ARABIC HANDWRITTEN CHARACTER RECOGNITION USING STRUCTURAL SHAPE DECOMPOSITION

Abdullah A. Al-Shaher[1] and Edwin R. Hancock[2]

[1]Department of Computer and Information Systems, College of Business Studies,
Public Authority for Applied Education and Training, Kuwait
[2]Department of Computer Science University of York, York, United Kingdom

ABSTRACT

This paper presents a statistical framework for recognising 2D shapes which are represented as an arrangement of curves or strokes. The approach is a hierarchical one which mixes geometric and symbolic information in a three-layer architecture. Each curve primitive is represented using a point-distribution model which describes how its shape varies over a set of training data. We assign stroke labels to the primitives and these indicate to which class they belong. Shapes are decomposed into an arrangement of primitives and the global shape representation has two components. The first of these is a second point distribution model that is used to represent the geometric arrangement of the curve centre-points. The second component is a string of stroke labels that represents the symbolic arrangement of strokes. Hence each shape can be represented by a set of centre-point deformation parameters and a dictionary of permissible stroke label configurations. The hierarchy is a two-level architecture in which the curve models reside at the nonterminal lower level of the tree. The top level represents the curve arrangements allowed by the dictionary of permissible stroke combinations. The aim in recognition is to minimise the cross entropy between the probability distributions for geometric alignment errors and curve label errors. We show how the stroke parameters, shape-alignment parameters and stroke labels may be recovered by applying the expectation maximization EM algorithm to the utility measure. We apply the resulting shape-recognition method to Arabic character recognition.

KEYWORDS

point distribution models, expectation maximization algorithm, discrete relaxation, hierarchical mixture of experts, Arabic scripts, handwritten characters

1. INTRODUCTION

The analysis and recognition of curved shapes has attracted considerable attention in the computer vision literature. Current work is nicely exemplified by point distribution models [1] and shape-contexts [2]. However, both of these methods are based on global shape-descriptors. This is potentially a limitation since a new model must be acquired for each class of shape and this is an inefficient process. An alternative and potentially more flexible route is to use a structural approach to the problem, in which shapes are decomposed into arrangements of primitives. This idea was central to the work of Marr [3]. Shape learning may then be decomposed into a two-stage process. The first stage is to acquire a models of the variability is the distinct primitives. The second stage is to learn the arrangements of the primitives corresponding to different shape classes.

DOI : 10.5121/sipij.2017.8201

Although this structural approach has found widespread use in the character recognition community, it has not proved popular in the computer vision domain. The reason for this is that the segmentation of shapes into stable primitives has proved to be an elusive problem. Hence, there is a considerable literature on curve polygonalisation, segmentation and grouping. However, one of the reasons that the structural approach has proved fragile is that it has been approached using geometric rather than statistical methods. Hence, the models learned and the recognition results obtained are highly sensitive to segmentation error. In order to overcome these problems, in this paper we explore the use of probabilistic framework for recognition.

We focus on the problem of developing hierarchical shape models for handwritten Arabic characters. These characters are decomposed into concave or convex strokes. Our statistical learning architecture is reminiscent of the hierarchical mixture of experts algorithm. This is a variant of the expectation maximisation algorithm, which can deal with hierarchically structured models. The method was first introduced in 1994 by Jordan and Jacobs [4]. In its simplest form the method models data using a doubly nested mixture model. At the top layer the mixture is over a set of distinct object classes. This is sometimes referred to as the gating layer. At the lower level, the objects are represented as a mixture over object subclasses. These sub-classes feed into the gating later with predetermined weights. The parameters of the architecture reside in the sublayer, which is frequently represented using a Gaussian mixture model. The hierarchical mixture of experts algorithm provides a means of learning both the gating weights and the parameters of the Gaussian mixture model.

