Arabic Character Recognition using Shape Mixtures

Abdullah A. Al-Shaher and Edwin. R. Hancock
Department of Computer Science,
University of York,York YO1 5DD, UK.
abdullah,erh@minster.cs.york.ac.uk

Abstract

This paper demonstrates how the EM algorithm can be used for learning and matching mixtures of point distribution models. We make two contributions. First, we show how the shape-classes can be learned in an unsupervised manner by updating the deformation matrix iteratively. Second, we show how recognition by alignment can be realised by fitting a mixture of linear shape deformations. We evaluate the method on the problem of learning class-structure and recognising Arabic characters.

1 Introduction

Deformable models have proved to be both powerful and effective tools in the analysis of objects which present variable shape and appearance. There are many examples in the literature. These include the point distribution model of Cootes and Taylor [1], Sclaroff and Pentland's [2] finite element method, and, Duta and Jain's [3] elastic templates. There are two issues to be considered when designing a deformable model. The first of these is how to represent the modes of variation of the object under study. The second is how to train the deformable model. One of the most popular approaches is to allow the object to undergo linear deformation using in the directions of the modal variations of shape. These modes of variation can be found by either performing principal components [4], or independent components analysis on the covariance matrix for a set of training examples [5], or by computing the modes of elastic vibration [6].

Our aim is to explore how point-distribution models can be trained and fitted to data when multiple shape classes or modes of shape-variation are present. The former case arises when unsupervised learning of multiple object models is attempted. The latter problem occurs when shape variations can not be captured by a single linear model. Here we show how both learning and model fitting can be effected using the apparatus of the EM algorithm.

In the learning phase, we use the EM algorithm to extract a mixture of point-distribution models from the set of training data. Here each shape-class is represented using a Gaussian distribution with its own mean-shape and covariance matrix. From the estimated parameters of the Gaussian mixtures, the point-distribution model can be constructed off-line by performing PCA on the class covariance matrices. In the model fitting phase, we fit a mixture of PDM's using an architecture reminiscent of the hierarchical mixture of experts algorithm of

BMVC 2002 doi:10.5244/C.16.48

Jordan and Jacobs [7]. Here each of the class-dependant PDM's identified in the learning step is treated as an expert. The recognition architecture is as follows. Each point in the test pattern may associated to each of the landmark points in each of the class-dependant PDM's with an *a posteriori* probability. In addition, we maintain a set of alignment parameters between the test pattern and each of the PDM's. We experiment with the method on Arabic characters. Here we use the new methodology to learn character classes and perform recognition by alignment. This is a challenging problem since the data used exhibits a high degree of variability.

The resulting framework clearly has a great deal in common with work reported elsewhere in the literature. For instance Jojic and Frey [8] have used the EM algorithm to fit mixture models to the appearance manifolds for faces. Bishop and Winn [9] have used a mixture of principal components analysers to learn and synthesise variations in facial appearance. Vasconcelos and Lippman [10] have used the EM algorithm to learn queries for content-based image retrieval. Finally, several authors have used the EM algorithm to track multiple moving objects [11][12].

2 Point Distribution Models

The point distribution model of Cootes and Taylor commences from a set training patterns. Each training pattern is a configuration of labelled point co-ordinates or landmarks. The landmark patterns are collected as the the object in question undergoes representative changes in shape. To be more formal, each landmark pattern consists of L labelled points whose co-ordinates are represented by the set of position co-ordinates $\{X_1, X_2, ..., X_l\} = \{(x_1, y_1), ...(x_L, y_L)\}$. Suppose that there are T landmark patterns. The t^{th} training pattern is represented using the long-vector of landmark co-ordinates $X_t = (x_1, y_1, x_2, y_2, \cdots, x_L, y_L)^T$, where the subscripts of the co-ordinates are the landmark labels. For each training pattern the labelled landmarks are identically ordered. The mean landmark pattern is represented by the average long-vector of co-ordinates $Y = \frac{1}{T}\sum_{t=1}^{T} X_t$. The covariance matrix for the landmark positions is

