A Mixture Model for Representing Shape Variation *

T.F. Cootes and C.J. Taylor
Dept. Medical Biophysics,
University of Manchester,
Oxford Road,
Manchester M13 9PT, UK

Abstract

The shape variation displayed by a class of objects can be represented as a probability density function, allowing us to determine plausible and implausible examples of the class. Given a training set of example shapes we can align them into a common co-ordinate frame and use kernel based density estimation techniques to represent this distribution. Such an estimate is complex and expensive, so we generate a simpler approximation using a mixture of gaussians. We show how to calculate the distribution, and how it can be used in image search to locate examples of the modelled object in new images.

1 Introduction

Deformable models have proved effective for interpreting images of objects whose shape can vary [2]. Where a training set of example images is available, a successful approach is to build a statistical model of the shape variation seen in the training set. Such a model can be used for image search to locate objects in new images. A good model will be sufficiently general that it can fit to valid unseen examples, but *specific* in that it will not allow significantly different shapes.

Here we deal with the case where we can place n landmark points repeatably on each example object (for instance around the boundary), and thus represent a shape by this set of landmarks $\{(x_i, y_i)\}$ [2]. Given a set of such shapes, aligned into a common co-ordinate frame, each shape corresponds to a vector $\mathbf{x} = (x_1, \ldots, x_n, y_1, \ldots, y_n)^T$ in a $2n$ dimensional space. The set of shapes then forms a cloud of points in this space, which can be thought of as drawn from a probability distribution. If we can estimate the probability density function (*p.d.f.*) $p(\mathbf{x})$ for the distribution of shapes, we can decide whether any new shape is plausible, and can use this information when attempting to locate examples of the object in new images.

*This paper appears in: Image and Vision Computing 17, No.8, 1999, pp 567-574

A general approach is to use a density estimation technique such as the kernel method [9]. This represents the distribution as a sum of gaussians, one placed at every original data point. However, when there are many points this becomes far too expensive (in both time and memory). It is necessary to further approximate the distribution, for instance using a mixture of a small number of gaussians, which can be fit to the kernel estimate using a modification of the Expectation Maximisation (EM) algorithm.

We have previously described Active Shape Model (ASM) search - an efficient approach to interpreting images containing known objects represented by statistically defined deformable templates. The method is iterative; at each step an initial hypothesis is deformed using the image evidence, then regularized to the nearest plausible shape (as defined by the p.d.f.). Given a 'good enough' starting point, this can converge rapidly to locate objects in new images. The method was originally intended for use with PDMs, but can be extended to use mixture models.

In the following we will demonstrate how mixtures of gaussians can be used to approximate the p.d.f. for a shape model, and how the Active Shape Model approach can be used to find objects in new images.

2 Background

There have been several attempts to derive models of the form $\mathbf{x} = f(\mathbf{b})$ which can approximate any example, $\mathbf{x}$, of a class of objects using a small number of shape parameters, $\mathbf{b}$, which are assumed to be independent. New plausible examples can be generated by choosing new values for the parameters $\mathbf{b}$ within certain limits, derived from the training set. The simplest of such models is the 'Point Distribution Model', in which a principal component analysis is applied to the data to pick out the main linear modes of shape variation. In this case $\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b}$, where $\mathbf{x}$ is an example of a shape, $\bar{\mathbf{x}}$ is the mean shape over the training set and $\mathbf{P}$ is a matrix containing the first t eigenvectors of the covariance matrix of the training set.

This model works well for a wide variety of examples [2], but cannot adequately represent non-linear shape variations, such as those generated when parts of the object rotate, or there are changes in viewing position of a 3D object. There have been several non-linear extensions to the PDM, either using polynomial modes [11], using a multi-layer perceptron to perform non-linear PCA [10] or using polar co-ordinates for rotating sub-parts of the model [5].

However, all these approaches assume that varying the parameters $\mathbf{b}$ within given limits will always generate plausible shapes, and that all plausible shapes can be so generated. This is not always the case. For instance, if a sub-part of the shape can appear in one of two positions, but not in-between, then the distribution has two separate peaks, with an illegal space in between. Without imposing more complex constraints on the parameters $\mathbf{b}$, models of the form $\mathbf{x} = f(\mathbf{b})$ are likely to generate illegal shapes.