Here our structural decomposition of 2D shapes is a hierarchical one which mixes geometric and symbolic information. The hierarchy has a two-layer architecture. At the bottom layer we have strokes or curve primitives. These fall into different classes. For each class the curve primitive is represented using a point-distribution model [5] which describes how its shape varies over a set of training data. We assign stroke labels to the primitives to distinguish their class identity. At the top level of the hierarchy, shapes are represented as an arrangement of primitives. The representation of the arrangement of primitives has two components. The first of these is a second point distribution model that is used to represent how the arrangement of the primitive centre-points varies over the training data. The second component is a dictionary of configurations of stroke labels that represents the arrangements of strokes at a symbolic level. Recognition hence involves assigning stroke symbols to curves primitives, and recovering both stroke and shape deformation parameters. We present a probabilistic framework which can be for the purposes of shape-recognition in the hierarchy. We apply the resulting shape-recognition method to Arabic character recognition.

2. SHAPE REPRESENTATION

We are concerned with recognising a shape $W = \{\vec{w}_1,...,\vec{w}_p\}$ which consists of a set of p ordered but unlabelled landmark points with 2D co-ordinate vectors $\vec{w}_1$, ..., $\vec{w}_p$. The shape is assumed to be segmented into a set of K non-overlapping strokes. Each stroke consists of a set of consecutive landmark points. The set of points belonging to the stroke indexed k is S_k. For each stroke, we compute the mean position

$$\vec{c}_k = \frac{1}{|S_k|}\sum_{i\in S_k}\vec{w}_i \qquad (1)$$

The geometry of stroke arrangement is captured by the set of mean position vectors $C = \{\vec{c}_1,....\vec{c}_K\}$.

Our hierarchical model of the characters uses both geometric and symbolic representations of the shapes. The models are constructed from training data. Each training pattern consists of a set of landmark points that are segmented into strokes. We commence by specifying the symbolic components of the representation. Each training pattern is assigned to shape class and each component stroke is assigned to stroke class. The set of shape-labels is Ω_c and the set of stroke labels is Ω_s. The symbolic structure of each shape is represented a permissible arrangement of stroke-labels. For shapes of class $\omega \in \Omega_c$ the permissible arrangement of strokes is denoted by

$$\Lambda_\omega = < \lambda_1^\omega, \lambda_2^\omega, ... > \qquad (2)$$

We model the geometry of the strokes and stroke-centre arrangements using point distribution models. To capture the shape variations, we use training data. The data consists of a set of shapes which have been segmented into strokes. Let the t^{th} training pattern consist of the set of p landmark co-ordinate vectors $X^t = \{\vec{x}_1, ..., \vec{x}_p\}$. Each training pattern is segmented into strokes. For the training pattern indexed t there are K_t strokes and the index set of the points belonging to the k^{th} stroke is S_k^t. To construct the point distribution model for the strokes and stroke-centre arrangements, we convert the point co-ordinates into long-vectors. For the training pattern indexed t, the long-vector of stroke centres is $X_t = ((\vec{c}_1^{\,t})^T, (\vec{c}_2^{\,t})^T, ..., (\vec{c}_L^{\,t})^T)^T$. Similarly for the stroke indexed k in the training pattern indexed t, the long-vector of co-ordinates is denoted by $z_{t,k}$. For examples shapes belonging to the class ω, to construct the stroke-centre point distribution model we need to first compute the mean long vector

$$Y_\omega = \frac{1}{|T_\omega|} \sum_{t \in T_\omega} X_t \qquad (3)$$

where T_ω is the set of index patterns and the associated covariance matrix

$$\Sigma_\omega = \frac{1}{|T_\omega|} \sum_{t \in T_\omega} (X_t - Y_\omega)(X_t - Y_\omega)^T \qquad (4)$$

The eigenmodes of the stroke-centre covariance matrix are used to construct the point-distribution model. First, the eigenvalues e of the stroke covariance matrix are found by solving the eigenvalue equation $| \Sigma_\omega - e^\omega I | = 0$ where I is the $2L \times 2L$ identity matrix. The eigen-vector ϕ_i corresponding to the eigenvalue e_i^ω is found by solving the eigenvector equation $\Sigma \phi_i^\omega = e_i^\omega \phi_i^\omega$. According to Cootes and Taylor [6], the landmark points are allowed to undergo displacements relative to the mean-shape in directions defined by the eigenvectors of the covariance matrix Σ_ω. To compute the set of possible displacement directions, the M most significant eigenvectors are ordered according to the magnitudes of their corresponding eigenvalues to form the matrix of column-vectors $\Phi_\omega = (\phi_1^\omega | \phi_2^\omega | ... | \phi_M^\omega)$, where $e_1^\omega, e_2^\omega, ..., e_M^\omega$ is the order of the magnitudes of the eigenvectors. The landmark points are allowed to move in a direction which is a linear combination of the eigenvectors. The updated landmark positions are given by $\hat{X} = Y_\omega + \Phi_\omega \gamma_\omega$, where γ_ω is a vector of modal co-efficients. This vector represents the free-parameters of the global shape-model.

