



Learning mixtures of point distribution models with the EM algorithm

Abdullah A. Al-Shaher*, Edwin R. Hancock

Department of Computer Science, University of York, York YO1 5DD, UK

Received 5 December 2002; accepted 2 April 2003

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 shape-classes can be learned in an unsupervised manner. We present a fast procedure for training point distribution models using the EM algorithm. Rather than estimating the class means and covariance matrices needed to construct the PDM, the method iteratively refines the eigenvectors of the covariance matrix using a gradient ascent technique. 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 the class-structure and recognising Arabic characters. © 2003 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.

Keywords: Point distribution models; Expectation maximization algorithm; Unsupervised learning; Alignment; Shape recognition; Arabic character

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, i.e. how to learn its parameters. One of the most popular approaches is to allow the object to undergo linear deformation 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].

Recently, there have been attempts to extend the utility of such methods by allowing for non-linear deformations of shape [7]. Here there are two contrasting approaches. The first of these is to use a non-linear deformation model. While this approach, increases the representational power of the method, this is at the expense of greater difficulty in controlling and regulating the model parameters. The second approach is to use a combination of locally linear models [8]. Here parameter estimation is easier, but control of the model order still raises difficulties. In this paper, because of its relative simplicity, we focus on the latter approach.

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 cannot 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. In the conventional approach to learning, the EM algorithm is used to estimate a mixture of point-distribution

* Corresponding author.

E-mail address: abdullah.ersh@minster.cs.york.ac.uk
(A.A. Al-Shaher).

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 Principal Component Analysis (PCA) [9] on the class covariance matrices. This can prove time-consuming since the covariance matrix must be re-estimated at each iteration, and this involves costly matrix inversion operations. In this paper, we adopt an alternative approach. This involves the on-line refinement of the PDM. Here we iteratively modify the modal directions in the maximisation steps of the EM algorithm. In this way we avoid the need for repeated matrix inversion.

In the model fitting phase, we fit a mixture of PDMs using an architecture reminiscent of the hierarchical mixture of experts algorithm of Jordan and Jacobs [10]. Here each of the class-dependant PDMs identified in the learning step is treated as an expert. The recognition architecture is as follows. Each point in the test pattern may associate to each of the landmark points in each of the class-dependant PDMs with an a posteriori probability. In addition, we maintain a set of alignment parameters between the test pattern and each of the PDMs. Of course, there are limitations involved in using the EM algorithm. For instance, it can be slow to converge. The method is also sensitive to initialisation. However, this latter problem can be overcome by prior or domain knowledge. In fact, good initialisation can significantly improve the learning mixture of shape distributions [11,12].

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 shape variability.

The resulting framework clearly has a great deal in common with work reported elsewhere in the literature. For instance Jovic and Frey [13] have used the EM algorithm to fit mixture models to the appearance manifolds for faces. Bishop and Winn [14] have used a mixture of principal components analysers to learn and synthesise variations in facial appearance. Vasconcelos and Lippman [15] 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 [16–18]. Revov et al. [19] has developed a generative model which can be used for handwritten character recognition. Their method employs the EM algorithm to model the distribution of sample points. The model can be used for both recognition and learning.

The outline of this paper is as follows. In Section 2, we review the point distribution model in its simplest form. Section 3 details our learning method. In Section 4 we describe how the model may be fitted to the data using the EM algorithm. Experiments are presented in Section 5. Finally, Section 6 offers conclusions and directions for future work.

2. Point distribution models

The point distribution model of Cootes and Taylor commences from a set training patterns. In our study we have normalised the point patterns by subjecting them to Procrustes alignment as a preprocessing step [20]. This involves the following steps. First, the centroids of the patterns are brought into alignment. Second, the patterns are scaled so that the variances of the point-positions are identical. Third, the patterns are rotated so that their principle components axes are aligned. Each training pattern is a configuration of labelled point co-ordinates or landmarks.

The landmark patterns are collected as the object in question undergoes representative changes in shape. To be more formal, each landmark pattern consists of L labelled points. Suppose that there are T landmark patterns. The t th training pattern is represented using the long-vector of landmark co-ordinates $X_t = (x_1, y_1, x_2, y_2, \dots, x_T, y_T)^T$, 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

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

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 eigenvector 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 [21], 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 $\Phi = (\phi_1 | \phi_2 | \dots | \phi_K)$, where $\lambda_1, \lambda_2, \dots, \lambda_K$ 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 PDMs

In Cootes and Taylor's method [22], learning involves extracting a single covariance matrix from the sets of landmark points. Hence, the method can only reproduce vari-

ations 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 cannot 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 landmark point-pattern Y_ω and covariance matrix Σ_ω . With these ingredients, the likelihood function for the set of training patterns is

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

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

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

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

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

where

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

3.1. Off-line learning

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

$$\begin{aligned} p(X_t | Y_\omega^{(n)}, \Sigma_\omega^{(n)}) &= \frac{1}{(2\pi)^L \sqrt{|\Sigma_\omega^{(n)}|}} \exp \left[-\frac{1}{2} (X_t - Y_\omega^{(n)})^T \right. \\ &\quad \left. \times (\Sigma_\omega^{(n)})^{-1} (X_t - Y_\omega^{(n)}) \right]. \end{aligned} \quad (6)$$

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

$$Y_\omega^{(n+1)} = \sum_{t=1}^T P(t \in \omega | X_t, Y_\omega^{(n)}, \Sigma_\omega^{(n)}) X_t, \quad (7)$$

while the revised estimate of the covariance matrix is

$$\begin{aligned} \Sigma_\omega^{(n+1)} &= \sum_{t=1}^T P(t \in \omega | X_t, Y_\omega^{(n)}, \Sigma_\omega^{(n)}) \\ &\quad \times (X_t - Y_\omega^{(n)}) (X_t - Y_\omega^{(n)})^T. \end{aligned} \quad (8)$$

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 2. For the class ω , we denote the eigenvector matrix by $\Phi_\omega = (\phi_1^\omega | \phi_2^\omega | \dots)$. It is the estimation of the covariance matrix and its inverse which proves to be the main computational bottleneck in this approach.

