the BSM (9), this vote is weighted according to the distance between the pixel and the focus. Finally, $x$ and $y$ coordinates of the position of the focus in vector $\mathbf{s}$ are normalized by the width and height of the image, respectively, in order to achieve scale invariance. In the following, this method will be denoted as Non-Rigid Blurred Shape Model (nrBSM).

## 3. Non-rigid Appearance Model

The deformable extensions of the Blurred Shape Model (DBSM and nrBSM) can be seen as descriptors that extract information related to the structure and the texture of a shape. The information related to the structure of the shape can be obtained from the deformation that each focus has suffered (in terms of location). And, the BSM value of the focuses, that is the pixel density measure around them, can be seen as a texture-related feature. Using these information extracted from the DBSM or the nrBSM, we will generate statistical models by learning patterns of variability from the training set, based on the Active Appearance Model (AAM) (23). It results in a model for structure variation and a model for texture variation. Moreover, after capturing the variability in *structure* and *texture* of the shape independently, we are going to generate a final *model of appearance.* This statistical model of appearance matches variations of the structure and texture simultaneously by combining their respective statistical models of variation.

### *3.1. Learning patterns of variability*

In this section we are going to detail the process of building a combined model of variation, which is based on the method developed by T.F. Cootes *et al.* (23). In order to build the model, first, they require a training set of annotated images where corresponding points have been marked on each example. However, in our case, this pre-process is automatically done with the focus-based representation of the Adaptive BSM (*i.e.*, DBSM or nrBSM): using the same number of $k \times k$ focuses for all the images in the training set we can use their own correspondence to track the variability and build the statistical model. Moreover, the Adaptive BSM also extracts both kind of features necessary to build the combined model.

Once the Adaptive BSM is computed in all the images of the training set, we obtain two output vectors for every image: a vector $\mathbf{s} \in \mathbb{R}^{2k^2}$ containing the final coordinates of the focuses, and a vector $\mathbf{t} \in \mathbb{R}^{k^2}$ containing the density measure of pixels around each focus. With these two vectors we are going to build two different statistical models by learning the variability of the deformations in the focuses positions (related to the structure of the shape) and the variability in the pixel density (related to the texture). This is done by first constructing two different matrices with $\mathbf{s}$ and $\mathbf{t}$ vectors and applying principal component analysis (PCA) to both matrices, resulting in a *structure model* and a *texture model.* A property of these models is that they make possible the reconstruction of the shape and texture information of the training images using

$$\begin{aligned} \mathbf{s} &= \overline{\mathbf{s}} + \mathbf{Q_s b_s} \\ \mathbf{t} &= \overline{\mathbf{t}} + \mathbf{Q_t b_t}, \end{aligned} \tag{2}$$

where $\overline{\mathbf{s}}$ is the mean *structure*, $\overline{\mathbf{t}}$ the mean *texture information*, $\mathbf{Q_s}$, $\mathbf{Q_t}$ are the matrices of eigenvectors that describe the modes of variation derived from the training set, and $\mathbf{b_s}$, $\mathbf{b_t}$ are the vectors of weights that represent *structure* and *texture*, respectively. Vectors $\mathbf{b_s}$ and $\mathbf{b_t}$ can be seen as the parameters of the model, or the representation of descriptors $\mathbf{s}$ and $\mathbf{t}$ in the PCA space.

Using this property, the following step consists in learning correlations between *structure* and *texture* using their respective models. We obtain the representation in the PCA space of structure and texture of all training images using

$$\begin{aligned} \mathbf{b_s} &= \mathbf{Q_s}^T(\mathbf{s} - \overline{\mathbf{s}}) \\ \mathbf{b_t} &= \mathbf{Q_t}^T(\mathbf{t} - \overline{\mathbf{t}}). \end{aligned} \tag{3}$$