This procedure may be repeated to construct a point distribution model for each stroke class. The set of long vectors for strokes of class λ is $T_\lambda = \{Z_{t,k} \mid \lambda_k^t = \lambda\}$. The mean and covariance matrix for this set of long-vectors are denoted by Y_λ and Σ_λ and the associated modal matrix is Φ_λ. The point distribution model for the stroke landmark points is $\hat{Z} = Y_\lambda + \Phi_\lambda \gamma_\lambda$.

3. LEARNING MIXTURES OF PDM'S

In Cootes and Taylor's method [5], learning involves extracting a single covariance matrix from the sets of landmark points. Hence, the method can only reproduce variations in shape which can be represented as linear deformations of the point positions. To reproduce more complex variations in shape either a non-linear deformation or a series of local piecewise linear deformations must be employed.

Here we adopt an approach based on mixtures of point-distributions [7]. Our reasons for adopting this approach are twofold. First, we would like to be able to model more complex deformations by using multiple modes of shape deformation. The need to do this may arise in a number of situations. The first of these is when the set of training patterns contains examples from different classes of shape. In other words, we are confronted with an unsupervised learning problem and need to estimate both the mean shape and the modes of variation for each class of object. The second situation is where the shape variations in the training data can not be captured by a single covariance matrix, and a mixture is required.

Our approach is based on fitting a Gaussian mixture model to the set of training examples. We commence by assuming that the individual examples in the training set are conditionally independent of one-another. We further assume that the training data can be represented by a set of shape-classes Ω. Each shape-class ω has its own mean point-pattern Y_ω and covariance matrix Σ_ω. With these ingredients, the likelihood function for the set of training patterns is

$$p(X_t, t = 1,...T) = \prod_{t=1}^{T} \sum_{\omega \in \Omega} p(X_t \mid Y_\omega, \Sigma_\omega) \tag{5}$$

where $p(X_t \mid Y_\omega, \Sigma_\omega)$ is the probability distribution for drawing the training pattern X_t from the shape-class ω.

According to the EM algorithm, we can maximise the likelihood function above, by adopting a two-step iterative process. The process revolves around the expected log-likelihood function

$$Q_L(C^{(n+1)} \mid C^{(n)}) = \sum_{t=1}^{T} \sum_{\omega \in \Omega} P(t \in \omega \mid X_t, Y_\omega^{(n)}, \Sigma_\omega^{(n)}) \ln P(X_t \mid Y_\omega^{(n+1)}, \Sigma_\omega^{(n+1)}) \tag{6}$$

where $Y_\omega^{(n)}$ and $\Sigma_\omega^{(n)}$ are the estimates of the mean pattern-vector and the covariance matrix for class ω at iteration n of the algorithm. The quantity$P(t \in \omega | X_t, Y_\omega^{(n)}, \Sigma_\omega^{(n)})$ is the a posteriori probability that the training pattern X_t belongs to the class ω at iteration n of the algorithm. The probability density for the pattern-vectors associated with the shape-class ω, specified by the estimates of the mean and covariance at iteration n+1 is$P(X_t | Y_\omega^{(n+1)}, \Sigma_\omega^{(n+1)})$. In the M, or maximisation, step of the algorithm the aim is to find revised estimates of the mean pattern-vector and covariance matrix which maximise the expected log-likelihood function. The update equations depend on the adopted model for the class-conditional probability distributions for the pattern-vectors.