3.2. On-line learning

Rather than estimating the point-distribution model off-line using the estimated covariance matrix, our second approach involves refining the eigenvector matrix iteratively using the EM algorithm. We develop a fast method of learning the class point-distribution model in an on-line fashion. To do this we first compute the mean pattern vector

$$Y_\omega^{(n+1)} = \sum_{t=1}^T P(t \in \omega | X_t, Y_\omega^{(n)}, \Sigma_\omega^{(n)}) X_t. \quad (9)$$

Keeping the class means fixed, for each pattern vector X_t and each point distribution model, we find the vectors of

modal co-efficients $\gamma_{\omega}^{(n)}$ which minimises the squared distance, i.e. which satisfies the condition

$$\gamma_{\omega}^{(n)} = \arg \min_{\gamma} (X_t - Y_{\omega} - \Phi_{\omega}^{(n)} \gamma)^T (X_t - Y_{\omega} - \Phi_{\omega}^{(n)} \gamma), \quad (10)$$

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

$$\gamma_{\omega}^{(n)} = \frac{1}{2} [(\Phi_{\omega}^{(n)})^T \Phi_{\omega}^{(n)}]^{-1} (\Phi_{\omega}^{(n)} + (\Phi_{\omega}^{(n)})^T) (X_t - Y_{\omega}^{(n)}). \quad (11)$$

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

$$\begin{aligned} p(X_t | Y_{\omega}^{(n+1)}, \Sigma_{\omega}^{(n+1)}) \\ = \frac{1}{(2\pi)^L \sqrt{|\Sigma_{\omega}^{(n+1)}|}} \exp \left[-\frac{1}{2\sigma^2} (X_t - [Y_{\omega}^{(n)} + \Phi_{\omega}^{(n)} \gamma_{\omega}^{(n)}])^T \right. \\ \left. \times (X_t - [Y_{\omega}^{(n)} + \Phi_{\omega}^{(n)} \gamma_{\omega}^{(n)}]) \right]. \end{aligned} \quad (12)$$

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

$$\begin{aligned} \frac{\partial Q(C^{(n+1)} | C^{(n)})}{\partial \Phi_{\omega}^{(n)}} \\ = \sum_{t=1}^T P(t \in \omega | X_t, Y_{\omega}^{(n)}, \Sigma_{\omega}^{(n)}) \times [(X_t - Y_{\omega}^{(n)}) (\gamma_{\omega}^{(n)})^T \\ + \gamma_{\omega}^{(n)} (X_t - Y_{\omega}^{(n)})^T + 2\Phi_{\omega}^{(n)} \gamma_{\omega}^{(n)} (\gamma_{\omega}^{(n)})^T]. \end{aligned} \quad (13)$$

The deformation matrices are updated by summing the matrices elements and the product of the aligned elements

$$\Phi_{\omega}^{(n+1)} = \Phi_{\omega}^{(n)} + \eta \frac{\partial Q(C^{(n+1)} | C^{(n)})}{\partial \Phi_{\omega}^{(n)}}, \quad (14)$$

where η is a stepsize, which is controlled heuristically. The value of this stepsize parameter is selected from the interval $[0, 1]$.

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 PDMs can be fitted to data using a variant of the hierarchical mixture of experts algorithm of Jordan and Jacobs [10]. 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 position 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 $W = (\vec{w}_1^T, \vec{w}_2^T, \dots, \vec{w}_D^T)^T$

which is constructed by concatenating D individual co-ordinate vectors $\vec{w}_1, \dots, \vec{w}_D$. However, here we assume that the labels associated with the co-ordinate vectors may be unreliable, i.e. we cannot 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 $\mathcal{W} = \{\vec{w}_1, \vec{w}_2, \dots, \vec{w}_D\}$. 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 the test pattern 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 eigenmode indexed λ of 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

$$A_j^\omega = \begin{pmatrix} \Phi_\omega(2j-1, 1) & \Phi_\omega(2j-1, 2) & \dots & \Phi_\omega(2j-1, K) \\ \Phi_\omega(2j, 1) & \Phi_\omega(2j, 2) & \dots & \Phi_\omega(2j, K) \end{pmatrix}. \quad (15)$$

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 $A_j^\omega \gamma_\omega$ from the mean-position \bar{y}_j^ω .

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

$$p(\bar{w}_i | \bar{y}_j^\omega, \gamma_\omega) = \frac{1}{2\pi\sigma} \exp \left[-\frac{1}{2\sigma^2} (\bar{w}_i - \bar{y}_j^\omega - A_j^\omega \gamma_\omega)^T \times (\bar{w}_i - \bar{y}_j^\omega - A_j^\omega \gamma_\omega) \right], \quad (16)$$

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

Basic to our philosophy of exploiting the EM algorithm is the idea that every point in the test pattern can in principle associate to each of the landmarks in any of the previously learnt point-distribution 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. Specifically, we aim to construct a mixture model for the conditional data-likelihood $p(W | \Gamma^{(n)})$ where $\Gamma^{(n)} = \{\gamma_1^{(n)} | \gamma_2^{(n)} | \dots | \gamma_{|\Omega|}^{(n)}\}$ is the set of vectors of modal alignment parameters for each of the point-distribution models residing in memory. We commence our development by assuming that the measurements of the individual points in the test-pattern are conditionally independent given the current matrix of PDM alignment parameters:

$$p(W | \Gamma^{(n)}) = \prod_{i=1}^D p(\bar{w}_i | \Gamma^{(n)}). \quad (17)$$

Our next step is to develop a mixture model for the individual measurement densities, i.e. for $p(\bar{w}_i | \Gamma^{(n)})$. Accordingly, we apply the Bayes rule over the space of potential model-data associations between the test pattern and the landmarks of the stored PDMs

$$p(\bar{w}_i | \Phi^{(n)}) = \sum_{\omega \in \Omega} \sum_{j=1}^L p(\bar{w}_i, \bar{y}_j^\omega | \Gamma^{(n)}). \quad (18)$$