The method presented below can model distinct classes of shape as well as non-linear shape variation, and does not require any labelling of the class of each training example.

3 Modelling Shape Variation

Suppose we have a set of N shapes, each labelled with n landmark points (x_i, y_i) in such a way that the i^{th} point always represents a particular position on the shape [2]. A single shape can be represented as the $2n$ dimensional vector $\mathbf{x} = (x_1, \ldots, x_n, y_1, \ldots, y_n)^T$. To compare shapes they must be aligned into a common co-ordinate frame.

3.1 Aligning a Set of Shapes

There is considerable literature on methods of aligning shapes into a common co-ordinate frame, the most popular approach being Procrustes Analysis [4]. This aligns each shape so that the sum of distances of each shape to the mean ($D = \sum |\mathbf{x}_i - \bar{\mathbf{x}}|^2$) is minimised. It is poorly defined unless constraints are placed on the alignment of the mean (for instance, ensuring it is centred on the origin, has unit scale and some fixed but arbitrary orientation). Though analytic solutions exist to the alignment of a set, a simple iterative approach is as follows: First align all the shapes with one of the examples, and calculate an initial estimate of the mean. Apply constraints on the mean (eg on its scale and c.o.g.). Then repeatedly re-align the shapes with the mean and recalculate it, until convergence. The operations allowed during the alignment will affect the shape of the final distribution. A common approach is to centre each shape on the origin, scale each so that $|\mathbf{x}| = 1$ and then choose the orientation for each which minimises D. The scaling constraint means that the aligned shapes $\mathbf{x}$ lie on a hyper-sphere, which can introduce significant non-linearities if large shape changes occur. For instance, Figure 1(a) shows the corners of a set of rectangles with varying aspect ratio, aligned in this fashion. The scale constraint ensures all the corners lie on a circle about the origin. A linear change in the aspect ratio introduces a non-linear variation in the point positions. An alternative approach is to allow both scaling and orientation to vary when minimising D.

To align two shapes, $\mathbf{x}_1$ and $\mathbf{x}_2$, each centred on the origin ($\mathbf{x}_1.\mathbf{1} = \mathbf{x}_2.\mathbf{1} = 0$), we choose a scale s and rotation θ so as to minimise $|s\mathbf{A}\mathbf{x}_1 - \mathbf{x}_2|$, where $\mathbf{A}$ performs a rotation of a shape $\mathbf{x}$ by θ. Let

$$\begin{array}{c} a = (\mathbf{x}_1.\mathbf{x}_2)/|\mathbf{x}_1|^2 \\ b = \left(\sum_{i=1}^{n}(x_{1i}y_{2i} - y_{1i}x_{2i})\right)/|\mathbf{x}_1|^2 \end{array} \quad (1)$$

Then $s^2 = a^2 + b^2$ and $\theta = \tan^{-1}(b/a)$. If the shapes do not have C.o.G.s on the origin, the optimal translation is chosen to match their C.o.G.s, the scaling and rotation chosen as above.

If this approach is used to align the set of rectangles, (Figure 1(b)), their corners lie on circles offset from the origin. A third approach is to rotate and scale each shape into the *tangent space* to the mean so as to minimise D. The tangent space to $\mathbf{x}_t$ is the hyperplane of vectors normal to $\mathbf{x}_t$, passing through $\mathbf{x}_t$. ie All the vectors $\mathbf{x}$ such that $(\mathbf{x}_t - \mathbf{x}).\mathbf{x}_t = 0$, or $\mathbf{x}.\mathbf{x}_t = 1$ if $|\mathbf{x}_t| = 1$. Figure 1(c) demonstrates that for the rectangles this leads to the corners varying along a straight line, preserving the linear nature of the shape variation. The simplest way to achieve this is to align the shapes with the mean, allowing scaling and rotation, then project into the tangent space by scaling $\mathbf{x}$ by $1/(\mathbf{x}.\bar{\mathbf{x}})$.