In the E, or expectation, step the a posteriori class membership probabilities are updated. This is done by applying the Bayes formula to the class-conditional density. At iteration n+1, the revised estimate is

$$P\left(t\epsilon\omega \middle| X_t, Y_\omega^{(n)}, \Sigma_\omega^{(n)}\right) = \frac{P\left(X_t \middle| Y_\omega^{(n)}, \Sigma_\omega^{(n)}\right)\pi_\omega^{(n)}}{\sum_{\omega\in\Omega} P(X_t|Y_\omega^{(n)}, \Sigma_\omega^{(n)})\pi_\omega^{(n)}} \tag{7}$$

where

$$\pi_\omega^{(n+1)} = \frac{1}{T}\sum_{t=1}^{T} P(t\in \omega | X_t, Y_\omega^{(n)}, \Sigma_\omega^{(n)}) \tag{8}$$

3.1 Mixtures of Gaussians

We now consider the case when the class conditional density for the training patterns is Gaussian. Here we assume that the pattern vectors are distributed according to the distribution

$$P(X_t | Y_\omega^{(n)}, \Sigma_\omega^{(n)}) = \frac{1}{(2\pi)^L\sqrt{|\Sigma_\omega^{(n)}|}}\exp\left[-\frac{1}{2}(X_t - Y_\omega^{(n)})^T(\Sigma_\omega^{(n)})^{-1}(X_t - Y_\omega^{(n)})\right] \tag{9}$$

At iteration n+1 of the EM algorithm the revised estimate of the mean pattern vector for class ω is

$$Y_\omega^{(n+1)} = \sum_{t=1}^{T} P(t\in \omega | X_t, Y_\omega^{(n)}, \Sigma_\omega^{(n)}) X_t \tag{10}$$

while the revised estimate of the covariance matrix is

$$\Sigma_\omega^{(n+1)} = \sum_{t=1}^{T} P(t\in \omega | X_t, Y_\omega^{(n)}, \Sigma_\omega^{(n)})(X_t - Y_\omega^{(n)})(X_t - Y_\omega^{(n)})^T \tag{11}$$

When the algorithm has converged, then the point-distribution models for the different classes may be constructed off-line using the procedure outlined in Section 3. For the class ω, we denote the eigenvector matrix by Φ_ω.

4. HIERARCHICAL ARCHITECTURE

With the stroke and shape point distribution models to hand, our recognition method proceeds in a hierarchical manner. To commence, we make maximum likelihood estimates of the best-fit parameters of each stroke-model to each set of stroke-points. The best-fit parameters γ_λ^k of the stroke-model with class-label λ to the set of points constituting the stroke indexed k is

$$\gamma_\lambda^k = \arg\max_{\gamma} p(z_k | \Phi_\lambda, \gamma) \tag{12}$$

We use the best-fit parameters to assign a label to each stroke. The label is that which has maximum a posteriori probability given the stroke parameters. The label assigned to the stroke indexed k is

$$l_k = \arg\max_{\lambda\in\Omega_s} P(l \mid z_k, \gamma_\lambda, \Phi_\lambda) \tag{13}$$

In practice, we assume that the fit error residuals follow a Gaussian distribution. As a result, the class label is that associated with the minimum squared error. This process is repeated for each stroke in turn. The class identity of the set of strokes is summarised the string of assigned stroke-labels

$$L = < l_1, l_2, \ldots.. > \tag{14}$$

Hence, the input layer is initialised using maximum likelihood stroke parameters and maximum a posteriori probability stroke labels.

The shape-layer takes this information as input. The goal of computation in this second layer is to refine the configuration of stroke labels using global constraints on the arrangement of strokes to form consistent shapes. The constraints come from both geometric and symbolic sources. The geometric constraints are provided by the fit of a stroke-centre point distribution model. The symbolic constraints are provide by a dictionary of permissible stroke-label strings for different shapes.