The result is that every image in the training set is represented by a vector containing *structure* model parameters and a vector containing *texture* model parameters ($\mathbf{b_s}$ and $\mathbf{b_t}$ weight vectors). Then, the final step consists in concatenating both vectors of every sample image in a single vector $\mathbf{c}$, construct a new matrix, and apply PCA again, extracting the combined modes of variation. In order to give *structure* and *texture* variation approximately equal significance, before applying PCA we scale *structure* parameters so their cumulative variance within the training set is equal to the cumulative variance of the *texture* parameters. The resulting appearance model has parameters, $\mathbf{b_a}$, controlling *structure* and *texture* models parameters (which control *structure* and *texture* descriptions) according to

$$\mathbf{a} = \overline{\mathbf{a}} + \mathbf{Q_a}\mathbf{b_a}, \tag{4}$$

where $\mathbf{a}$ is the concatenation of *structure* and *texture* models parameters, $\overline{\mathbf{a}}$ is the mean appearance, and $\mathbf{Q_a}$ is the matrix of eigenvectors that describes the modes of variation of the appearance. Finally, the vector of $\mathbf{b_a}$ can be seen as the feature vector that represents an image in the combined *model of appearance*. Given a model of appearance, it can be computed using

$$\mathbf{b_a} = \mathbf{Q_a}^T(\mathbf{a} - \overline{\mathbf{a}}). \tag{5}$$

## 4. Classification

The Non-Rigid Appearance Model (NRAM) generates statistical models of appearance, which combines *structure* and *texture* variations learned from a training set. Therefore, we can generate a model that represents independently every different class in the dataset. We have designed two different classification schemes using the Non-Rigid Appearance Model for shape recognition tasks. On one hand, a scheme based on the ability of the appearance model to generate "synthetic" representations of a given shape. On the other hand, a scheme using the parameters of the model, *i.e.*, the descriptors represented in the PCA space, to train a Support Vector Machine (SVM) for each class.

### *4.1. Distance to the model*

The representation of the shape obtained with the model in the PCA space can be used to obtain a reconstruction of the shape in the original space. This reconstruction will reflect the utility of the model to represent the shape. It is expected that for shapes belonging to the class it will be similar. Therefore, we can use this property to, given a new image and its respective *structure* and *shape* feature vectors, generate a synthetic sample with a model of a given class that matches it as closely as possible and design a measure of similarity. We have integrated it into a matching process for shape classification. It consists in, given an image $I$ and an appearance model $M$, first computing the structure $\mathbf{s_I}$ and texture $\mathbf{t_I}$ descriptors of that image. Then we approximate these descriptors to the corresponding parameters of the structure and texture model of $M$ using the expressions in Equation 3, resulting in two new vectors $\mathbf{b_{s_I}}$ and $\mathbf{b_{t_I}}$. Following, we concatenate them using the normalization learned in the training step to make equal both contributions. And then we approximate again this new vector $\mathbf{a_I}$ to the model of combined appearance in $M$ using Equation 5.

Finally, the resulting parameters of the appearance model, $\mathbf{b_{a_I}}$, are split in $\mathbf{b_{s_J}}$ and $\mathbf{b_{t_J}}$ in order to generate the new synthetic descriptors $\mathbf{s_J}$ and $\mathbf{t_J}$ by back-projecting them with the models of structure and texture, respectively. These are the descriptors that best match to $I$ according to the appearance model $M$. Thus, we can use this new descriptors of structure, $\mathbf{s_J}$, and texture, $\mathbf{t_J}$, to compute a distance with the descriptors of the original image $I$. For this purpose, we use the euclidean distance between each corresponding vector. Furthermore, we want to add some information about the necessary deformation applied to adjust the model to the test image. For that end, we also compute the euclidean distance between the generated descriptors and the mean values of the model, both for structure $\overline{\mathbf{s}}$ and texture $\overline{\mathbf{t}}$. Then, the final distances are

$$\begin{aligned} d_s &= dist(\mathbf{s_I}, \mathbf{s_J}) + \beta \cdot dist(\mathbf{s_J}, \bar{\mathbf{s}}) \\ d_t &= dist(\mathbf{t_I}, \mathbf{t_J}) + \beta \cdot dist(\mathbf{t_J}, \bar{\mathbf{t}}) \end{aligned} \tag{6}$$

where $\beta$ is the factor that weights the contribution of the information of deformation in the final distance, and $d_s$ and $d_t$ structure and texture distances, respectively. Finally, the structure and texture measures of similarity are combined using $\theta$ as another factor of contribution, resulting in the following expression

$$d_a = d_s \cdot \theta + d_t \cdot (1 - \theta) \tag{7}$$

This distance can be used for classification tasks, being applied to, for example, a nearest neighbor classifier: given a test image, and a set of models representing shape classes, we assign the image to the class which results in the minimum distance from the representation synthetically generated.

### *4.2. Support Vector Machine-based scheme*

The second scheme we propose uses the representation in the space of the appearance model space, *i.e.*, the vector of weights $\mathbf{b_a}$. It is used as a feature vector to describe the shapes contained in the dataset to train a different Support Vector Machine for each class.