$$\Sigma = \frac{1}{T}\sum_{t=1}^{T}(X_t - Y)(X_t - Y)^T \tag{1}$$

The eigenmodes of the landmark covariance matrix are used to construct the point-distribution model. First, the eigenvalues λ of the landmark covariance matrix are found by solving the eigenvalue equation $|\Sigma - \lambda I| = 0$ where I is the $2L \times 2L$ identity matrix. The unit eigen-vector ϕ_i corresponding to the eigenvalue λ_i is found by solving the eigenvector equation $\Sigma\phi_i = \lambda_i\phi_i$. According to Cootes and Taylor [1], the landmark points are allowed to undergo displacements relative to the mean-shape in directions defined by the eigenvectors of the covariance matrix Σ. To compute the set of possible displacement directions, the K most significant eigenvectors are ordered according to the magnitudes of their corresponding eigenvalues to form the matrix of column-vectors $\Phi = (\phi_1|\phi_2|...|\phi_K)$, where $\lambda_1, \lambda_2, ..., \lambda_K$ 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 $\hat{X} = Y + \Phi\gamma$, where γ is a vector of modal co-efficients. This vector represents the free-parameters of the global shape-model.

3 Learning Mixtures of PDM's

In Cootes and Taylor's method [?], 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.

In this paper we adopt an approach based on mixtures of point-distributions. 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

$$p(X_t, t = 1, ..., T) = \prod_{t=1}^{T} \sum_{\omega \in \Omega} p(X_t | Y_\omega, \Sigma_\omega) \tag{2}$$

where $p(X_t | Y_\omega, \Sigma_\omega)$ 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

$$Q_L(C^{(n+1)} | C^{(n)}) = \sum_{t=1}^{T} \sum_{\omega \in \Omega} P(t \in \omega | X_t, Y_\omega^{(n)}, \Sigma_\omega^{(n)}) \ln p(X_t | Y_\omega^{(n+1)}, \Sigma_\omega^{(n+1)}) \tag{3}$$

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

$$P(t \in \omega | X_t, Y_\omega^{(n)}, \Sigma_\omega^{(n)}) = \frac{p(X_t|Y_\omega^{(n)}, \Sigma_\omega^{(n)})\pi_\omega^{(n)}}{\sum_{\omega\in\Omega} p(X_t|Y_\omega^{(n)}, \Sigma_\omega^{(n)})\pi_\omega^{(n)}} \tag{4}$$

where

$$\pi_\omega^{(n+1)} = \frac{1}{T}\sum_{t=1}^{T} P(t \in \omega | X_t, Y_\omega^{(n)}, \Sigma_\omega^{(n)}) \tag{5}$$

Rather than estimating the point-distribution model off-line using the estimated covariance matrix, our approach involves refining the eigenvector matrix iteratively using the EM algorithm. To do this we first compute the mean pattern vector

$$Y_\omega^{(n+1)} = \sum_{t=1}^{T} P(t \in \omega | X_t, Y_\omega^{(n)}, \Sigma_\omega^{(n)}) X_t \tag{6}$$

Keeping the class means fixed, for each pattern vector X_t and each point distribution model, we find the vectors of modal co-efficients $\gamma_{t\omega}^{(n)}$ which minimises the squared distance, i.e. which satisfies the condition

$$\gamma_{t\omega}^{(n)} = \arg\min_{\gamma} (X_t - Y_\omega - \Phi_\omega^{(n)}\gamma)^T (X_t - Y_\omega - \Phi_\omega^{(n)}\gamma) \tag{7}$$

where $\Phi_\omega^{(n)}$ is the current estimate of the deformation matrix. The vector which satisfies this condition is

$$\gamma_{t\omega}^{(n)} = \frac{1}{2}\left[(\Phi_\omega^{(n)})^T \Phi_\omega^{(n)}\right]^{-1} \left(\Phi_\omega^{(n)} + (\Phi_\omega^{(n)})^T\right)(X_t - Y_\omega^{(n)}) \tag{8}$$

We use the residual errors between the fitted point distribution models to each class to define a probability distribution for the training patterns. We assume that the residuals follow the Gaussian distribution

$$p(X_t|Y_\omega^{(n+1)}, \Sigma_\omega^{(n+1)}) = \frac{1}{(2\pi)^L \sqrt{|\Sigma_\omega^{(n)}|}} \times \exp\left[-\frac{1}{2}(X_t - [Y_\omega^{(n)} + \Phi_\omega^{(n)}\gamma_{t\omega}^{(n)}])^T \right. \\ \left. (X_t - [Y_\omega^{(n)} + \Phi_\omega^{(n)}\gamma_{t\omega}^{(n)}])\right] \tag{9}$$