The parameters of the stroke-centre point distribution model are found using the EM algorithm [8]. Here we borrow ideas from the hierarchical mixture of experts algorithm, and pose the recovery of parameters as that of maximising a gated expected log-likelihood function for the distribution of stroke-centre alignment errors $p(X \mid \Phi_\omega, \Gamma_\omega)$. The likelihood function is gated by two sets of probabilities. The first of these are the a posteriori probabilities $P(\lambda_k^\omega \mid z_k, \gamma_{\lambda_k^\omega}, \Phi_{\lambda_k^\omega})$ of the individual strokes. The second are the conditional probabilities $P(L \mid \Lambda_\omega)$ of the assigned stroke-label string given the dictionary of permissible configurations for shapes of class ω. The expected log-likelihood function is given by

$$L = \sum_{\omega\in\Omega_c} P(L \mid \Lambda_\omega)\{\prod_k P(\lambda_k^\omega \mid z_k, \gamma_{\lambda_k^\omega}, \Phi_{\lambda_k^\omega})\} \ln p(X \mid \Phi_\omega, \Gamma_\omega) \tag{15}$$

The optimal set of stroke-centre alignment parameters satisfies the condition

$$\Gamma_\omega^* = \arg\max_{\Gamma} P(L \mid \Lambda_\omega)\{\prod_k P(\lambda_k^\omega \mid z_k, \gamma_{\lambda_k^\omega}, \Phi_{\lambda_k^\omega})\} \ln p(X \mid \Phi_\omega, \Gamma_\omega) \tag{16}$$

From the maximum likelihood alignment parameters we identify the shape-class of maximum *a posteriori* probability. The class is the one for which

$$\omega^* = \arg\max_{\omega\in\Omega_c} P(\omega \mid X, \Phi_\omega, \Gamma_\omega^*) \tag{17}$$

The class identity of the maximum *a posteriori* probability shape is passed back to the stroke-layer of the architecture. The stroke labels can then be refined in the light of the consistent assignments for the stroke-label configuration associated with the shape-class ω.

$$l_k = \arg\max_{\lambda\in\Omega_s} P(\lambda \mid z_k, \gamma_l^k, \Phi_\lambda) P(L(\lambda,k) \mid \Lambda_\omega) \tag{18}$$

Finally, the maximum likelihood parameters for the strokes are refined

$$\gamma_k = \arg\max_{\gamma} p(\vec{z}_k \mid \Phi_{l_k}, \gamma, \Gamma_\omega^*)) \tag{19}$$

These labels are passed to the shape-layer and the process is iterated to convergence.

5. MODELS

In this section we describe the probability distributions used to model the point-distribution alignment process and the symbol assignment process.

5.1 Point-set Alignment

To develop a useful alignment algorithm we require a model for the measurement process. Here we assume that the observed position vectors, i.e. $\vec{w}_i$ are derived from the model points through a Gaussian error process. According to our Gaussian model of the alignment errors,

$$p(\vec{z}_k \mid \Phi_\lambda, \gamma_\lambda) = \frac{1}{2\pi\sigma} \exp[-\frac{1}{2\sigma^2}(\vec{z}_k - Y_\omega - \Phi_\omega\gamma_\lambda)^T(\vec{z}_k - Y_\omega - \Phi_\omega\gamma_\lambda)] \tag{20}$$

where σ^2 is the variance of the point-position errors which for simplicity are assumed to be isotropic. The maximum likelihood parameter vector is given by

$$\gamma_\lambda^k = \frac{1}{2}(\Phi_\omega^T\Phi_\omega)^{-1}(\Phi_\omega + \Phi_\omega^T)(\vec{z}_k - Y_\omega) \tag{21}$$

A similar procedure may be applied to estimate the parameters of the stroke centre point distribution model.

5.2 Label Assignment

The distribution of label errors is modelled using the method developed by Hancock and Kittler [9]. To measure the degree of error we measure the Hamming distance between the assigned string of labels L and the dictionary item Λ. The Hamming distance is given by

$$H(L,\Lambda_\omega) = \sum_{i=1}^{K} \delta_{l_i,\lambda_i^\omega} \tag{22}$$

where δ is the DiracDelta function. With the Hamming distance to hand, the probability of the assigned string of labels L given the dictionary item Λ is

$$P(L \mid \Lambda_\omega) = K_p \exp[-k_p H(L,\Lambda_\omega)] \tag{23}$$

$$\text{where } K_p = (1-p)^K \text{ and } k_p = \ln\frac{1-p}{p} \tag{24}$$

are constants determined by the label-error probability p.

