

# Articulated Shape Mixtures for Object Recognition

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

## Abstract

This paper describes a probabilistic framework for recognising 2D shapes with articulated components. The shapes are represented using both geometrical and a symbolic primitives, that are encapsulated in a two layer hierarchical architecture. Each primitive is modelled so as to allow a degree of articulated freedom using a polar point distribution model that captures how the primitive movement varies over a training set. Each segment is assigned a symbolic label to distinguish its identity, and the overall shape is represented by a configuration of labels. We demonstrate how both the point-distribution model and the symbolic labels can be combined to perform recognition using a probabilistic hierarchical algorithm. This involves recovering the parameters of the point distribution model that minimise an alignment error, and recovering symbol configurations that minimise a structural error. We apply the recognition method to human pose recognition.

## 1 Introduction

The task of recognising articulated shapes has attracted considerable interest in computer vision. The main problem is how to robustly recover correspondence when the object under study undergoes deformations and the detected feature points defining the object are subject to noise. One of the most effective ways of developing matching techniques is to draw on probabilistic and statistical methods. This approach has led to the development of point distribution models [1], deformable templates [2] and condensation [3]. One of the most important modes of variation in a moving shape, especially biological forms, is that of articulation.

There are a number of ways in which object articulation can be modelled. Perhaps the simplest of these is to decompose the shape into a skeletal form, consisting of limbs or branches, and to model the articulation of the branches. The mechanical movement of the resulting shape can be captured by the rotation of the components. However, in order to constrain the change in shape to be physically realistic bounds, or distributions, must be imposed on the rotation angles [4, 5]. Hence, the mechanical constraints on articulation must be combined with a statistical model of limb movement. In addition to movement of the limbs, the articulated shape also has a structural composition, since the limbs can be assigned labels to distinguish them, and the arrangement of the labels can be used to provide further constraints for shape-recognition.

The aim in this paper is to develop a statistical framework that can be used to recognise articulated shapes using information concerning limb movement and symbolic constraints concerning the overall shape structure. To do this, we develop a hierarchical algorithm. Each shape is represented as an arrangement of articulated limbs. The movement of the

limbs is represented by a polar point distribution model. The structural component of the model is represented by a configuration of limb-labels. The recognition architecture has two intercommunicating layers. The first of these is concerned with limb alignment, and this aims to recover the lengths and polar angles of the limbs. The second aims to assign limb-labels so that the overall structure is recovered consistently. The fitting of this two-level model to data is effected using a variant of the expectation-maximisation algorithm.

## 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  $\{(x_1, y_1), \dots, (x_L, y_L)\}$ . Suppose that there are  $T$  landmark patterns. The  $t^{\text{th}}$  training pattern is represented using the long-vector of landmark co-ordinates  $X_t = (x_1, y_1, x_2, y_2, \dots, x_L, y_L)^T$ , 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

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

The eigenmodes of the landmark covariance matrix are used to construct the point distribution model. First, the unit eigenvalues  $E$  of the landmark covariance matrix are found by solving the eigenvalue equation  $|\Sigma - \alpha I| = 0$  where  $I$  is the  $2L \times 2L$  identity matrix. The eigen-vector  $\phi_i$  corresponding to the eigenvalue  $\alpha_i$  is found by solving the eigenvector equation  $\Sigma \phi_i = \alpha_i \phi_i$ . 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  $\Sigma$ . 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  $\Phi = (\phi_1 | \phi_2 | \dots | \phi_M)$ , where  $\alpha_1, \alpha_2, \dots, \alpha_M$  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  $\gamma$  is a vector of modal co-efficients. This vector represents the free-parameters of the global shape-model. When fitted to an observed set of landmark measurements  $X_o$ , the least-squares estimate of the parameter vector is

$$\gamma = \frac{1}{2} (\Phi + \Phi^T) (X_o - Y)$$

## 3 Shape Representation

Our aim is to use point distribution models to account for shape deformations due to limb articulation. The model is a two component one. First, we have a limb-model. This accounts for the variations in shape of each of the individual limbs using a point distribution model to describe the modes of variation of the landmark points about a mean shape. Second, we have a limb arrangement model. This is an augmented point distribution model that describes the arrangement of the centre points of the limbs, and their polar angles.