We use this distribution to update the estimates of the deformation matrices $\Phi_\omega^{(n)}$ in the M-step of the EM algorithm. However, this is not tractable in closed form. Hence, we adopt a gradient ascent approach. Accordingly, we compute the derivative of the expected log-likelihood function with respect to the deformation matrices. Using the rules of matrix differentiation, the required derivative is

$$\frac{\partial Q_L(C^{(n+1)}|C^{(n)})}{\partial \Phi_\omega^{(n)}} = \sum_{t=1}^{T} P(t \in \omega | X_t, Y_\omega^{(n)}, \Sigma_\omega^{(n)}) \times \left[(X_t - Y_\omega^{(n)})(\gamma_{t\omega}^{(n)})^T + \right. \\ \left. \gamma_{t\omega}^{(n)}(X_t - Y_\omega^{(n)})^T + 2\Phi_\omega^{(n)}\gamma_{t\omega}^{(n)}(\gamma_{t\omega}^{(n)})^T\right] \tag{10}$$

The deformation matrices are updated as follows

$$\Phi_\omega^{(n+1)} = \Phi_\omega^{(n)} + \eta \frac{\partial Q_L(C^{(n+1)}|C^{(n)})}{\partial \Phi_\omega^{(n)}} \tag{11}$$

where η is a stepsize, which is controlled heuristically.

4 Recognition by Alignment

Once the set of shape-classes and their associated point-distribution models has been learnt, then they can be used for the purposes of alignment or classification. The simplest recognition strategy would be to align each point-distribution model in turn and compute the associated residuals. This may be done by finding the least-squares estimate of the modal co-efficient vector for each class in turn. The test pattern may then be assigned to the class of whose vector gives the smallest alignment error. However, this simple alignment and recognition strategy can be criticised on a number of grounds. First, it is difficult to apply if the training patterns and the test pattern contain different numbers of landmark points. Second, certain shapes may actually represent genuine mixtures of the patterns encountered in training.

To overcome these two problems, in this Section we detail how the mixture of PDM's can be fitted to data using a variant of the hierarchical mixture of experts algorithm of Jordan and Jacobs [7]. We view the mixture of point-distribution models learnt in the training phase as a set of experts which can preside over the interpretation of test patterns. Basic to our philosophy of exploiting the HME algorithm is the idea that every data-point can in principle associate to each of the landmark points in each of stored class shape-models with some *a posteriori* probability. This modelling ingredient is naturally incorporated into the fitting process by developing a mixture model over the space of potential matching assignments.

The approach we adopt is as follows. Each point in the test pattern is allowed to associate with each of the landmark points in the mean shapes for each class. The degree of association is measured using an *a posteriori* correspondence probability. This probability is computed by using the EM algorithm to align the test-pattern to each mean-shape in turn. This alignment process is effected using the point-distribution model to each class in turn. The resulting point alignment errors are used to compute correspondence probabilities under the assumption of Gaussian errors. Once the probabilities of individual correspondences between points in the test pattern and each landmark point in each mean shape are to hand, then the probability of match to each shape-class may be computed.

4.1 Landmark displacements

Suppose that the test-pattern is represented by the vector $W = (\vec{w}_1, \vec{w}_2, ..., \vec{w}_D)^T$ which is constructed by concatenating D individual co-ordinate vectors $\vec{w}_1$,.....$\vec{w}_D$. However, here we assume that the labels associated with the co-ordinate vectors is unreliable, i.e. we can not use the order of the components of the test-pattern to establish correspondences. We hence wish to align the point distribution model for each class in turn to the unlabelled set of D point position vectors $W = \{\vec{w}_1, \vec{w}_2, ..., \vec{w}_D\}$. The size of this point set may be different to the number of landmark points L used in the training. The free parameters that must be adjusted to align the landmark points with W are the vectors modal co-efficients γ_ω for each component of the shape-mixture learnt in training.