Applying the chain-rule, we develop the conditional density under the summation as follows:

$$p(\bar{w}_i | \Gamma^{(n)}) = \sum_{\omega \in \Omega} \sum_{j=1}^L p(\bar{w}_i | \bar{y}_j^\omega, \Gamma^{(n)}) P(\bar{y}_j^\omega | \Gamma^{(n)}). \quad (19)$$

Written in this way, the mixture density has two distinct model-ingredients. The first of these is the set of individual component conditional measurement densities $p(\bar{w}_i | \bar{y}_j^\omega, \Gamma^{(n)})$. This density represents the likelihood that the test-pattern point position measurement \bar{w}_i originated from the landmark point indexed j in the PDM indexed ω . Since the landmark point \bar{y}_j^ω from the mean shape indexed ω only transforms under the prevailing vector of parameters $\gamma_\omega^{(n)}$, we remove the conditional redundancy and write

$$p(\bar{w}_i | \bar{y}_j^\omega, \Gamma^{(n)}) = p(\bar{w}_i | \bar{y}_j^\omega, \gamma_\omega^{(n)}). \quad (20)$$

The second ingredient appearing in the mixture distribution are the model mixing proportions. We use the shorthand notation $\alpha_{j,\omega}^{(n)} = P(\bar{y}_j^\omega | \Gamma^{(n)})$ to represent the mixing proportion for the landmark point j from the model ω . With these ingredients, we can turn our attention to the log-likelihood function for the set of parameter vectors, i.e.

$$\mathcal{L}(\Gamma^{(n)}) = \sum_{i=1}^D \ln p(\bar{w}_i | \Gamma^{(n)}).$$

Substituting for the mixture distribution given in Eq. (21)

$$\mathcal{L}(\Gamma^{(n)}) = \sum_{i=1}^D \ln \sum_{\omega \in \Omega} \sum_{j=1}^L p(\bar{w}_i | \bar{y}_j^\omega, \gamma_\omega^{(n)}) P(\bar{y}_j^\omega | \Gamma^{(n)}). \quad (21)$$

The EM algorithm aims to estimate the data log-likelihood function when the data under consideration is incomplete. In our point-pattern matching example this incompleteness is a consequence of the fact that we do not know how to associate tokens in the test pattern and the landmark points for the set of stored PDMs. It was Dempster et al. [23] who observed that maximising the weighted log-likelihood was equivalent to maximising the conditional expectation of the log-likelihood for a new parameter set given an old parameter set. For our matching problem, maximisation of the expectation of the conditional likelihood, is equivalent to maximising the weighted log-likelihood function

$$\begin{aligned} Q_A(\Gamma^{(n+1)} | \Gamma^{(n)}) &= \sum_{\omega \in \Omega} \sum_{i=1}^D \sum_{j=1}^L P(\bar{y}_j^\omega | \bar{w}_i, \gamma_\omega^{(n)}) \\ &\quad \times \{ \ln p(\bar{w}_i | \bar{y}_j^\omega, \gamma_\omega^{(n+1)}) + \ln P(\bar{y}_j^\omega | \Gamma^{(n)}) \}. \end{aligned} \quad (22)$$

Viewed in this way, the EM algorithm potentially involves two separate maximisation steps for each of the terms under the curly braces. However, the second term is couched purely in terms of the model representation (i.e. the PDM mixing proportions) and is hence not of direct relevance to the data-likelihood. In other words we confine our attention to the quantity,

$$\hat{Q}_A(\Phi^{(n+1)}|\Phi^{(n)}) = \sum_{\omega \in \Omega} \sum_{i=1}^D \sum_{j=1}^L P(\vec{y}_j^{\omega} | \vec{w}_i, \gamma_{\omega}^{(n)}) \times \ln p(\vec{w}_i | \vec{y}_j^{\omega}, \gamma_{\omega}^{(n+1)}). \quad (23)$$

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

$$\mathcal{E} = \sum_{i=1}^D \sum_{j=1}^L \zeta_{ij\omega}^{(n)} (\vec{w}_i - \vec{y}_j^{\omega} - A_j^w \gamma_{\omega}^{(n+1)})^T \times (\vec{w}_i - \vec{y}_j^{\omega} - A_j^w \gamma_{\omega}^{(n+1)}), \quad (24)$$

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

$$\frac{\partial \mathcal{E}}{\partial \gamma_{\omega}^{(n+1)}} = 0. \quad (25)$$

After applying the rules of matrix differentiation and simplifying the resulting saddle-point equations, the solution vector is

$$\gamma_{\omega}^{(n+1)} = 2 \left(\left(\sum_{i=1}^D \sum_{j=1}^L \zeta_{ij\omega}^{(n)} A_j^{\omega T} A_j^{\omega} \right)^{-1} \left\{ \sum_{i=1}^D \sum_{j=1}^L \zeta_{ij\omega}^{(n)} \vec{w}_i^T A_j^{\omega} - \sum_{i=1}^D \sum_{j=1}^L \zeta_{ij\omega}^{(n)} \vec{y}_j^{\omega T} A_j^{\omega} \right\} \right). \quad (26)$$

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

$$P(\vec{y}_j^{\omega} | \vec{w}_i, \gamma_{\omega}^{(n)}) = \frac{\beta_{\omega}^{(n)} \alpha_{j,\omega}^{(n)} P(\vec{w}_i | \vec{y}_j^{\omega}, \gamma_{\omega}^{(n)})}{\sum_{\omega' \in \Omega} \sum_{j'=1}^L \beta_{\omega'}^{(n)} \alpha_{j',\omega'}^{(n)} P(\vec{w}_i | \vec{y}_j^{\omega'}, \gamma_{\omega'}^{(n)})}. \quad (27)$$

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.

$$\alpha_{j,\omega}^{(n+1)} = \frac{1}{D} \sum_{i=1}^D P(\vec{y}_j^{\omega} | \vec{w}_i, \gamma_{\omega}^{(n)}). \quad (28)$$

The a posteriori probabilities for the components of the shape mixture are found by summing the relevant set of point mixing proportions, i.e.

$$\beta_{\omega}^{(n+1)} = \sum_{j=1}^L \alpha_{j,\omega}^{(n+1)}. \quad (29)$$

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 .