In the training step (Figure 6) we compute first *structure* and *texture* descriptors, $\mathbf{s}$ and $\mathbf{t}$, and we generate a Non-Rigid Appearance Model $M_i$ for each one of the $n$ classes in the dataset using the procedure explained in Section 3.1. Then, we train a binary Support Vector Machine for each class. This is, for class $i$, we use as positive samples those training samples belonging to class $i$, projected in the appearance model space of the model $M_i$. And as negative samples the rest of the training set (*i.e.*, those which do not belong to class $i$) also projected with model $M_i$.

Then, given a test sample, it is projected in the PCA space of all the appearance models of all the classes. And then, the score is computed with all the SVMs, using their corresponding vector. Each score is normalized (27) by subtracting the mean and then divided by the average score norm computed for each SVM. Finally, the test sample is assigned to the class which results in the highest score. A scheme of the process is shown in Figure 7.

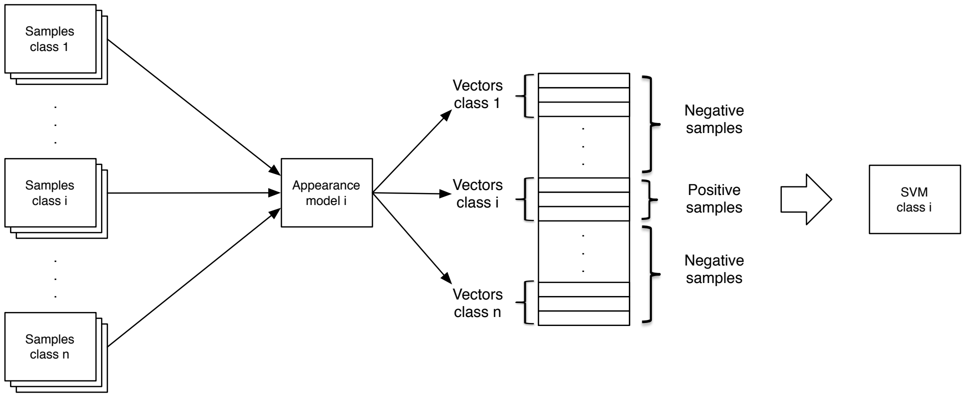


Figure 6: Training procedure for the SVM-based scheme.

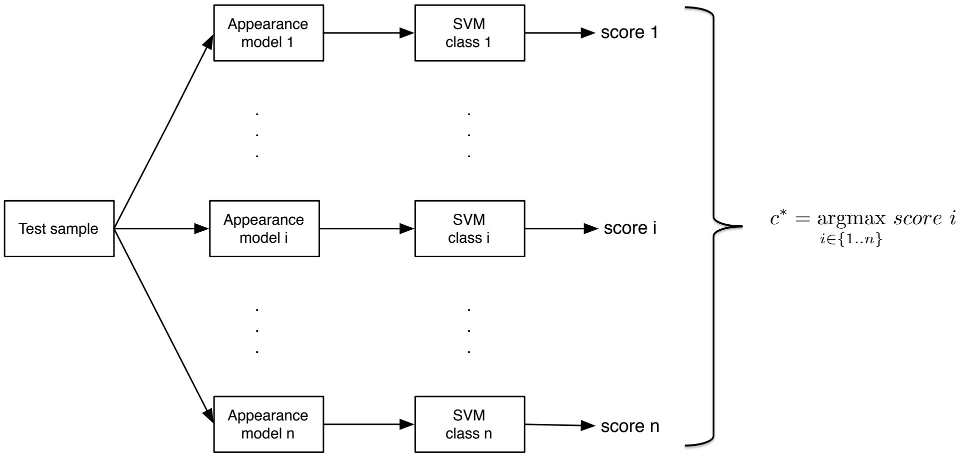


Figure 7: Test procedure for the SVM-based scheme.

## 5. Experiments

In this section we are going to show the performance of the proposed Non-Rigid Appearance Model for shape recognition tasks using two different datasets.

### *5.1. Datasets*

We have tested our methods for shape recognition tasks, and for this purpose, we have used the MNIST and NicIcon dataset. In both cases, we have used all the elements of the training and test sets, and all the elements of the validation set in the case of the NicIcon. Following, we describe these datasets as well as the experimental protocol used in each one.