We are concerned with recognising 2D shapes by modelling segment movement around the centre of the shape. The shape under study is assumed to be segmented into a set of

$K$  jointed and non-overlapping limbs. The  $k^{th}$  limb is represented by a long-vector of consecutive landmark points

$$X_k = (x_1^k, y_1^k, x_2^k, y_2^k, \dots, x_{n_k}^k, y_{n_k}^k)^T$$

The centre-of-gravity of the limb indexed  $k$  is

$$\vec{c}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} (x_i^k, y_i^k)^T$$

The overall shape is represented by a long-vector of consecutive limb centres  $C = (\vec{c}_1^T, \vec{c}_2^T, \dots, \vec{c}_K^T)^T$ . The centre of articulated shape is computed by averaging the centre of the limbs

$$\vec{U} = \frac{1}{K} \sum_{k=1}^K \vec{c}_k$$

To model the articulated shape, we use a polar variant of the standard point distribution model [6]. This model allows the primitives to move about the centre of articulation. According to this model the shape is viewed as an arrangement of non-overlapping primitives. Each primitive is represented by mean point  $\vec{c}_k$ . Limb articulation is represented by a set of limb-angles. For the  $k^{th}$  limb the angle is defined to be

$$\theta_k = \tan^{-1} \frac{U(y) - c_k(y)}{U(x) - c_k(x)}$$

and the angular arrangement of the limbs is represented by the vector  $\Theta = (\theta_1, \theta_2, \dots, \theta_K)^T$ . The global movement of the limbs within a shape is specified by the concatenated long-vector of angles and the centres-of-articulation, i.e. by the vector  $S = (\Theta^T, C^T)^T$ .

To augment the geometric information, we assign symbols to the articulated components. Each training pattern is assigned to a shape class and each component primitive is assigned to a primitive class. The set of shape-labels is  $\Omega_c$  and the set of articulated component or limb labels is  $\Omega_s$ . The symbolic structure of each shape is represented by a permissible arrangement of limb-labels. For shapes of class  $\omega \in \Omega_c$  the permissible arrangement of limbs is denoted by  $\Lambda_\omega = \langle \lambda_1^\omega, \lambda_2^\omega, \dots \rangle$ .

## 4 Learning Mixtures of PDM's

In Cootes and Taylor's method [1], 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 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  $\Omega$ . Each shape-class  $\omega \in \Omega$ , has its own mean point-pattern  $Y_\omega$

and covariance matrix  $\Sigma_\omega$ . With these ingredients, the likelihood function for the set of training patterns is

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

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

$$Q_{n+1} = \sum_{t=1}^T \sum_{\omega \in \Omega_s} P(t \in \omega | X_t, Y_\omega^{(n)}, \Sigma_\omega^{(n)}) \ln p(X_t | Y_\omega^{(n+1)}, \Sigma_\omega^{(n+1)}) \quad (3)$$

where  $Y_\omega^{(n)}$  and  $\Sigma_\omega^{(n)}$  are the estimates of the mean pattern-vector and the covariance matrix for class  $\omega$  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  $\omega$  at iteration  $n$  of the algorithm. The probability density for the pattern-vectors associated with the shape-class  $\omega$ , 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

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

where

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

#### 4.1 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

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

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

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

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.

We apply this learning procedure separately to the landmark data for the individual limbs, and to the combined limb angle and limb-centre data. For the limb with label  $\lambda$ , the estimated modal matrix is  $\Phi_\lambda$  and the estimated parameter vector is  $\gamma_\lambda$ . For the shape-class with label  $\omega$ , on the other hand, the combined modal matrix for the articulation angles and limb-centres is  $\hat{\Phi}_\omega$  and the result of fitting to data is a parameter vector  $\Gamma_\omega$ . The

first  $K$  rows of  $\tilde{\Gamma}_\omega$  correspond to the limb angles, and the remaining  $2K$  to the long-vectors of limbs centres. However, we need to constrain the parameters corresponding to the limb angles. Suppose that the mean-vector for the limb-angles is  $\hat{\Theta}_w$  and the corresponding covariance matrix is  $\Sigma_w$ . The angular deformations are constrained to avoid flipping by limiting the deformation vector. We use the variance associated with the eigen-modes to constrain the deformation. The  $k^{\text{th}}$  component of the parameter vector is constrained to fall in the interval  $-3\sqrt{\alpha_k} \leq \Gamma_{(k)} \leq 3\sqrt{\alpha_k}$ . The articulation angles lie in the range  $-180^\circ$  to  $180^\circ$  to avoid discontinuities associated with the flip from  $0^\circ$  to  $360^\circ$ . A similar procedure for learning is followed to learn the variation in the polar representation of the limb and limb classes.