5. Experiments

We have evaluated our learning and recognition methods on sets of Arabic characters. Here the landmarks used to construct the point-distribution models have been positioned by distributing points uniformly along the length of the characters. In practice we use 20 landmarks per character. In Fig. 1 we show some of the characters used in our study. In total there are 23 different classes of character. We use 100 samples of each character for the purposes of training. In Fig. 2 we show some examples of the placement of landmark points.

5.1. Learning

In this section we provide some evaluation of the different learning methods. Table 1 lists the mean-squared errors for the training set when the Off-Line and On-Line learning methods are used. For this experiment there are seven distinct classes present in the training-set and there are 100 samples of each character. The On-Line learning method gives a considerably lower error than the Off-Line method.

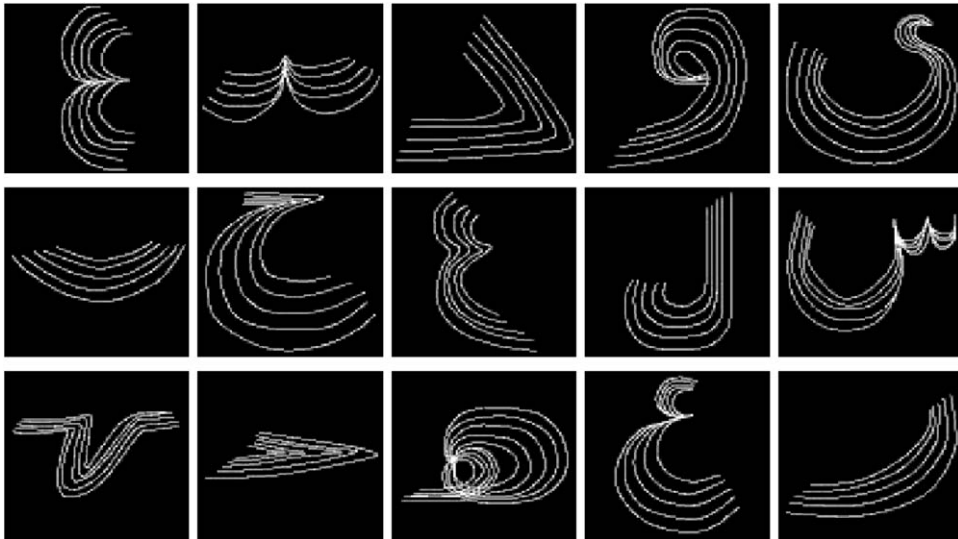


Fig. 1. Sample character training patterns.

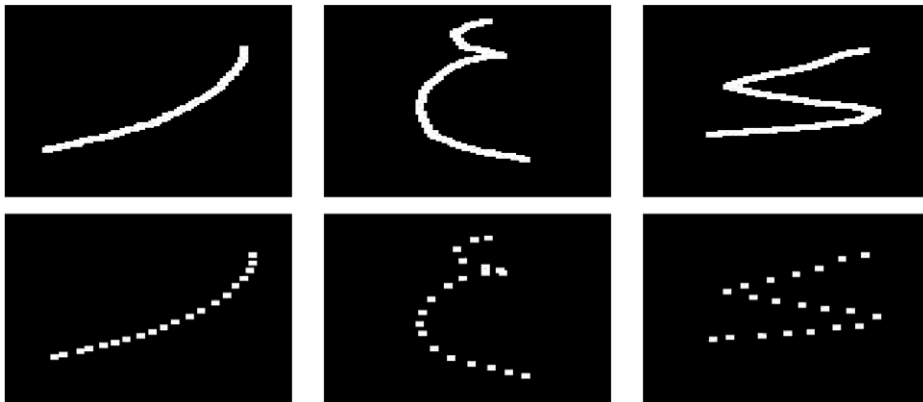


Fig. 2. Handwritten characters and landmark points.

Table 1
Error measure comparison

Measure	Single PDM	On-line PDMs	Off-line PDMs
Least-Square Error	578076.452704	483339.546138	566767.975989

In Figs. 3 and 4 we examine the rate of convergence for the off-line and on-line learning methods. The plot again shows the average a posteriori class probabilities for the seven different character classes. 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 both methods converge in a comparable number of iterations. In both cases the convergence is rapid.

In Fig. 5, we show the mean-shapes learnt in training. The first column of the figure shows the ground-truth mean shapes. The second column shows the mean shapes used to initialize the EM algorithm. The third column shows the final mean shapes learnt using the off-line method. Finally, the fourth column shows the mean shapes learnt using the on-line method. There is a slight difference the shapes recovered by the two methods. However, they are generally in good agreement. The mean shapes used to initialise the EM algorithm are found by fitting a PDM to the median shape in each class.

5.2. Recognition

We now turn our attention to the recognition results that can be obtained when the mixture of PDMs is trained with

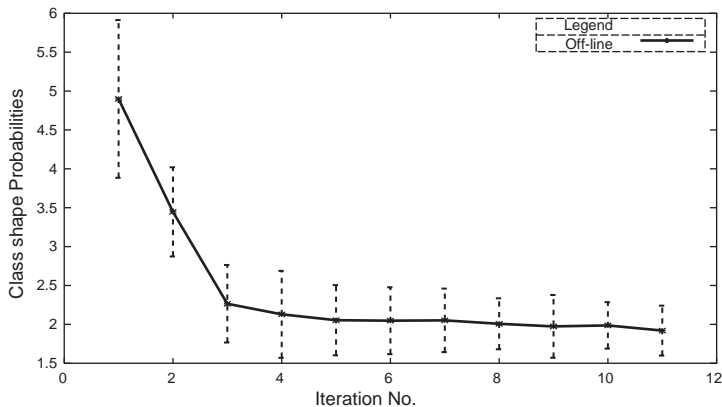


Fig. 3. Convergence rate for shape classes learnt off-line.