The matrix formulation of the point-distribution model adopted by Cootes and Taylor allows the global shape-deformation to be computed. However, in order to develop our correspondence method we will be interested in individual point displacements. We will focus our attention on the displacement vector for the landmark point indexed j produced by the covariance matrix of the shape-mixture indexed ω. The two components of displacement are the elements eigenvectors indexed $2j-1$ and $2j$. For each landmark point the set of displacement vectors associated with the individual eigenmodes are concatenated to form a displacement matrix. For the j^{th} landmark of the mixing component indexed ω the displacement matrix is

$$\Delta_j^\omega = \begin{pmatrix} \Phi_\omega(2j-1,1) & \Phi_\omega(2j-1,2) & .... & \Phi_\omega(2j-1,K) \\ \Phi_\omega(2j,1) & \Phi_\omega(2j,2) & .... & \Phi_\omega(2j,K) \end{pmatrix} \tag{12}$$

The point-distribution model allows the landmark points to be displaced by a vector amount which is equal to a linear superposition of the displacement-vectors associated with the individual eigenmodes. To this end let γ_ω represent a vector of modal superposition co-efficients for the different eigenmodes. With the modal superposition co-efficients to hand, the position of the landmark j is displaced by an amount $\Delta_j^\omega \gamma$ from the mean-position $\vec{y}_j^\omega$.

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,

$$p(\vec{w}_i|\vec{y}_j^\omega, \Phi_\omega, \gamma_\omega) = \frac{1}{2\pi\sigma} \exp\left[-\frac{1}{2\sigma^2}(\vec{w}_i - \vec{y}_j^\omega - \Delta_j^\omega\gamma_\omega)^T(\vec{w}_i - \vec{y}_j^\omega - \Delta_j^\omega\gamma_\omega)\right] \tag{13}$$

where σ^2 is the variance of the point-position errors which for simplicity are assumed to be isotropic.

4.2 Mixture Model for Alignment

We make a second application of the EM algorithm, with the aim of estimating the matrix of alignment parameters $\Gamma^{(n)} = (\gamma_1^{(n)}|\gamma_2^{(n)}|....|\gamma_{|\Omega|}^{(n)})$ is the matrix of vectors of modal alignment parameters for each of the point-distribution models residing in memory. Under the assumption that the measurements of the individual points in the test-patterns are conditionally independent of one-another, the matrix maximises the expected log-likelihood function

$$Q_A(\Gamma^{(n+1)}|\Gamma^{(n)}) = \sum_{\omega\in\Omega}\sum_{i=1}^{D}\sum_{j=1}^{L} P(\vec{y}_j^\omega|\vec{w}_i, \gamma_\omega^{(n)}) \ln p(\vec{w}_i|\vec{y}_j^\omega, \gamma_\omega^{(n+1)}) \tag{14}$$

With the displacement model developed in the previous section, maximisation of the expected log-likelihood function Q_A reduces to minimising the weighted square error measure

$$\mathcal{E}_A = \sum_{i=1}^{D}\sum_{j=1}^{L} \zeta_{ij\omega}^{(n)} (\vec{w}_i - \vec{y}_j^\omega - \Delta_j^w\gamma_\omega^{(n+1)})^T(\vec{w}_i - \vec{y}_j^\omega - \Delta_j^w\gamma_\omega^{(n+1)}) \tag{15}$$

where we have used the shorthand notation $\zeta_{ij\omega}^{(n)}$ to denote the *a posteriori* correspondence probability $P(\vec{y}_j^\omega|\vec{w}_i, \gamma_\omega^{(n)})$.

4.3 Maximisation

Our aim is to recover the vector of modal co-efficients which minimize this weighted squared error. To do this we solve the system of saddle-point equations which results by setting $\frac{\partial\mathcal{E}_A}{\partial\gamma_\omega^{(n+1)}} = 0$. After applying the rules of matrix differentiation and simplifying the resulting saddle-point equations, the solution vector is

$$\gamma_\omega^{(n+1)} = (\sum_{j=1}^{L} \Delta_j^{\omega\,T}\Delta_j^\omega)^{-1}\{\sum_{i=1}^{D}\sum^{L} {}_{i\ \omega}^{(n)}\vec{w}_i^T\Delta_j^\omega - \sum_{j=1}^{L}\vec{y}_j^{\omega\,T}\Delta_j^\omega\} \tag{16}$$