6. EXPERIMENT

We have evaluated our approach on sets of Arabic characters. Figure 1 shows some of the data used for the purpose of learning and recognition. In total we use, 18 distinct classes of Arabic characters and for each class there are 200 samples. The landmarks are placed uniformly along the length of the characters. The top row shows the example pattern used for training a representative set of shape-classes. The second row of the figure shows the configuration of mean strokes for each class. Finally, the third row shows the stroke centre-points.

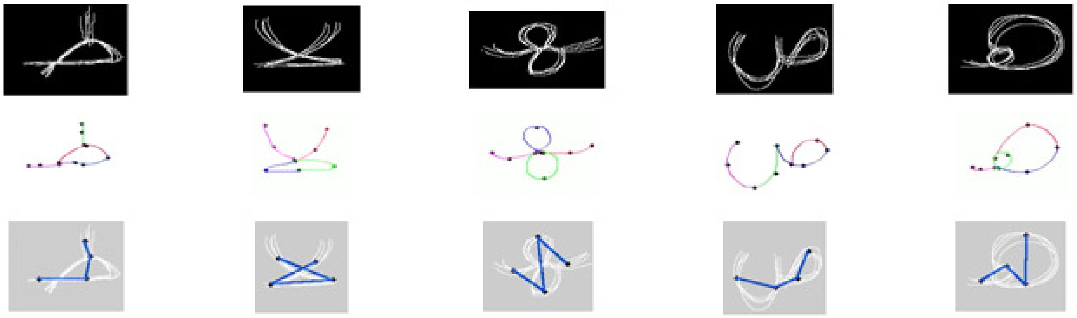


Figure 1: Raw 1 shows sample training sets. Raw 2 shows stroke mean shapes, Raw 3 shows stroke arrangements

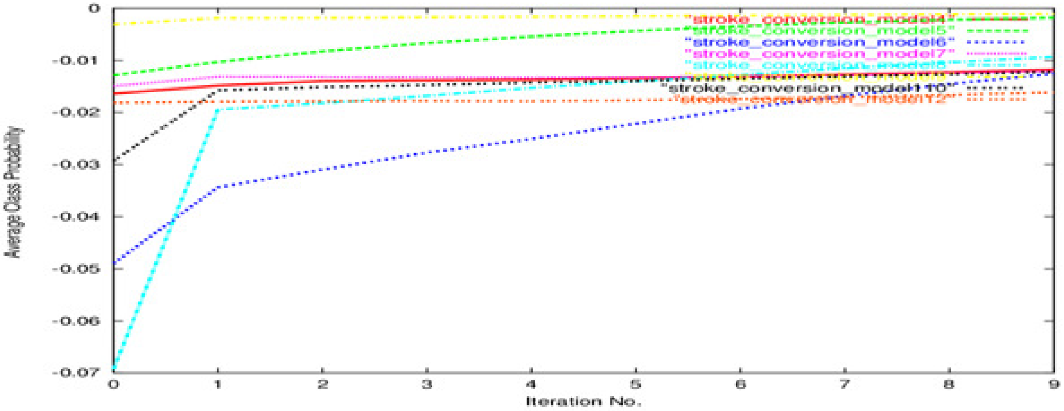


Figure 2: Learning convergence rate for stroke-shape classes as a function per iteration no.

Turning the attention to the learning results for 8 stroke-shape classes. Figure 2 illustrates the average a posteriori stroke-class probabilities for different components. The different curves are for different components shown in figure 1. It is clear from the plot that stroke-classe compnents are convering rapidly in a few iterations.

Table 1 shows the results for a series of recognition experiments. The top raw of the table shows the character shape class. For each shape-class, we have used 200 test patterns to evaluate the recognition performance. These patterns are separate from the data used in training. The raws of the table compare the recognition results obtained using a global point-distribution model to represent the character shape and using the stroke decomposition model described in this paper. We list the number of correctly identified patterns. In the case of the global PDM, the average recognition accuracy is 93.2% over the different character classes. However, in the case of the stroke decomposition method the accuracy is 97%. Hence, the new method offers a performance gain of some 5%.