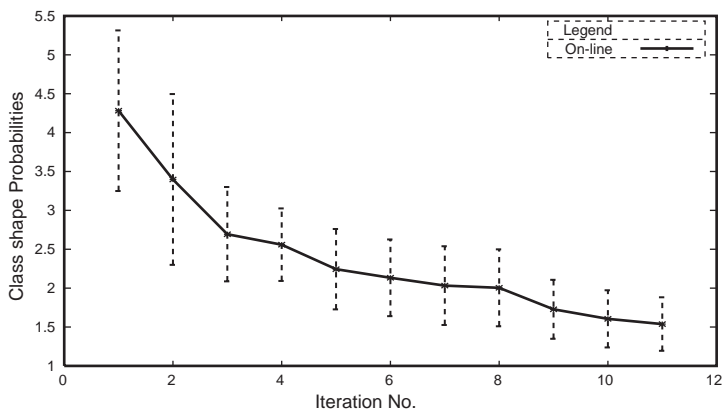


Fig. 4. Convergence rate for shape classes learnt on-line.

on-line and off-line learning. Table 2 summarises the results obtained in matching the mixture models obtained using the two different learning methods. Here we show the number of correct and incorrect class assignments for shapes drawn from seven different classes. The on-line method gives a slightly lower recognition rate than the off-line method.

In Fig. 6 we illustrate the fitting the shape-mixtures learnt with on-line and off-line update of the PDM eigenvectors. The different rows are for different test characters retained from the training-set. The left-hand column shows the original character data, the second column shows the final fit obtained with the offline method, while the third column shows the result obtained with the on-line method. There is little to distinguish the results. However, the off-line method takes 96 s to train while the on-line method takes only 41 s to train.

We now turn our attention to the results obtained when the shape-mixture learnt in training is used for the purposes of recognition by alignment. In Figs. 7 and 8, we respectively illustrate the fitting of a mixture models learnt using the

off-Line and on-Line methods. The mixture models have been fitted to a character retained from the training-set. The different images in the sequence show the fitted PDMs as a function of iteration number. The shape shown is the one with the largest a posteriori probability. There is little to distinguish the quality of the fitted shapes. In Fig. 9 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 Fig. 10 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.

In Fig. 11, we turn our attention to the iterative qualities of the alignment algorithm. Here we plot the average a posteriori class probabilities as a function of iteration number. The alignment of the mixture learnt using on-line

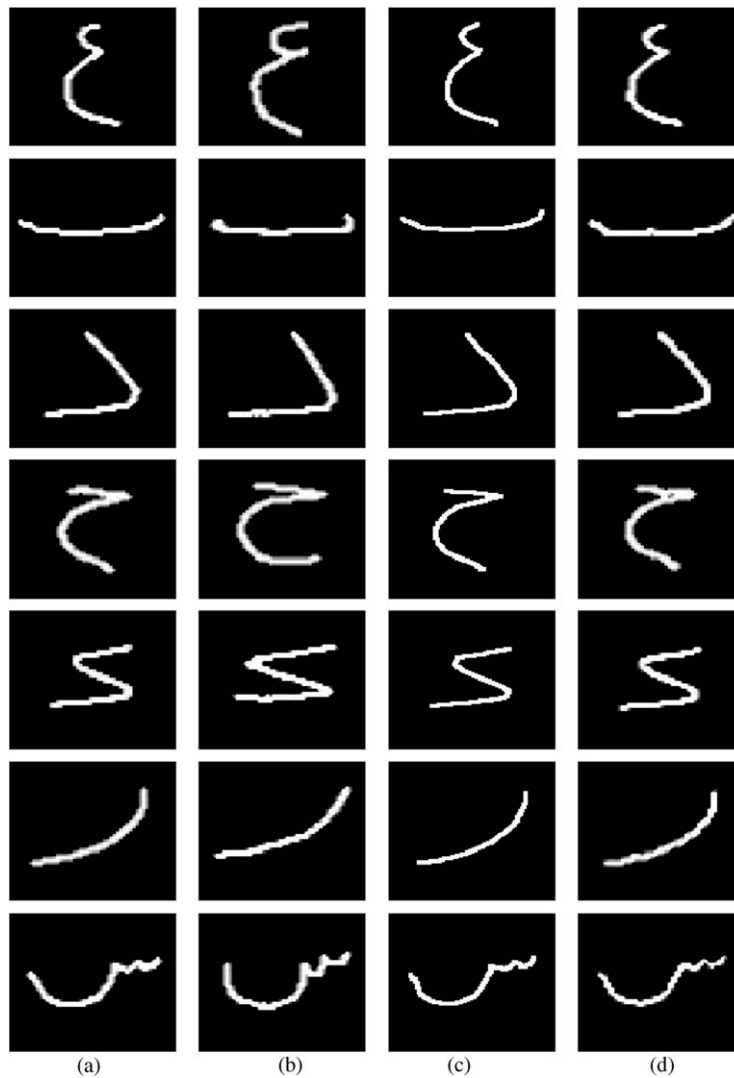


Fig. 5. (a) Actual mean shapes; (b) mean shapes used to initialize the EM; (c) off-line final mean shapes; (d) on-line final mean shapes.

Table 2
Recognition rate for shape-classes 1–7

Model No.	Samples	Single PDM		On-line PDMs		Off-line PDMs	
		Correct	Wrong	Correct	Wrong	Correct	Wrong
Shape-class 1	100	90	10	97	3	98	2
Shape-class 2	100	96	4	99	1	99	1
Shape-class 3	100	96	4	100	0	99	1
Shape-class 4	100	90	10	98	2	98	2
Shape-class 5	100	93	7	98	2	99	1
Shape-class 6	100	97	3	97	3	99	1
Shape-class 7	100	82	18	95	5	96	4
Recognition rate	700	92.0%	8.0%	97.7%	2.3%	98.3%	1.7%