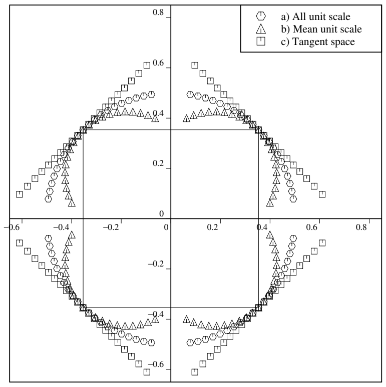

Figure 1: Aligning rectangles with varying aspect ratios. a) All shapes set to unit scale, b) Scale and angle free, c) Align into tangent space

Since we normalise the scale and orientation of the mean at each step, the mean of the shapes projected into the tangent space, $\bar{\mathbf{x}}$, may not be equal to the (normalised) vector defining the tangent space, $\mathbf{x}_t$. We must retain $\mathbf{x}_t$ so that when new shapes are studied, they can be projected into the same tangent space as the original data.

Different approaches to alignment can produce different distributions of the aligned shapes. We wish to keep the distribution compact and keep any non-linearities to a minimum, so use the tangent space approach in the following.

3.2 Density Estimation

The kernel method of density estimation [9] gives an estimate of the p.d.f. from which N samples, $\mathbf{x}_i$, have been drawn as

$$p(\mathbf{x}) = \sum_{i=1}^{N} \frac{1}{Nh^d} K(\frac{\mathbf{x} - \mathbf{x}_i}{h}) \tag{2}$$

where $K(\mathbf{t})$ defines the shape of the kernel to be placed at each point, h is a smoothing parameter defining the width of each kernel and d is the dimension of the data. In general, the larger the number of samples, the smaller the width of the kernel at each point. We use a gaussian kernel with a covariance matrix equal to that of the original data set, $\mathbf{S}$, ie $K(\mathbf{t}) = N(\mathbf{t} : \mathbf{0}, \mathbf{S})$. The optimal smoothing parameter, h, can be determined by cross-validation [9].

3.2.1 The Adaptive Kernel Method

The adaptive kernel method generalises the kernel method by allowing the scale of the kernels to be different at different points. Essentially, broader kernels are used in areas of low density where few observations are expected. The simplest approach is as follows:

1. Construct a pilot estimate $p'(\mathbf{x})$ using (2).
2. Define *local bandwidth factors* $\lambda_i = (p'(\mathbf{x}_i)/g)^{-\frac{1}{2}}$, where g is the geometric mean of the $p'(\mathbf{x}_i)$
3. Define the *adaptive kernel estimate* to be

$$p(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N}(h\lambda_i)^{-d}K(\frac{\mathbf{x}-\mathbf{x}_i}{h\lambda_i}) \tag{3}$$

3.3 Approximating the PDF from a Kernel Estimate

The kernel method can give a good estimate of the distribution. However, because it is constructed from a large number of kernels, it can be too expensive to use the estimate in an application. We wish to find a simpler approximation which will allow $p(\mathbf{x})$ to be calculated quickly.

We will use a weighted mixture of m gaussians to approximate the distribution derived from the kernel method.

$$p_{mix}(\mathbf{x}) = \sum_{j=1}^{m} w_j N(\mathbf{x} : \mu_j, \mathbf{S}_j) \tag{4}$$

where $N(\mathbf{x} : \mu, \mathbf{S})$ is the p.d.f. of a gaussian with mean μ and covariance $\mathbf{S}$.

Such a mixture can approximate any distribution up to arbitrary accuracy, assuming sufficient components are used. The hope is that a small number of components will give a 'good enough' estimate. The Expectation Maximisation (EM) algorithm [8] is the standard method of fitting such a mixture to a set of data. However, if we were to use as many components as samples ($m = N$), the optimal fit of the standard EM algorithm is to have a delta function at each sample point. This is unsatisfactory. We assume that the kernel estimate, $p_k(\mathbf{x})$ is in some sense an optimal estimate, designed to best generalise the given data. We would like $p_{mix}(\mathbf{x}) \rightarrow p_k(\mathbf{x})$ as $m \rightarrow N$.

A good approximation to this can be achieved by modifying the M-step in the EM algorithm to take into account the covariance about each data point suggested by the kernel estimate (see Appendix A).

The number of gaussians used in the mixture should be chosen so as to achieve a given approximation error between $p_k(\mathbf{x})$ and $p_{mix}(\mathbf{x})$.