4.4 Expectation

In the expectation step of the algorithm, we use the estimated alignment parameters to update the *a posteriori* matching probabilities. The *a posteriori* probabilities $P(\vec{y}_j^{\omega}|\vec{w}_i,\gamma_{\omega}^{(n)})$ represent the probability of match between the point indexed i and the landmark indexed j from the shape-mixture indexed ω. In other words, they represent model-datum affinities. Using the Bayes rule, we re-write the *a posteriori* matching probabilities in terms of the conditional measurement densities

$$P(\vec{y}_j^{\omega}|\vec{w}_i,\gamma_{\omega}^{(n)}) = \frac{\beta_{\omega}^{(n)}\alpha_{j,\omega}^{(n)}p(\vec{w}_i|\vec{y}_j^{\omega},\gamma_{\omega}^{(n)})}{\sum_{\omega'\in\Omega}\sum_{j'=1}^{L}\beta_{\omega'}^{(n)}\alpha_{j',\omega'}^{(n)}p(\vec{w}_i|\vec{y}_{j'}^{\omega'},\gamma_{\omega'}^{(n)})} \qquad (17)$$

The landmark mixing proportions for each model in turn are computed by averaging the *a posteriori* probabilities over the set of points in the pattern being matched, i.e. $\alpha_{j,\omega}^{(n+1)} = \frac{1}{D}\sum_{i=1}^{D}P(\vec{y}_j^{\omega}|\vec{w}_i,\gamma_{\omega}^{(n)})$. The *a posteriori* probabilities for the components of the shape mixture are found by summing the relevant set of point mixing proportions, i.e. $\beta_{\omega}^{(n+1)} = \sum_{j=1}^{L}\alpha_{j,\omega}^{(n+1)}$. In this way the *a posteriori* model probabilities sum to unity over the complete set of models. The probability assignment scheme allows for both model overlap and the assessment of ambiguous hypotheses. Above we use the shorthand notation $\alpha_{j,\omega}^{(n)}$ to represent the mixing proportion for the landmark point j from the model ω. The overall proportion of the model ω at iteration n is $\beta_{\omega}^{(n)}$. These quantities provide a natural mechanism for assessing the significance of the individual landmark points within each mixing component in explaining the current data-likelihood. For instance if $\alpha_{j,\omega}^{(n)}$ approaches zero, then this indicates that there is no landmark point in the data that matches the landmark point j in the model ω.

Table 1. Error measure comparison

Measure	Single PDM's	Mixture PDM's
Least Square Error	578076.452704	566767.975989
Mahalanobis Distance	11438.000	3260.010

Table 2. Recognition Rate for shape-classes 1-7

		Single PDM		Mixture of PDM's	
Model No.	Samples	Correct	Wrong	Correct	Wrong
Shape-Class 1	100	90	10	97	3
Shape-Class 2	100	96	4	99	1
Shape-Class 3	100	96	4	100	0
Shape-Class 4	100	90	10	98	2
Shape-Class 5	100	93	7	98	2
Shape-Class 6	100	97	3	97	3
Shape-Class 7	100	82	18	95	5
Recognition Rate	700	92.0%	8.0%	97.7%	2.3%

5 Experiments

We have evaluated our learning and recognition method on sets of Arabic characters. Here the landmarks used to construct the point-distribution models have

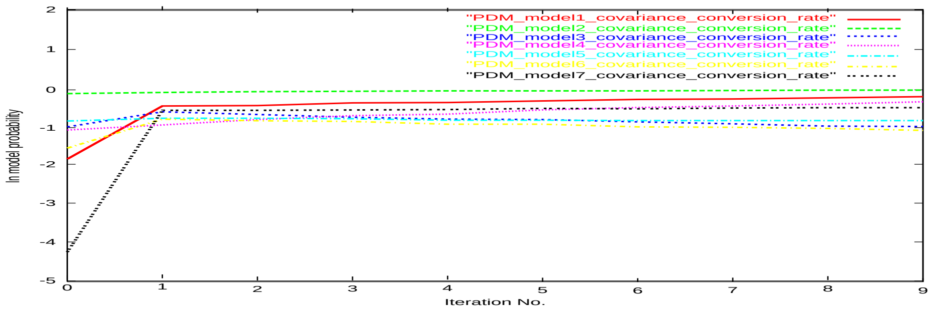

Figure 1: Convergence rate for mixture of PDM's shape-classes

been positioned by distributing points uniformly along the length of the characters. In practice we use 20 landmarks per character in 2D space. In total there are 7 different classes of character. We use 100 samples of each character for the purposes of training and 100 sample of each character class for recognition.