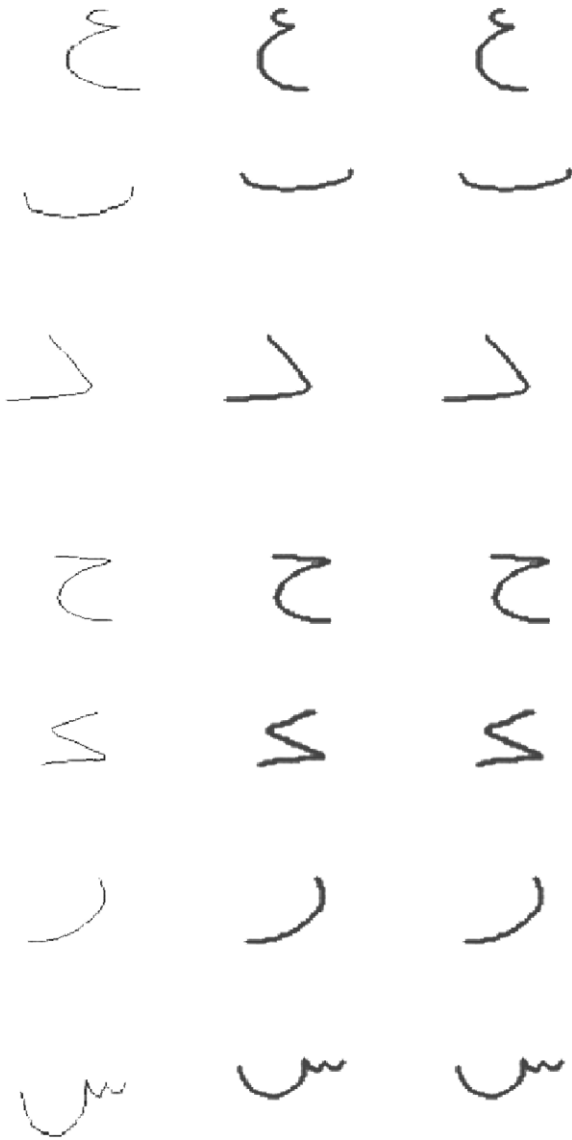


Fig. 6. Model alignment to data: (a) character; (b) off-line learning; (c) on-line learning.

method converges more rapidly than that learnt using the off-line method.

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 gives the numbers of characters recognised correctly and incorrectly for each of the shape-classes. We have compared the results obtained using the mixture model and those obtained using a single

PDM. The main conclusion to be drawn from the table is that the mixture of PDMs gives a better recognition rate than using separately trained single PDMs for each class. Hence, recognition can be improved using a more complex model of the shape-space.

We have also investigated the effect of point-position errors on the fitting process. Here we add random point-position errors to the individual landmark points. The errors are sampled from a circularly symmetric Gaussian distribution of zero mean and known variance. Fig. 12 shows the root mean squared error of the final aligned model as a function of the standard deviation of the Gaussian noise. There is an approximately linear relationship between the root mean squared error and the standard deviation of the added Gaussian noise. There is little to distinguish between results obtained with the on-line and off-line learning methods. To take this study one step further, we turn our attention to the point correspondences errors which result from the two methods. Fig. 13 shows the average correspondence error, i.e. the fraction of mismatched points, for a sample of fitted characters when Gaussian noise is added to the point-sets. The on-line method gives a slightly better correspondence errors than the off-line method. Both methods fail and locate correspondences to an incorrect pattern when the standard deviation of the point position error exceeds 30% of the interpoint distance.

Finally, we investigate the effect of different initialisation in the learning stage on the alignment stage. Fig. 14 shows the recognition rate of the patterns used for initialisation are displaced from their Procrustes centres. It is clear from the graph that on-line method performs better the off-line method. Both methods fail when the displacement is relatively large. We have extended this study to investigate the effect of rotation of the initial patterns from their Procrustes normalisation. Fig. 15 shows the recognition rate as a function of rotation angle for the on-line and off-line methods. It is apparent that the off-line method is less susceptible to rotation than the on-line methods. However, both methods fail when the rotation angle exceeds 50° .

6. Conclusion

In this paper, we have shown how mixtures of point-distribution models can be learned and then subsequently used for the purposes of recognition by alignment. We have described an efficient method for training point distribution models by iteratively updating the eigenvectors of the landmark covariance matrix. The gain in efficiency is demonstrated not to adversely affect the methods ability to represent and recognise variable shapes. 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.

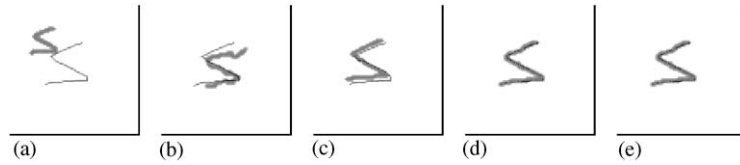


Fig. 7. Model alignment to data using off-line learning: (a) iteration 1; (b) iteration 2; (c) iteration 3; (d) iteration 5; and (e) iteration 7.

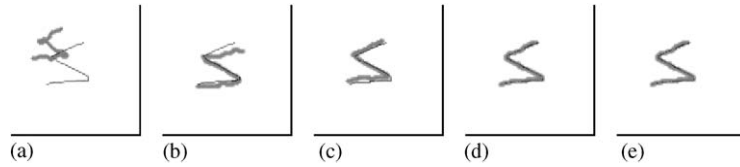


Fig. 8. Model alignment using on-line learning: (a) iteration 1; (b) iteration 2; (c) iteration 3; (d) iteration 5; and (e) iteration 7.



Fig. 9. Subdominant model alignment to data using mixture of PDMs with on-line learning.

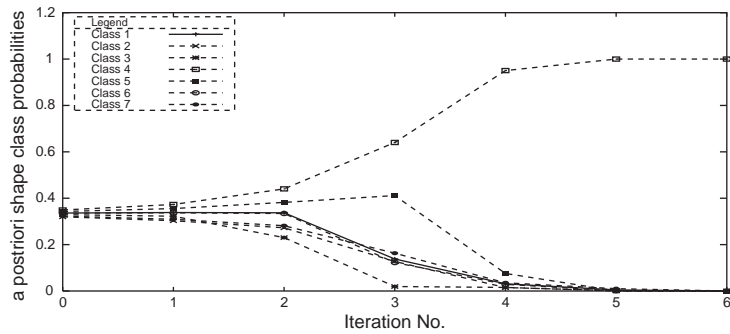


Fig. 10. Model fitting with mixture of PDMs.