3.4 Estimating the PDF for a Set of Shapes

Suppose we have a set of N shapes, represented as $2n$ dimensional vectors, $\mathbf{X}_i$. We wish to estimate their probability density function. We first align them into a common co-ordinate frame, giving a set of aligned shapes, $\mathbf{x}_i$, and a vector, $\bar{\mathbf{x}}_c$, defining the tangent space in which they dwell (see Section 3.1). We then project $\mathbf{x}_i$ into a lower dimensional space by applying principal component analysis (PCA). If we compute the eigenvalues, λ_j, of the covariance matrix of the data, only the first t will be large enough to be considered significant. For instance, if the noise on the measurements of the point positions has a variance of σ_n^2, then we choose the largest t such that $\lambda_t > \sigma_n^2$, assuming that the eigenvalues are sorted into descending order. The eigenvectors corresponding to these eigenvalues span the subspace containing most of the variation in the shapes. We can approximate each $2n$-vector, $\mathbf{x}$, using the t-vector, $\mathbf{b}$, given by

$$\mathbf{b} = \mathbf{P}^T(\mathbf{x} - \bar{\mathbf{x}}) \tag{5}$$

where $\mathbf{P}$ is the ($2n$ x t) matrix of the first t eigenvectors.

We can estimate $p(\mathbf{b})$ using a mixture model approximation to an adaptive kernel estimate, as described above.

3.5 A Synthetic Example

Suppose we wish to model the shape variation exhibited in the training set given in Figure 2. Here 28 points are used to represent a triangle rotating inside a square (there are 3 points along each line segment). If we apply PCA to the data, we find there are two significant components. Projecting the 100 original shapes $\mathbf{x}$ into the 2-D space of $\mathbf{b}$ (using (5)) gives the distribution shown in Figure 3. Figure 4 shows the p.d.f. estimated for this using the adaptive kernel method, with the initial h estimated using cross-validation. The desired number of components can be obtained by specifying an acceptable approximation error. Figure 5 shows the estimate of the p.d.f. obtained by fitting a mixture of 12 gaussians to the data.

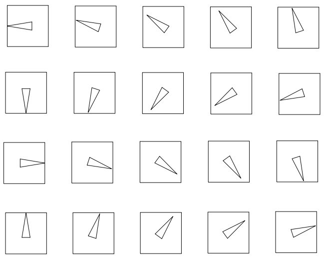

Figure 2: Examples from training set of synthetic shapes

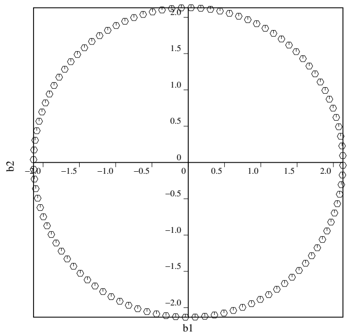

Figure 3: Distribution of $\mathbf{b}$ for 100 synthetic shapes

4 Image Search

Given an estimate of the p.d.f. for a class of shapes, $p(\mathbf{x})$, we can use it to help locate examples of the class in new images. If a reasonably good initial approximation is available, we can use local search to optimise the fit of a model to the data. Here we describe a modification to the Active Shape Model (ASM) framework [2] which allows search given the p.d.f. of the target shape.

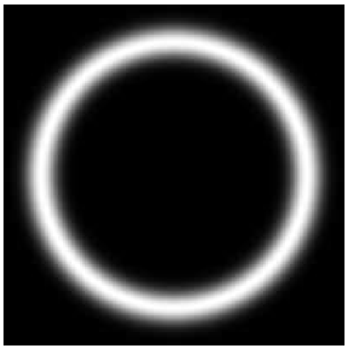

Figure 4: Plot of pdf estimated using the adaptive kernel method

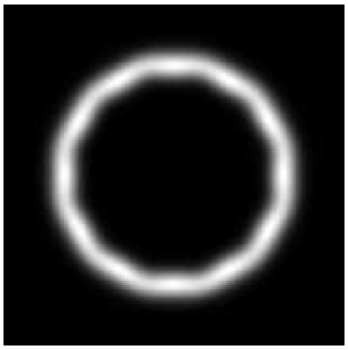

Figure 5: Plot of pdf approximation using mixture of 12 gaussians

Suppose that the shape of the object is represented by a set of points given by $\mathbf{x}$ in the model coordinate frame, and by $\mathbf{X} = M_{(s,\theta,\mathbf{t})}(\mathbf{x})$ in the image, where $M_{(s,\theta,\mathbf{t})}(.)$ rotates by θ, scales by s and translates by $\mathbf{t} = (t_x, t_y)^T$. Given an initial estimate, the ASM iterates through the following steps:

- For each model point (X_i, Y_i) look nearby in the image for a better position (X'_i, Y'_i).
- Update the model shape and the pose parameters to find the nearest plausible shape to $\mathbf{X}'$.

In the simplest case the first step involves finding the nearest strong edge along a normal through the current point. Better results can be obtained by using a training set to build a statistical model to represent the image structure expected at each point. During search we use the model at each point to find the best nearby match [3].

In the second step we first update M to minimise the errors in the image plane, $|\mathbf{X}' - M(\mathbf{x})|^2$. We then invert M to project $\mathbf{X}'$ into the model frame using $\mathbf{x}' = M^{-1}(\mathbf{X}')$.

If our model space is in the tangent space to some vector $\mathbf{x}_t$, then $M(\mathbf{x})$ is defined to rotate, translate and scale the (scale free) shape $\mathbf{x}/|\mathbf{x}|$. The inverse $M^{-1}(\mathbf{X})$ applies the inverse pose transformation, then projects the result into the tangent space. This ensures scale is defined only by the transformation M, and is kept independent of shape changes.

If $\mathbf{x}'$ is 'plausible' then we continue with it as our new shape estimate. If it is not, we must find the nearest shape which *is* plausible. This requires a definition of 'plausible', and a method of finding the nearest plausible shape to any given example.

We define the shape $\mathbf{x}$ as plausible if $p(\mathbf{x}) \geq p_t$, where p_t is chosen so that 99% of samples drawn from the p.d.f. pass the threshold. This can be determined stochastically, for instance by drawing 2000 examples from the distribution, ranking the value of $p(\mathbf{x})$ at each and setting p_t equal to an average about the 20^{th} smallest value.

4.1 Finding the Nearest Plausible Shape

If $p(\mathbf{x}) < p_t$ we wish to move $\mathbf{x}$ to the nearest point at which it is considered plausible. In practice this is difficult to locate, but an acceptable approximation can be obtained by gradient ascent - simply move uphill until the threshold is reached. The gradient of (4) is straightforward to compute, and suitable step sizes can be estimated from the distance to the mean of the nearest mixture component.

In most cases we will have built the mixture model in the t dimensional space, $p(\mathbf{b})$, using (5) to compute $\mathbf{b}$ for each $\mathbf{x}$. This projection acts to limit the allowed shape variation to a linear combination of t modes. Applying the density threshold further constrains the allowed shapes. Given an initial shape $\mathbf{x}$, we project into this lower dimensional space, $\mathbf{b}$, apply gradient ascent on $\mathbf{b}$ to find the nearest point which passes the threshold, then project back into the original space using $\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b}$.

For instance, Figure 6 shows an example of the synthetic shape with its points perturbed by noise. Figure 7 shows the result of projecting into the space of $\mathbf{b}$ and back. There is significant reduction in noise, but the triangle is unacceptably large compared with examples in the training set. Figure 8 shows the shape obtained by gradient ascent to the nearest plausible point using the 12 component mixture model estimate of $p(\mathbf{b})$. The triangle is now similar in scale to those in the training set.

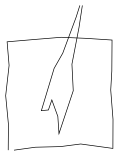

Figure 6: Shape with noise

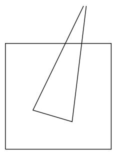

Figure 7: Projection into $\mathbf{b}$-space

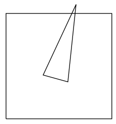

Figure 8: Nearby plausible shape

Given a good enough starting point, the search algorithm will converge. In practice, knowing the approximate position, scale and orientation is enough. The search can be attempted once for each mixture component, starting with the shape defined by the mean of each component. The method can be made more efficient and robust using multi-resolution techniques [3], first searching on a coarse scale image and refining on finer and finer scale images.