Table 1. Recognition Rate for shape-classes (Full Character, Stroke-based arrangements)

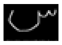
Shape												
Character	190	195	192	190	193	177	184	183	178	186	187	182
Stroke	197	198	197	199	198	189	196	192	194	190	193	188

Figure 3 examines the iterative qualities of the method. Here we plot the a posteriori class probability as a function of iteration number when recognition of characters of a specific class is attempted. The different curves are for different classes. The plot shows that the method converges in about six iterations and that the probabilities of the subdominant classes tend to zero.

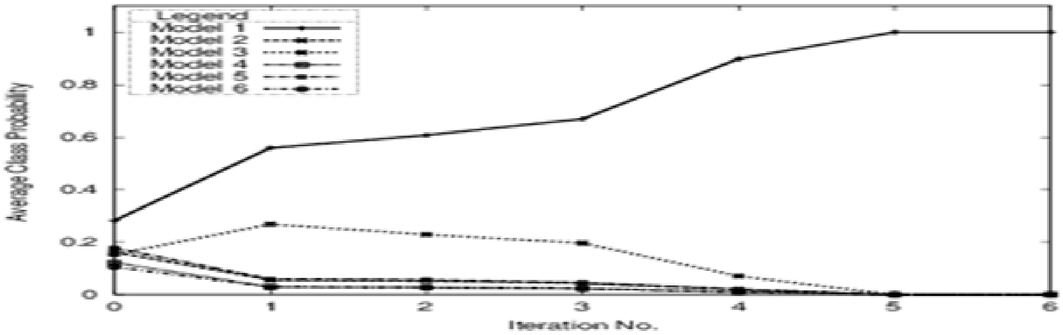


Figure 3. Alignment convergence rate as a function per iteration number

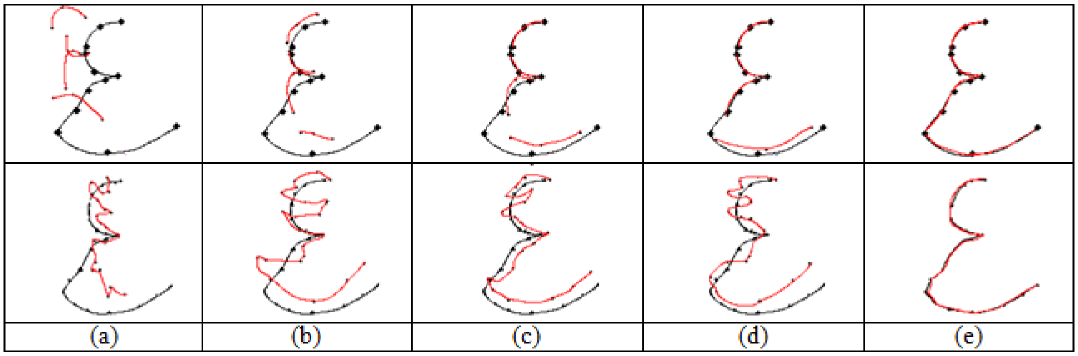


Figure 1. Alignment. First raw shows hierarchical strokes:(a)iteration 1, (b)iteration 2, (c)iteration 3, (d)iteration 4, (e)iteration 5. Second raw represents character models:(a)iteration 1, (b)iteration 2, (c)iteration 3, (d)iteration 4, (d)iteration 8

Figure 4 compares the fitting of the stroke model (top row) and a global PDM (bottom row) with iteration number. It is clear that the results obtained with the stroke model are better than those obtained with the global PDM, which develops erratic local deformations. Finally, we demonstrate in Figure 4 comparison of stroke decomposition and character with respect to recognition rate as a function of point position error. Stroke decomposition methods shows a better performance when points are moved randomly away of their original location.