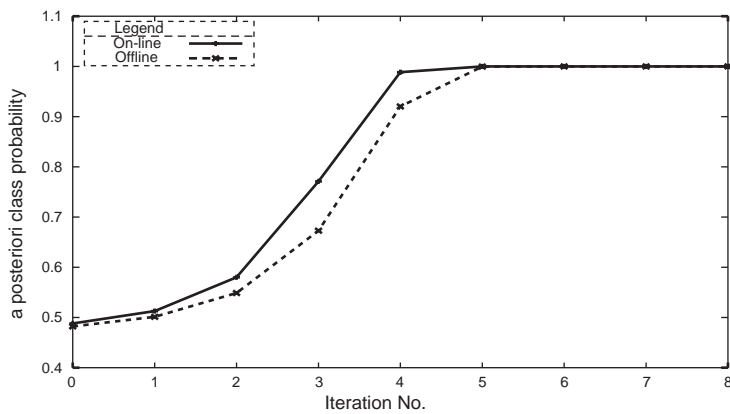


Fig. 11. A dominant shape hypothesis comparison as a function per iteration no. for on-line and off-line methods.

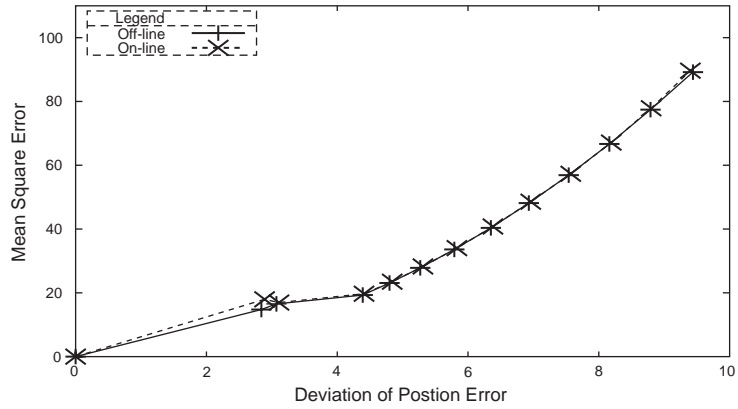


Fig. 12. Root mean square error as a function of standard deviation of Gaussian noise.

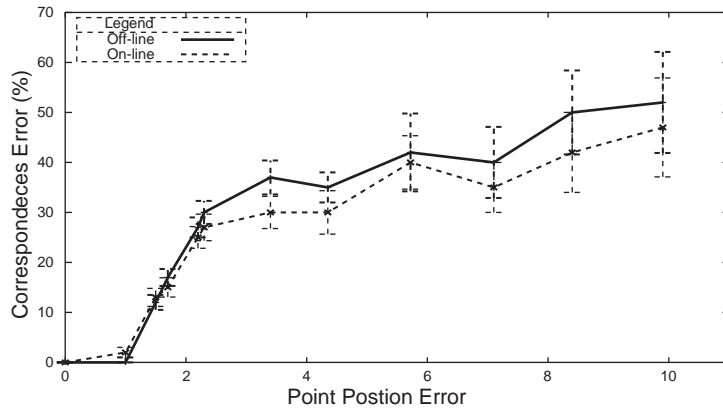


Fig. 13. Correspondence error as a function of the standard deviation of the Gaussian point position error.

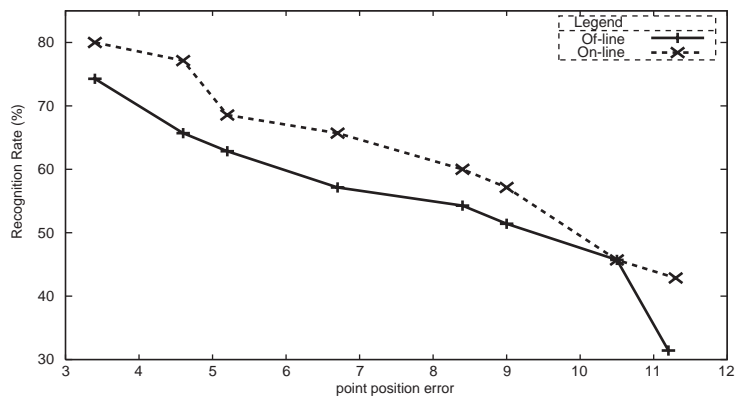


Fig. 14. Recognition rate as a function of initial pattern displacement.

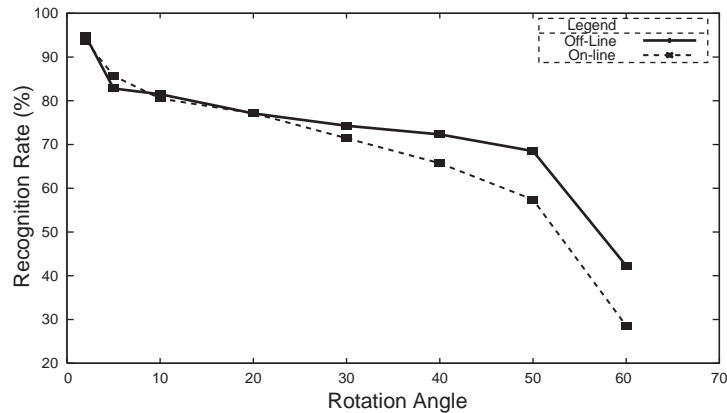


Fig. 15. Recognition rate as a function of rotation angle.

We present results on sets of Arabic characters. Here we show that the mixture of PDMs gives better performance than a single PDM. In particular we were able to capture more complex shape variations.

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. The study is in hand, and results will be reported in due course.