5 Example: Brain Stem Model

We have used the above approach to generate a model of the appearance of the brain stem in successive slices of an MR image of the head. Figure 9 shows a set of contours of a brain stem in sequential slices in a single brain image. (Figure 13 shows an example image). We built a shape model from 153 different slices from 10 different people. We aligned the shapes, performed PCA to lower the dimension and projected the examples into a 7 dimensional sub-space $\mathbf{b}$ as described above. Figure 11 shows the scatter of b_2 vs b_1 for the set of shapes. Because there is a sudden change in cross section as we move in the z direction, the shapes form two distinct groups, with a low probability of finding an intermediate shape. Figure 12 shows the p.d.f. from fitting two gaussians to the adaptive kernel estimate of the distribution. Figure 10 shows the shapes obtained by corresponding combinations of b_1 and b_2. The mixture model of the p.d.f.suggests that the shapes in the middle of the figure are less likely than those on the left and right. The full model uses a two gaussian mixture to represent the distribution of b_1 and b_2, and assumes the remaining 5 parameters $b_3 \dots b_7$ are independent and normally distributed with variances λ_i.

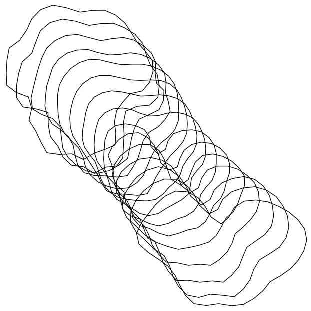

Figure 9: Contours from sequential slices

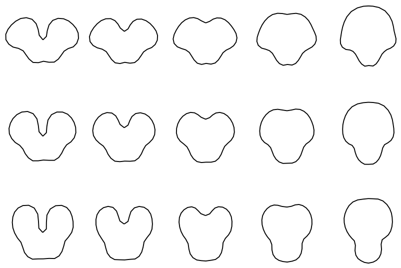

Figure 10: Shape for b_1 vs b_2 for brain stem

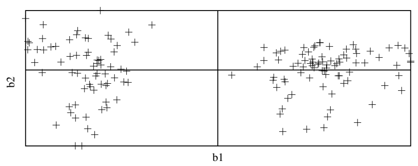

Figure 11: Plot of b_1 vs b_2 for brain stem

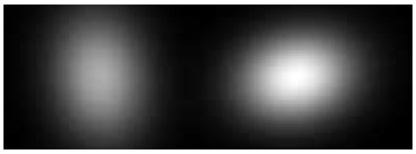

Figure 12: pdf approximation with 2 gaussians

Figure 13 demonstrates the Active Shape Model search for the brain stem in a new image. After 14 iterations it converges to a good solution.

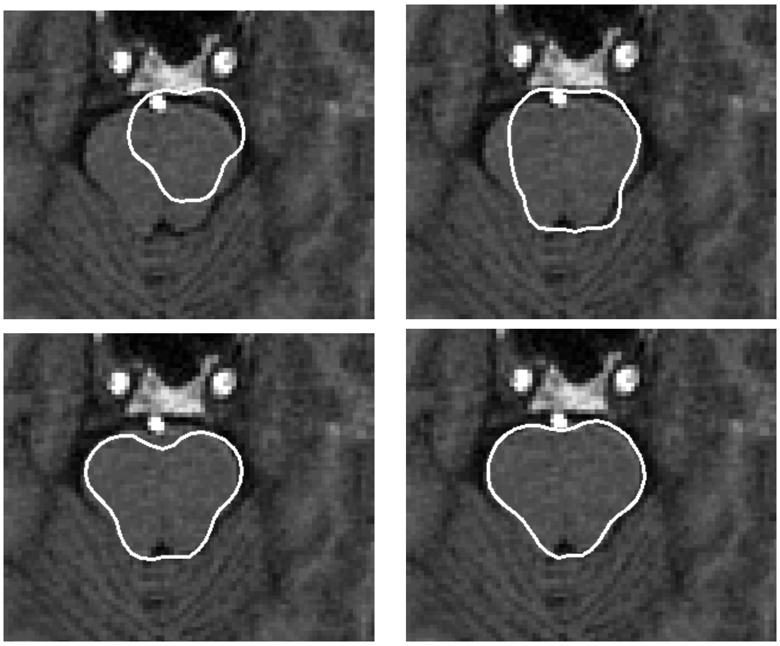

Figure 13: Searching for a brain stem. Initial position and after 4, 8 and 14 iterations.