## 5 Hierarchical Architecture

With the limb-articulation and limb-centre point distribution models to hand, our recognition method proceeds in a hierarchical manner. Our aim is to classify the set of limb landmark long-vectors  $X = (\vec{z}_1, \dots, \vec{z}_k, \dots, \vec{z}_K)$  representing a test-shape. To commence, we make maximum likelihood estimates of the best-fit parameters of each limb-model to each set of limb-points. The best-fit parameters  $\gamma_\lambda^k$  of the limb-model with class-label  $\lambda$  to the set of points constituting the limb indexed  $k$  is

$$\gamma_\lambda^k = \arg \max_\gamma p(\vec{z}_k | \Phi_\lambda, \gamma) \quad (9)$$

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

$$l_k = \arg \max_{l \in \Omega_s} P(l | \vec{z}_k, \gamma_\lambda, \Phi_\lambda) \quad (10)$$

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 limb in turn. The class identity of the set of limbs is summarised by the string of assigned limb-labels  $L = \langle l_1, l_2, \dots \rangle$ . Hence, the input layer is initialised using maximum likelihood limb parameters and maximum a posteriori probability limb labels. The shape-layer takes this information as input. The goal of computation in this second layer is to refine the configuration of limb labels using global constraints on the arrangement of limbs to form consistent shapes. The constraints come from both geometric and symbolic sources. The geometric constraints are provided by the fit of a polar limbs point distribution model. The symbolic constraints are provide by a dictionary of permissible limb-label strings for different shapes.

The parameters of the limb-centre point distribution model are found using the EM algorithm [7]. Here we borrow ideas from the hierarchical mixture of experts algorithm [8], and pose the recovery of parameters as that of maximising a gated expected log-likelihood function for the distribution of limb-centre alignment errors  $p(X | \Phi_\omega, \Gamma_\omega)$ . The likelihood function is gated by two sets of probabilities. The first of these are the a posteriori probabilities  $P(\lambda_k^\omega | \vec{z}_k, \gamma_{\lambda_k^\omega}, \Phi_{\lambda_k^\omega})$  of the individual limbs. The second are the conditional probabilities  $P(L | \Lambda_\omega)$  of the assigned limb-label string given the dictionary of permissible configurations for shapes of class  $\omega$ . The expected log-likelihood function is given by

$$\mathcal{L} = \sum_{\omega \in \Omega_c} P(L | \Lambda_\omega) \left\{ \prod_k P(\lambda_k^\omega | \vec{z}_k, \gamma_{\lambda_k^\omega}, \Phi_{\lambda_k^\omega}) \right\} \ln p(X | \tilde{\Phi}_\omega, \Gamma_\omega) \quad (11)$$

The optimal set of polar limb arrangement parameters satisfies the condition

$$\Gamma_\omega^* = \arg \max_{\Gamma} P(L | \Lambda_\omega) \left\{ \prod_k P(\lambda_k^\omega | \vec{z}_k, \gamma_{\lambda_k^\omega}, \Phi_{\lambda_k^\omega}) \right\} \ln p(X | \tilde{\Phi}_\omega, \Gamma_\omega) \quad (12)$$

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

$$\omega^* = \arg \max_{\omega \in \Omega_c} P(\omega | X, \tilde{\Phi}_\omega, \Gamma_\omega^*) \quad (13)$$

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

$$l_k = \arg \max_{\lambda \in \Omega_s} P(\lambda | \vec{z}_k, \gamma_l^k, \tilde{\Phi}_\lambda) P(L(\lambda, k) | \Lambda_\omega) \quad (14)$$

Finally, the maximum likelihood parameters for the limbs are refined

$$\gamma_k = \arg \max_{\gamma} p(\vec{z}_k | \tilde{\Phi}_{l_k}, \gamma, \Gamma_\omega^*) \quad (15)$$

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

## 6 Models

To apply the model to shape-recognition, we require models of the alignment error process and the label error process.