5.1 Learning

In Table 1 we list the mean-squared error and the Mahalanobis distance obtained when the characters in a single class are represented by a single PDM and by a mixture of PDM's. Here there are 7 examples in the training-set and the mixture of PDM's contains 100 components. The mixture of PDM's results in considerably lower error than the single PDM. In Figure 1 we show the convergence rate as a function of iteration number for three variants of the mixture model. The plot again shows the average *a posteriori* class probabilities for the different characters. The different curves are for the different components of the shape-mixture model. The main conclusion that can be drawn from the plot is that the method converges rapidly and uniformly. In Figure 2, we show the mean-shapes learnt in training. In columns (a,d) of the figure, we show the ground-truth mean shapes. In columns (c,f), we shows the learnt shapes. The two are in good agreement.

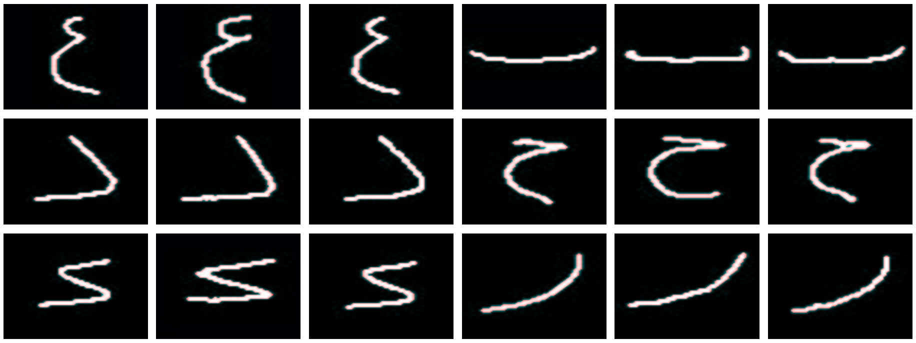

Figure 2: columns (a),(d) Actual mean shapes; (b),(e) EM Initializations; (c),(f) Final Mean Shapes

5.2 Recognition

We now turn our attention to the results obtained when the shape-mixture is used for the purposes of recognition by alignment. In Figure 3 we illustrate the fitting of a mixture of PDM's to a character retained from the training-set. The different images in the sequence show the fitted PDM's as a function of iteration number. The shape shown is the one with the largest a posteriori probability. Figure 3 shows the result obtained when training is performed using a mixture of PDM's. In Figure 4 we show the alignments of the subdominant shape-components of the mixture. These are all very poor and fail to account for the data. In Figure 5 we show the *a posteriori* probabilities β_ω for each of the mixing components on convergence. The different curves are for different shape-classes. A single dominant shape hypothesis emerges after a few iterations. The probabilities for the remaining shape-classes fall towards zero. Note that initially the different classes are equiprobable, i.e. we have not biased the initial probabilities towards a particular shape-class.

Finally, we measure the recognition rates achievable using our alignment method. Here we count the number of times the maximum a posteriori probability shape, i.e. the one for which $\omega = \arg\max \beta_\omega$, corresponds to the hand-labelled class of the character. This study is performed using 700 hand-labelled characters. Table 2 lists the recognition rates obtained in our experiments. The table lists the numbers of characters recognised correctly and incorrectly for each of the shape-classes; the results a given for both single PDM's and a mixture of PDM's. The main conclusions to be drawn from the table are as follows. First, the mixture of PDM's gives a better recognition rate than using separately trained single PDM's for each class. Hence, recognition can be improved using a more complex model of the shape-space.

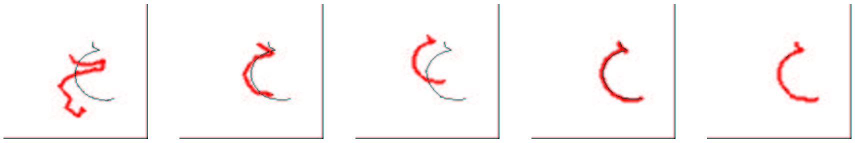

Figure 3: Model alignment to data using mixtures PDM's at iteration 1, 2, 3, 5, 7