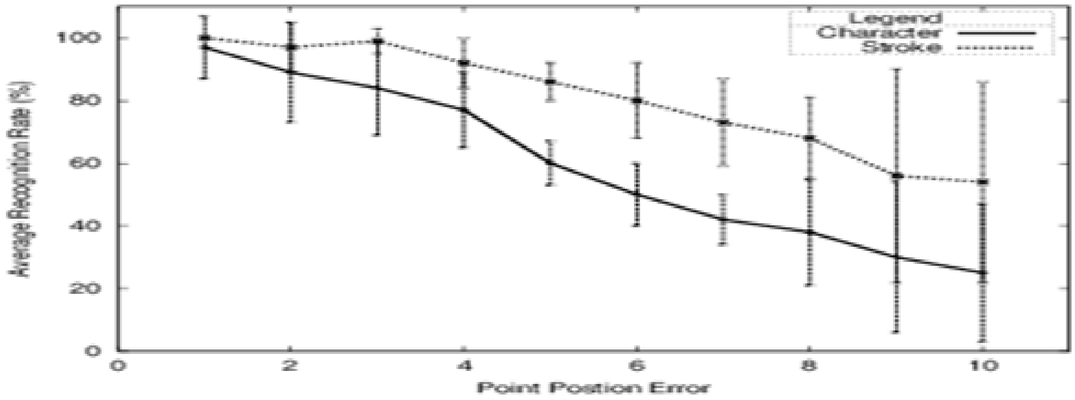
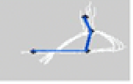
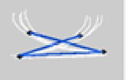
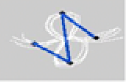
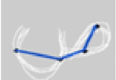
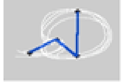


Figure 5. Recognition rate with respect to random point position.

Table 2 shows the shape-classes label replacement process for shapes segmented into 4 strokes. The table clearly shows the correct and incorrect label assignments for each shape-class as a function per iteration number. The stroke-class label replacement hypothesis converges in a fewer iterations.

Table 2: Stroke-Shape class label replacement error as a function per iteration number

Itertaion No.										
	Correct	Wrong	Correct	Wrong	Correct	Wrong	Correct	Wrong	Correct	Wrong
1	3	1	0	4	1	3	0	4	0	4
2	3	1	1	3	0	4	0	4	0	4
3	4	0	0	4	0	4	0	4	0	4
4	4	0	0	4	0	4	0	4	0	4
5	4	0	0	4	0	4	0	4	0	4
6	4	0	0	4	0	4	0	4	0	4

7. CONCLUSION

In This Paper, we have described a hierarchical probabilistic framework for shape recognition via stroke decomposition. The structural component of the method is represented using symbolic dictionaries, while geometric component is represented using Point Distribution Models. Experiments on Arabic character recognition reveal that the method offers advantages over a global PDM.

REFERENCES

[1] T. Cootes, C. Taylor, D. Cooper and J. Graham, "Active Shape Models-Their Training and Application," Computer Vision and Image Understanding, vol. 61, no. 1, pp. 38-59, 1995.

[2] S. Belongie, J. Malik and a. J. Puzicha, "Shape matching and object recognition using shape contexts," IEEE Transactions on PAMI, vol. 24, no. 24, pp. 509-522, 2002.

[3] D. C. Marr, Vision: a computational investigation into the human representation and processing of visual information, San Francisco: Freeman, 1982.

[4] M. Jordan and R. Jacobs, "Hierarchical mixtures of experts and the em algorithm," Neural Computation, vol. 6, pp. 181-214, 1994.

[5] T. Cootes and C. Taylor, "A mixture models for representing shape variation," Image and Vision Computing, vol. 17, pp. 403-409, 1999.

[6] T. Cootes and C. Taylor, "Combining point distribution models with shape models based on finite element analysis," Image and Vision Computing, vol. 13, no. 5, pp. 403-409, 1995.

[7] A. A. Al-Shaher and E. R. Hancock, "Linear shape recognition with mixtures of point distribution models," in SSPR, Windsor, Canada, 2002.

[8] A. Dempster, L. N. and R. D., "Maximum likelihood from incomplete data via the em algorithm," Journal of Royal Statistical Soc. Ser., vol. 39, pp. 1-38, 1977.

[9] E. R. Hancock and J. Kittler, "Discrete relaxation," Pattern Recognition, vol. 23, no. 7, pp. 711-733, 1990.