### 6.1 Alignment errors

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

$$p(\vec{z}_k | \tilde{\Phi}_\lambda, \gamma_\lambda) = \frac{1}{2\pi\sigma} \exp \left[ -\frac{1}{2\sigma^2} (\vec{z}_k - \hat{X}_\lambda - \tilde{\Phi}_\lambda \gamma_\lambda)^T (\vec{z}_k - \hat{X}_\lambda - \tilde{\Phi}_\lambda \gamma_\lambda) \right] \quad (16)$$

where  $\sigma^2$  is the variance of the point-position errors which for simplicity are assumed to be isotropic. A similar procedure may be applied to estimate the parameters of the polar limb-angle distribution model.

### 6.2 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  $\Lambda$ . The Hamming distance is given by

$$H(L, \Lambda_\omega) = \sum_{i=1}^K \delta_{l_i, \lambda_i^\omega} \quad (17)$$

where  $\delta$  is the Dirac delta function. With the Hamming distance to hand, the probability of the assigned string of labels  $L$  given the dictionary item  $\Lambda$  is

$$P(L | \Lambda_\omega) = K_p \exp[-k_p H(L, \Lambda_\omega)] \quad (18)$$

where  $K_p = (1 - p)^K$  and  $k_p = \ln \frac{1-p}{p}$  are constants determined by the label-error probability  $p$ .

## 7 Experiment

We have evaluated our method on sets of images of people in different poses. Figure 1 shows an example of the images used in our experiments, and illustrates the steps used to abstract the human subject for the purposes of recognition. In figure 1a, we show the original subject, who is wearing white markers to distinguish the main control points, or limb-centres, of our model. Figure 1b shows the subject "skeleton", and figure 1c the set of strokes constituting the shape. In figure 2 we show sets of example "skeletons" from a number of distinct body poses which we have used for the purposes of training. In total we have collected images of 14 distinct poses, and there are 20 examples of each pose. Each example is composed of 13 limb segments, and each limb-segment centre is identified by a marker. In Figure 3 we show four examples of the mean-shapes recovered as the output of the learning stage.

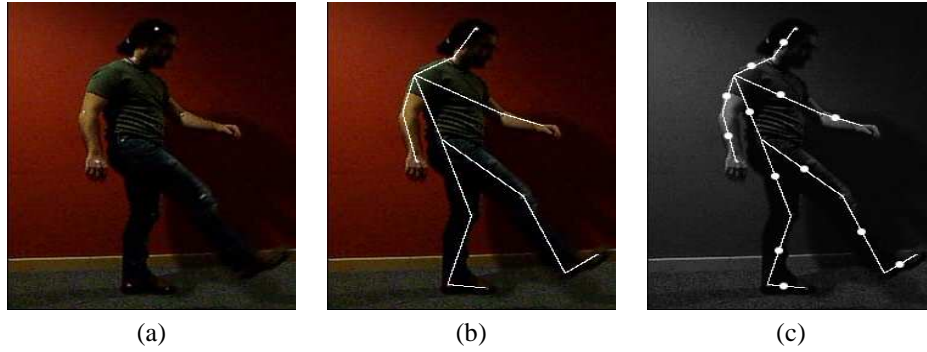


Figure 1: Shape Representation. (a) shooting; (b) extracted shape limbs; (c) shape limbs center

To evaluate recognition performance, we have used 1200 images corresponding to different pose classes for testing. Figure 4 show the learned shape models iteratively aligning to the “Stretching” pose sample. The figure illustrates how the model adapts in a flexible manner to fit the example shape in a relatively small number of iterations. In the figure the test-shape is shown in black lines while the model is shown in red.