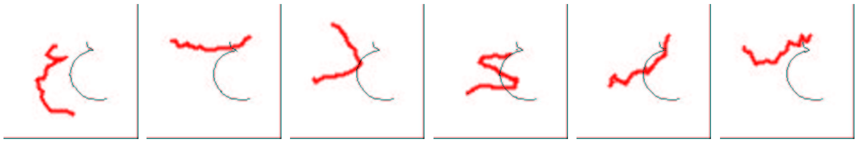

Figure 4: Sub dominant model alignment to data using mixture of PDM's

6 Conclusion

In this paper, we have shown how mixtures of point-distribution models can be learnt and then subsequently used for the purposes of recognition by alignment. In the training phase, we show how to use the method to learn the class-structure of complex and varied sets of shapes. In the recognition phase, we show how a variant of the hierarchical mixture of experts architecture can be used to perform detailed model alignment. We present results on sets of A

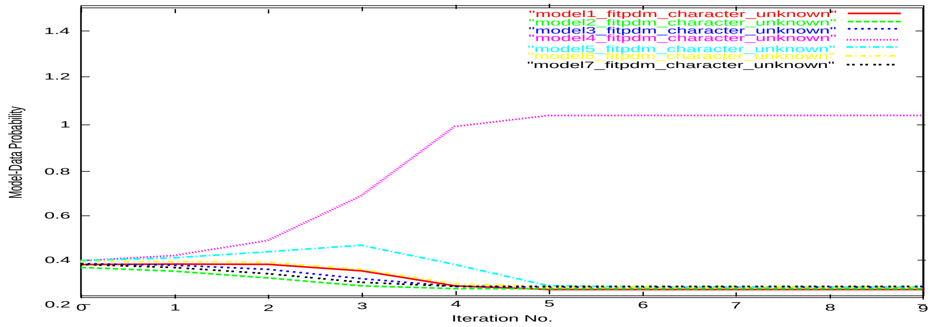

Figure 5: Model fitting with a mixture of PDM's.

that the mixture of PDM's gives excellent performance. Our future plans revolve around developing a hierarchical approach to the shape learning and recognition problem. Here we aim to decompose shapes into strokes and to learn both the variations in stroke shape, and the variation in stroke arrangement.

References

[1] Cootes T., Taylor C., Cooper D., Graham J. (1992). *Trainable method of parametric shape description*. Image and Vision Computing, Vol 10, no. 5, PP. 289-294.

[2] Sclaroff S., Pentland A. (1995). *Model Matching for correspondence and recognition*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 17, no. 6, PP. 545-561.

[3] Duta N., Jain A., Dubuisson P. (1999). *Learning 2D Shape Models*. International Conference on Computer Vision and pattern Recognition, vol 2, PP. 8-14, 1999.

[4] Cootes T., Taylor C., Cooper D., Graham J. (1995). *Active Shape Models-Their Training and Application*. Computer Vision and Image Understanding, Vol 61, no. 1, PP. 38-59.

[5] Duda R., Hart P. (1973) *Pattern Classification and Scene Analysis*. Wiley.

[6] Martin J., Pentland A., Sclaroff S., Kikinis R. (1998) *Characterization of neuropathological shape deformations*. IEEE transactions on Pattern Analysis and Machine Intelligence, vol 20, no. 2, pp. 97-112.

[7] Jordan M.; Jacobs R. (1994). *Hierarchical mixtures of experts and the EM algorithm*. Neural Computation, vol 6, PP. 181-214.

[8] Brendan J; Jojic N. (1999). *Estimating Mixture Models of Images and Inferring Spatial Transformations using the EM Algorithm*. IEEE Computer Vision and Pattern Recognition, vol 2, PP. 416-422.

[9] Bishop C.; Winn J. (2000). *Non-Linear Bayesian Image Modelling*. Proceedings of 6th European Conference on Computer Vision, part 1, PP. 3-17, 2000.

[10] Vasconcelos N.; Lippman A. (2000). *A Probabilistic Architecture for Content-based Image Retrieval*. Proceedings of International Conference on Computer Vision and Pattern Recognition, South Carolina, pp. 216-221.

[11] North B.; Blake A. (1999). *Using Expectation-Maximisation to learn dynamical models from visual data*. Image and Vision Computing , vol 17, part 8, pp. 611-616.

[12] Weiss Y.; Adelson E. (1996). *A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models*. IEEE Computer Vision and Pattern Recognition , pp. 321-326.