References

- [1] T. Cootes, C. Taylor, D. Cooper, J. Graham, Trainable method of parametric shape description, *Image Vision Comput.* 10 (5) (1992) 289–294.
- [2] S. Sclaroff, A. Pentland, Model matching for correspondence and recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* 17 (6) (1995) 545–561.
- [3] N. Duta, A. Jain, P. Dubuisson, Learning 2d shape models, *Int. Conf. Comput. Vision Pattern Recognition 2* (1999) 8–14.
- [4] T. Cootes, C. Taylor, D. Cooper, J. Graham, Active shape models—their training and application, *Comput. Vision Image Understanding* 61 (1) (1995) 38–59.
- [5] R. Duda, P. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.
- [6] J. Martin, A. Pentland, S. Sclaroff, R. Kikinis, Characterization of neuropathological shape deformations, *IEEE Trans. Pattern Anal. Mach. Intell.* 20 (2) (1998) 97–112.
- [7] R. Bowden, T. Mitchel, M. Sarhadi, Non-linear statistical models for the 3d reconstruction of human pose and motion from monocular image sequences, *Image Vision Comput.* 18 (9) (2000) 729–737.
- [8] A. Colmenarez, B.J. Frey, T.S. Huang, Mixtures of local linear subspaces for face recognition, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society Press, Los Alamitos, CA, 1998.
- [9] I.T. Jolliffe, *Principal Component Analysis*, Springer, Berlin, 1986.
- [10] M. Jordan, R. Jacobs, Hierarchical mixtures of experts and the em algorithm, *Neural Comput.* 6 (1994) 181–214.
- [11] S. Raudys, Shun-ichi Amari, Effect of initial values in simple perception, *Second World Congress on Computational Intelligence*, 1998, pp. 1530–1535.
- [12] S. Raudys, Prior weights in adaptive pattern classification, *15th International Conference on Pattern Recognition*, Barcelona, Spain, 2000, pp. 1014–1017.
- [13] J. Brendan, N. Jovic, Estimating mixture models of images and inferring spatial transformations using the em algorithm, *IEEE Comput. Vision Pattern Recognition 2* (1999) 416–422.
- [14] C. Bishop, J. Winn, Non-linear Bayesian image modelling, *Proceedings of Sixth European Conference on Computer Vision*, Vol. 1, 2000, pp. 3–17.
- [15] N. Vasconcelos, A. Lippman, A probabilistic architecture for content-based image retrieval, *Proceedings of International Conference on Computer Vision and Pattern Recognition*, 2000, pp. 216–221.
- [16] S. Rowe, A. Blake, Statistical mosaics for tracking, *Image Vision Comput.* 14 (8) (1996) 549–564.
- [17] B. North, A. Blake, Using expectation-maximisation to learn dynamical models from visual data, *Image Vision Comput.* 17 (8) (1999) 611–616.
- [18] Y. Weiss, E. Adelson, A unified mixture framework for motion segmentation: incorporating spatial coherence and estimating the number of models, *IEEE Comput. Vision Pattern Recognition* (1996) 321–326.
- [19] M. Revow, C. Williams, G.E. Hinton, Using generative models for handwritten digit recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* 20 (2) (1996) 592–606.
- [20] C. Goodall, Procrustes methods in the statistical analysis of shapes, *J. Roy. Statist. Soc. B* 53 (1991) 285–339.
- [21] T. Cootes, C. Taylor, Combining point distribution models with shape models based on finite element analysis, *Image Vision Comput.* 13 (5) (1995) 403–409.
- [22] T. Cootes, C. Taylor, A mixture models for representing shape variation, *Image Vision Comput.* 17 (1999) 403–409.
- [23] A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incomplete data via the em algorithm, *J. Roy. Statist. Soc. Ser. 39* (1977) 1–38.

About the Author—ABDULLAH A. AL-SHAHER received his B.Sc. degree in Computer Science and B.Sc. in Biology from Huston Tillotson College, Austin, TX, USA in 1987. In 1995 he was awarded an M.Sc. (honor list) in Computer Science from the University of Denver, Denver, CO. USA. From 1987 to 1993, he worked in the Computer Information Centre in Kuwait as head of the system development section. From 1995 to 1999 he worked as a lecturer in the Department of Computer Science for the Public Authority for Applied Education and Training also in Kuwait. Currently, he is a research student in the Computer Vision group in the Department of Computer Science at the University of York, York, England. His research interests include handwritten character recognition, statistical pattern recognition and neural networks.

About the Author—EDWIN HANCOCK studied physics as an undergraduate at the University of Durham and graduated with honours in 1977. He remained at Durham to complete a Ph.D. in the area of high energy physics in 1981. Following this he worked for 10 years as a researcher in the fields of high-energy nuclear physics and pattern recognition at the Rutherford-Appleton Laboratory (now the Central Research Laboratory of the Research Councils). During this period he also held adjunct teaching posts at the University of Surrey and the Open University.

In 1991 he moved to the University of York as a lecturer in the Department of Computer Science. He was promoted to Senior Lecturer in 1997 and to Reader in 1998. In 1998 he was appointed to a Chair in Computer Vision.

Professor Hancock now leads a group of some 15 faculty, research staff and Ph.D. students working in the areas of computer vision and pattern recognition. His main research interests are in the use of optimisation and probabilistic methods for high and intermediate level vision. He is also interested in the methodology of structural and statistical pattern recognition. He is currently working on graph-matching, shape-from-X, image data-bases and statistical learning theory. His work has found applications in areas such as radar terrain analysis, seismic section remote sensing and medical imaging. Professor Hancock has published some 60 journal papers and 200 refereed conference publications. He was awarded the Pattern Recognition Society medal in 1991 for the best paper to be published in the journal *Pattern Recognition*. The journal also awarded him an outstanding paper award in 1997.

Professor Hancock has been a member of the Editorial Boards of the journals *IEEE Transactions on Pattern Analysis and Machine Intelligence*, and, *Pattern Recognition*. He has also been a guest editor for special editions of the journals *Image and Vision Computing* and *Pattern Recognition*, and he is currently a guest editor of a special edition of *IEEE Transactions on Pattern Analysis and Machine Intelligence* devoted to energy minimisation methods in computer vision. He has been on the programme committees for numerous national and international meetings. In 1997 he established a new series of international meetings on energy minimisation methods in computer vision and pattern recognition.