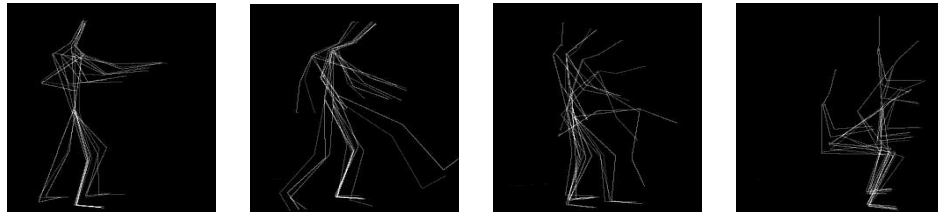


Figure 2: Training Sets

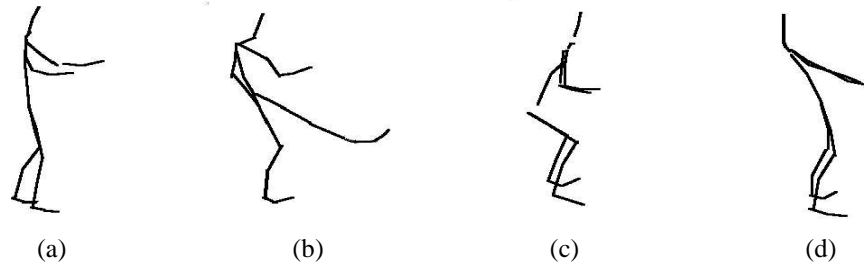


Figure 3: Learnt Shapes:(a) Boxing, (b) Kicking, (c) Relaxing, (d) Stretching

To illustrate the effectiveness of the recognition method when the input is confused, we have experimented with a test shape that overlaps two model shapes. In this example the test shape can fall into either the “Kicking” or “Shooting” classes. Figure 5 shows the alignment process. Initially, both the “Kicking” and “Shooting” hypotheses have low a *posteriori* probability. However, after several iterations the “Kicking” model dominates the alternative hypothesis.

To explore the capacity of the method to identify clusters of poses, we have applied principal components analysis and multidimensional scaling to the vectors of fitted model parameters extracted using our recognition method. Figures 6 and 7 respectively show the result of applying MDS and PCA to the alignment parameters for 54 sample shapes drawn at random from the set of shape-classes. In the top row of the two figures, we show the

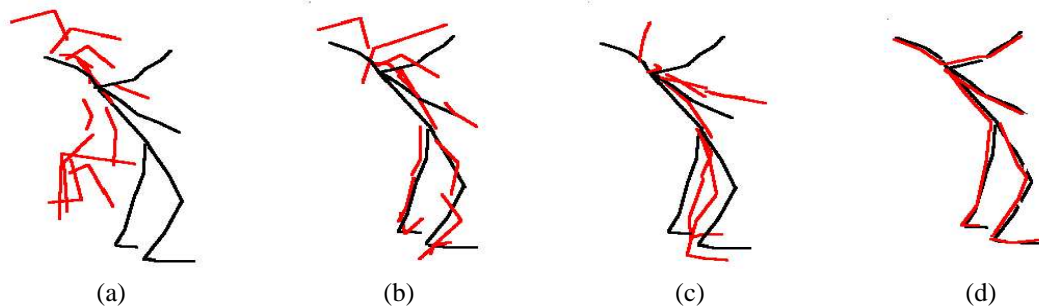


Figure 4: Stretching Alignment:(a)iteration 1,(b)iteration 2,(c)iteration 3,(d)iteration 5

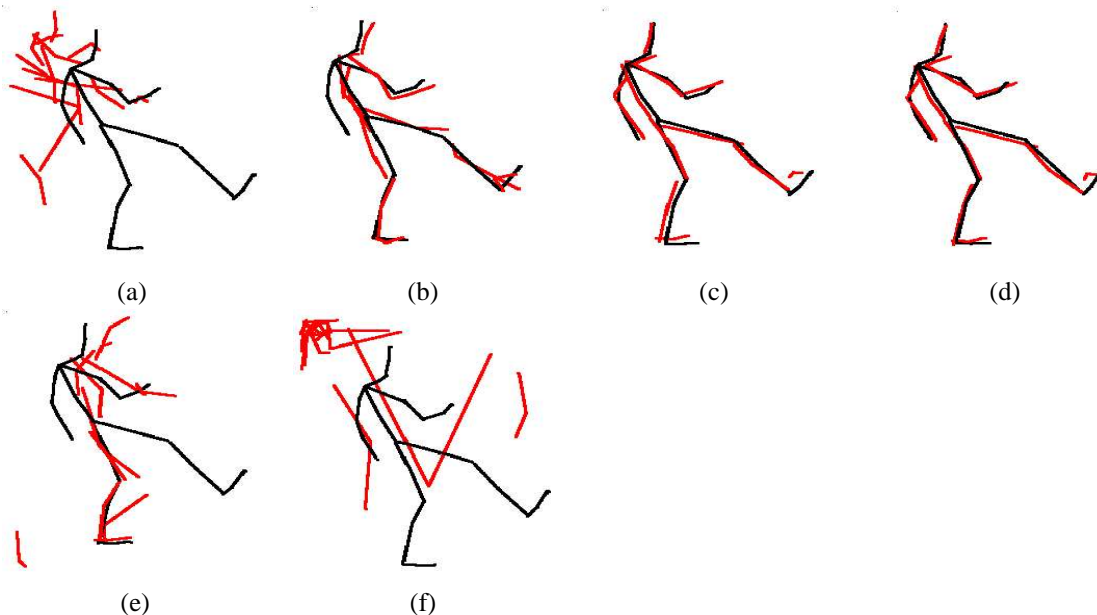


Figure 5: Alignment: Kicking: (a) iteration 1, (b) iteration 2, (c) iteration 3, (d) iteration 4; Shooting: (e) iteration 1, (f) iteration 2