**MNIST.** The MNIST (28) (Figure 8) is a database of handwritten digits from different writers and it is divided in a training set of $60,000$ examples, and a test set of $10,000$ examples. The digit size is normalized and centered in a fixed-size image of $28 \times 28$ pixels. We have re-centered the digits by their bounding box, as it is reported in (28) to improve error rates when classification methods like SVM or K-nearest neighbors are applied. This dataset has been commonly used in learning techniques and pattern recognition methods.

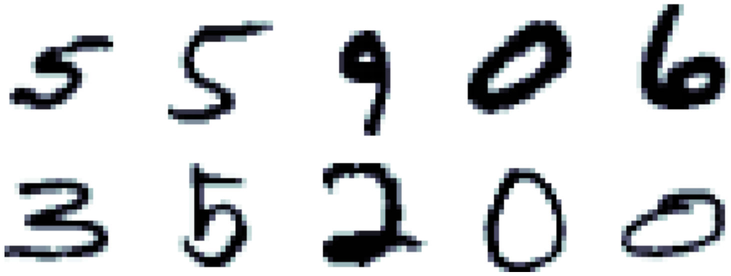


Figure 8: Digit samples of MNIST dataset.

**NicIcon.** The NicIcon dataset (10) (Figure 9) is composed of 26,163 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 already divided in three subsets (training, validation and test) for both *writer dependent* and *independent* settings. Approximately, and depending on the setting, $9,300$, $6,200$ and $10,700$ symbols are contained in the training, validation and test sets, respectively. We have selected the off-line data with both configurations as a benchmark to test our method. It is worth to mention that off-line data is presented as scanned forms where writers where said to draw the symbols. So first, we have extracted individually every symbol from the scanned forms, and then binarized and scale-normalized in an image of $256 \times 256$ pixels.

### *5.2. Results*

We now show the benefits of the proposed method using the datasets introduced in Section 5.1. We test our Non-Rigid Appearance Model (NRAM) (Section 3) by learning the variability using as basic features both Adaptive

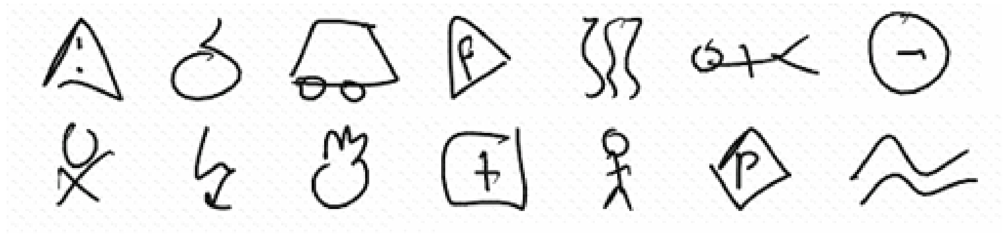


Figure 9: Samples of the 14 different classes of the NicIcon dataset.

Blurred Shape Model descriptors proposed, the **DBSM** and the **nrBSM** (Section 2.3 and Section 2.4 respectively). It results in two different configurations: NRAM+DBSM and NRAM+nrBSM. We apply them for shape recognition tasks using both classification schemes proposed in Section 4.

Table 1: Accuracy rate (%) comparison of the Non-Rigid Appearance Model (NRAM) (in combination with the DBSM and the nrBSM) with the original BSM and the DBSM and nrBSM, using a NN classifier.

| Method | | BSM (9) | DBSM | nrBSM | NRAM+DBSM | NRAM+nrBSM |
|---|---|---|---|---|---|---|
| MNIST | | 92.65 | 94.39 | 94.78 | 89.29 | 94.65 |
| NicIcon | WD | 93.73 | 95.45 | 95.14 | 90.88 | 97.70 |
| | WI | 90.02 | 90.29 | 91.09 | 86.35 | 95.18 |