result of projecting the data onto the leading three eigenvectors. In the bottom row of the two plots, we show the matrix of pairwise distances computed from the projection of the data into the three dimensional eigenspace. In the left-hand column of each plot, we show the results obtained with the initial parameter vectors, while the second column shows the results with the parameter vectors at convergence. The main effect of iterating the recognition method is to improve the clusters of shapes. This is reflected both by the distribution of data in the eigenspace, and the block structure in the pairwise distance matrices. Moreover, a better cluster structure emerges when MDS is used rather than PCA. However, a deeper analysis of the data reveals that PCA gives a better cluster structure when the shapes are subject to scaling.

To take this investigation one step further, Figure 8 shows the effect of adding random noise to the patterns. Here we apply MDS to the Euclidean distances between the final reconstructed shapes using the model parameters obtained at the final iteration of the



fitting algorithm. The left-hand panel shows the MDS embedding of 6 shapes with 15 examples each. Here the different shape classes are shown in different colours. The right-hand panel shows the pairwise distance matrix. It is clear that the method produces good shape clusters even under conditions of noise.

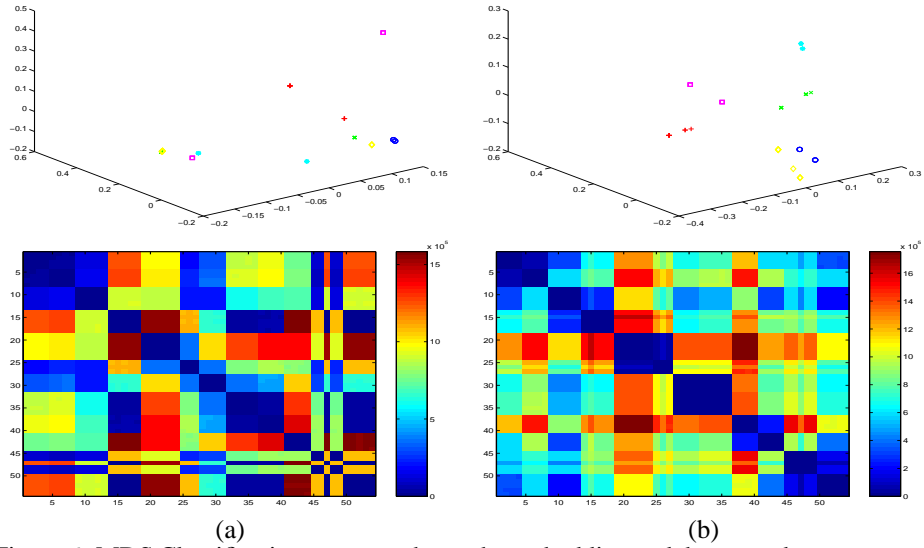


Figure 6: MDS Classification: top row shows the embedding and the second row represent distance matrix;(a) iteration 1, (b) final iteration