First, we report in Table 1 results over the nearest neighbor classifier using the distance to the model (Section 4.1). We compare the performance with the original approach, the BSM (9), and also with the DBSM and nrBSM descriptor without applying the Non-Rigid Appearance Model. We can appreciate that, while the combination of the NRAM with the DBSM (NRAM+DBSM) features results in a lower performance, the appearance models obtained with the nrBSM features (NRAM+nrBSM) outperform the rest of approaches. These results lead us to conclude that the NRAM methodology is not able to learn the variation models in a suitable way when structure and texture features are extracted using the DBSM. However, the validity of the model is shown when we use features extracted from the nrBSM, where accuracy increase considerably compared to the situation where we do not apply the NRAM. This difference in performance is due to the deformation procedure. Analyzing the focuses distribution, we can appreciate that, in the case of the nrBSM, focuses distribute along the whole shape, which is contrary to the DBSM case. This is because the nrBSM is not limited by a pre-defined initial position of the focuses or a fixed *deformation area*, while the DBSM adaptability is affected by both factors. Thus, small deformation areas lead the focuses to stay close to their initial position, while large areas make that all the focuses converge to the same location. Therefore, the better adaptability of the nrBSM makes it more suitable to learn the variability, and the variation models obtained with these features are more representative. The NRAM benefits from this fact, and results in a better performance compared to the DBSM features. Finally, note that, in all the cases, DBSM and nrBSM descriptors outperform the original BSM. This shows that the integration of deformations to the fixed grid-based representation leads to a better performance when large shape distortions are present.

Regarding the SVM classification scheme, the results are shown in Table 2. We can appreciate that performance increases for both descriptors, being remarkable in the cases of the MNIST and *writer dependent* NicIcon datasets. Note that results obtained over the MNIST do not reach the state of the art (28). This is mainly due to the fact that the original BSM descriptor has not been specifically designed for the task of handwritten character recognition. However, note also that results with the proposed model are much better than results with the original BSM descriptor. Thus, it is expected that combining the NRAM methodology with a specific feature extraction method for character could result in a competitive performance, comparable to the current state of the art.

Table 2: Results of the Non-Rigid Appearance Model combined with the DBSM and the nrBSM using the SVM classification scheme.

| Method | | NRAM+DBSM | NRAM+nrBSM |
|---|---|---|---|
| MNIST | | 97.76 | 97.50 |
| NicIcon | WD | 96.52 | 97.35 |
| | WI | 93.26 | 94.29 |

Concerning the NicIcon dataset, the state of the art, which only exists for on-line data, achieves 92.63% and 98.57% of accuracy rate in classification, using a SVM, for WI and WD, respectively (10) . Comparatively, we see that the recognition rate of our approach is slightly below in the case of WD, but higher for the WI configuration. Furthermore, we only use off-line data, which makes the problem much more difficult. Thus, we can consider the obtained results very competitive. It is also remarkable the significant increase of performance in relation to the original BSM approach. 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 NRAM combined with the nrBSM representation leads to a good representation of the shape, tolerant to large variations and different writing styles.

### *5.3. Parameter selection*

Our Adaptive BSM descriptors (*i.e.* DBSM and nrBSM) have two parameters (leaving aside the *deformation area* of the DBSM ) to be adjusted: the number of $k \times k$ focuses (defined by termination level $L$ for the nrBSM), and the $h \times w$ size of the *influence area*. 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

$$h = \alpha * \frac{H}{k}, \;\; w = \alpha * \frac{W}{k}. \tag{8}$$

In order to select the best $\alpha$, 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. Experimentally, we appreciate that using small influence areas performs better in the combination of the NRAM with the nrBSM descriptor. This is due to the fact that focuses are well distributed over the whole shape, and we can analyze the pixel distribution variability locally for each region of the shape. The best performance for both datasets has been obtained for values of $\alpha$ around 1. Regarding the number of focuses $k$, it depends on the size of the image, and its adjustment is a compromise between performance and dimensionality. Experimentally, we see that accuracy becomes stable for a certain number, and a higher number of focuses does not contribute to a significant improvement in the performance. We have set $k$ equal to 16 for the MNIST dataset, and equal to 32 for the NicIcon dataset.

## 6. Conclusions

In this paper, a method for modeling the appearance deformations of a shape by learning the variability of the training set, the Non-Rigid Appearance Model is described. It is developed on the top of two recently introduced adaptive shape descriptors based on the BSM. These descriptors are used as appearance features to build the statistical model. Then we describe two classification schemes to integrate the appearance models in shape recognition tasks. The experimental performance evaluation shows the ability of the appearance models to learn *structure* and *texture* variability, achieving a satisfactory performance in shape recognition. Additionally, results also show the capacity of both novel Adaptive Blurred Shape Model descriptors to capture the structure of the shape and deal with large deformations, outperforming the rigid grid-based approach of the original BSM. In this work, the BSM has been used as the basic descriptor to build the model. However, the non-rigid appearance model introduces a perfect framework that can be used with a large number of different appearance features, which may be selected depending on the application. Moreover, the methodology can be easily extended for a larger number of models, where the combination can be done at different levels or following different criteria.

## 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, G. Lu, Review of shape representation and description techniques, Pattern Recognition 37 (2004) 1–19.

[2] F. Mokhtarian, A. Mackworth, Scale-based description and recognition of planar curves and two-dimensional shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (1986) 34–43.

[3] A. Khotanzad, Y. Hong, Invariant image recognition by zernike moments, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (1990) 489–497.

[4] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (2002) 509–522.

[5] D. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2004) 91–110.

[6] H. Bunke, Attributed programmed graph grammars and their application to schematic diagram interpretation, IEEE Transactions on Pattern Analysis and Machine Intelligence 1 (1982) 574–582.

[7] J. LLadós, E. Martí, J. Villanueva, Symbol recognition by error-tolerant subgraph matching between region adjacency graphs, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2001) 1137–1143.

[8] J. LLadós, E. Valveny, G. Sánchez, E. Martí, Symbol recognition: Current advances and perspectives, Lecture Notes in Computer Science 2390 (2002) 104–127.

[9] S. Escalera, A. Fornés, O. Pujol, P. Radeva, J. Lladós, Blurred shape model for binary and grey-level symbol recognition, Pattern Recognition Letters 30 (2009) 1424–1433.

[10] D. Willems, R. Niels, M. van Gerven, L. Vuurpijl, Iconic and multistroke gesture recognition., Pattern Recognition 42 (2009) 3303–3312.

[11] N. Paragios, R. Deriche, Geodesic active regions and level set methods for supervided texture segmentation, International Journal of Computer Vision 46 (2002) 223–247.

[12] M. Kass, A. Witkin, D. Terzopoulos, Snakes: Active contour models, International Journal of Computer Vision 1 (1988) 321–331.

[13] J. C. Nascimento, J. S. Marques, Adaptive snakes using the em algorithm, IEEE Transactions on Image Processing 14 (2005) 1678–1686.

[14] A. K. Mishra, P. W. Fieguth, D. A. Clausi, Decoupled active contour (dac) for boundary detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (2011) 310–324.

[15] K. Rohr, H. S. Stiehl, R. Sprengel, T. M. Buzug, J. Weese, M. H. Kuhn, Landmark-based elastic registration using approximating thin-plate splines, IEEE Transactions on Medical Imaging 20 (2001) 526–534.

[16] H. Chui, A. Rangarajan, A new point matching algorithm for non-rigid registration, Computer Vision and Image Understanding 89 (2003) 114–141.

[17] K. Cheung, D. Yeung, R. Chin, A bayesian framework for deformable pattern recognition with application to handwritten character recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (1998) 1382–1387.

[18] S. Barrat, S. Tabbone, A bayesian network for combining descriptors: application to symbol recognition, International Journal on Document Analysis and Recognition 13 (2010) 65–75.

[19] F. Perronin, J. L. Dugelay, K. Rose, Iterative decoding of two-dimensional hidden markov models, in: International Conference on Acoustics, Speech and Signal Processing, 2003, volume 3, pp. 329–332.

[20] S. Kuo, O. Agazzi, Keyword spotting in poorly printed documents using pseudo 2-d hidden markov models, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (1994) 842–848.

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

[22] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (2010) 1627–1645.

[23] T. Cootes, G. Edwards, C. Taylor, Active appearance models, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2001) 681–685.

[24] J. Almazán, E. Valveny, A. Fornés, Deforming the blurred shape model for shape description and recognition, in: Pattern Recognition and Image Analysis, Lecture Notes in Computer Science, 2011, volume 6669, pp. 1–8.

[25] J. Almazán, A. Fornés, E. Valveny, A non-rigid feature extraction method for shape recognition, in: International Conference on Document Analysis and Recognition, 2011, pp. 987–991.

[26] P. Sidiropoulos, S. Vrochidis, I. Kompatsiaris, Content-based binary image retrieval using the adaptative hierarchical density histogram, Pattern Recognition 44 (2010) 739–750.

[27] M. Douze, A. Ramisa, C. Schmid, Combining attributes and fisher vectors for efficient image retrieval, in: IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 745–752.

[28] Y. Lecun, C. Cortes, The mnist database of handwritten digits., `http://yann.lecun.com/exdb/mnist/`.