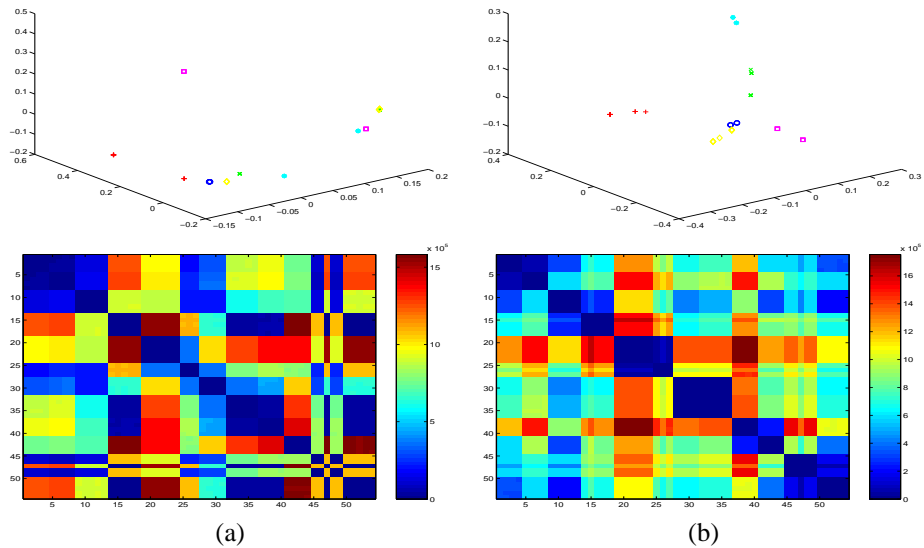


Figure 7: PCA Classification: top row shows the embedding and the second row shows the distance matrix; (a) iteration 1, (b) final iteration

Table 1 shows the recognition rates for six shape classes. The overall correct recognition rate is 93.16%. The poorest recognition occurs for the kicking, the picking and the shooting classes. Since these classes share similar limb configuration, we can conclude that recognition is reasonably high.

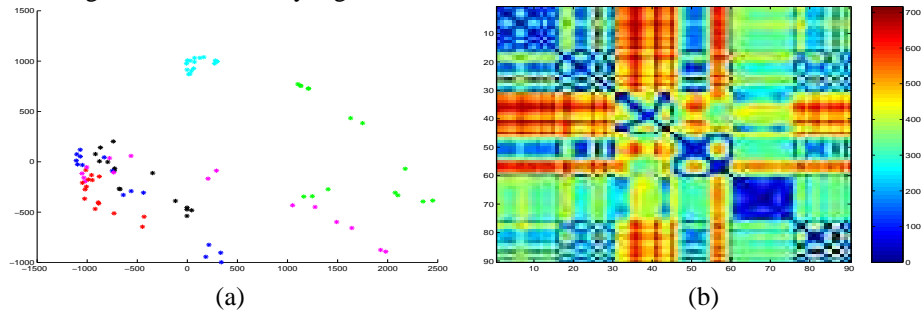


Figure 8: MDS data-model distances. (a) 6 clusters of 90 shapes, (b) dissimilarity matrix

**Table 1. Recognition rate for shape classes**

Shape	Boxing	Kicking	Relaxing	Picking	Stretching	Shooting	Rec. R.
Sample	200	200	200	200	200	200	
Correct	198	163	197	181	200	179	93.16%
Wrong	2	37	3	19	0	21	6.84%

## 8 Conclusion

In this paper, we have described a method for fitting articulated shape-models to landmark point data. The shape deformation process adopted is based on point distribution models. The model representation is a hierarchical one. There is a Cartesian deformation model for the limbs and the limb-centres, together with a polar model which represents limb articulation. We develop a probabilistic framework for fitting a mixture of articulated models to data. The method delivers good results of human shape modelling.

## References

- [1] T. Cootes and C. Taylor. Combining point distribution models with shape models based on finite element analysis. *Image and Vision Computing*, 13(5):403–409, 1995.
- [2] N. Duta, A. Jain, and P. Dubuisson. Learning 2D Shape Models. *International Conference on Computer Vision and pattern Recognition*, 2:8–14, 1999.
- [3] Michael Isard; Andrew Blake. Contour tracking by stochastic propagation of conditional density. In *Proc. European Conf. on Computer Vision*, pages 343–356, 1996.
- [4] J. Gonzales, J. Varona, F. Roca, and J. Villanueva. aspace: Action space for recognition and synthesis of human actions. *2nd IWAMDO, Palma de Mallorca, Spain*, pages 189–200, 2002.
- [5] James M. Rehg and Takeo Kanade. Visual tracking of high dof articulated structures: an application to human hand tracking. In *Third ECCV, Stockholm, Sweden*, pages 35–46, 1994.
- [6] T. Heap and D. Hogg. Extending the point distribution model using polar coordinates. *Image and Vision Computing*, 14:589–599, 1996.
- [7] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of Royal Statistical Soc. Ser.*, 39:1–38, 1977.
- [8] M. Jordan and R. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6:181–214, 1994.
- [9] Edwin R. Hancock and Josef Kittler. Edge-labelling using dictionary-based relaxation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 12(2):165–181, 1990.