6 Discussion and Conclusions

The approach described above is a generalisation of the PDM, and if a single gaussian is used in the mixture, is identical to it. Its main advantages are that it is able to model more complex variations, including those cases where there are two or more distinct classes of shape variation. The approach can be extended into 3D, though the alignment stage becomes more complex.

One problem with the method is that more examples are required to obtain reliable parameters for a mixture model than for a single gaussian. However, if we use a mixture model in only the first two or three dimensions of model parameter space (assuming the rest to be independent normal), we may not need too many more training examples. We anticipate that this requirement can in part be addressed by automatic labelling schemes [7], which, given a set of contours, choose the optimal positions of the landmarks required for the shape model.

Using a mixture model representation of the p.d.f.introduces more constraints on valid parameter combinations than assuming a single gaussian. During search this is only important when the data is sufficiently noisy or ambiguous that unconstrained search would lead to illegal parameter combinations. A common source of non-linearity is the rotation of sub-parts or of 3D objects viewed in 2D. Heap and Hogg [6] and Bowden *et. al.* [1] describe applications where the training examples form a highly non-linear space, which can be successfully represented by piece-wise linear sub-models. These are essentially the same as the approach presented above, but each sub-model (corresponding to one mixture component) is generated using clustering algorithms rather than by applying the EM, and the probabalistic nature of the problem is not properly addressed. Note, however, that an initial clustering of the data is one way of seeding the EM algorithm. The results presented in their papers demonstrate the robustness that can be obtained by using non-linear constraints.

The choice of the optimal number of mixture components to used is difficult. Automatic approaches are discussed in [8]. In the above we have chosen the number by inspection of the data. More work is required to determing the number automatically. If we assume that the kernel estimate gives the optimal estimate of the p.d.f. given the data, we can use this as a baseline to measure the accuracy of the mixture model p.d.f.The number of modes can then be chosen to pass a user defined error threshold.

We have demonstrated how mixture models can be used to represent the non-linear shape variations displayed by a class of objects and how such models can be used in the search for examples of the class in new images. This approach generates more specific models than earlier methods, leading to more robust search algorithms.

Appendix A: The Modified EM Algorithm

To fit a mixture of m gaussians to N samples $\mathbf{x}_i$, assuming a covariance of $\mathbf{T}_i$ at each sample, we iterate on the following 2 steps:

E-step Compute the contribution of the i^{th} sample to the j^{th} gaussian

$$p_{ij} = \frac{w_j N(\mathbf{x}_i : \mu_j, \mathbf{S}_j)}{\sum_{j=1}^{m} w_j N(\mathbf{x}_i : \mu_j, \mathbf{S}_j)} \tag{6}$$

M-step Compute the parameters of the gaussians,

$$w_j = \tfrac{1}{N}\textstyle\sum_i p_{ij} \quad , \quad \mu_j = \tfrac{1}{N w_j}\textstyle\sum_i p_{ij}\mathbf{x}_i \tag{7}$$

$$\mathbf{S}_j = \frac{1}{N w_j}\sum_i p_{ij}[(\mathbf{x}_i - \mu_j)(\mathbf{x}_i - \mu_j)^T + \mathbf{T}_i] \tag{8}$$

Strictly we ought to modify the E-step to take $\mathbf{T}_i$ into account as well, but in our experience just changing the M-step gives satisfactory results.

Acknowledgements

Tim Cootes is funded by an EPSRC Advanced Fellowship. The MR images were generated by C. Hutchinson and his team at Dept. Diagnostic Radiology, University of Manchester. They were annotated by Dr A.Hill.

References

[1] R. Bowden, T. Michell, and M. Sarhadi. Reconstructing 3d pose and motion from a single camera view. In P. Lewis and M. Nixon, editors, 9^{th} *British Machine Vison Conference*, volume 2, pages 904–013, Southampton, UK, Sept. 1998. BMVA Press.

[2] T. F. Cootes, C. J. Taylor, D. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, Jan. 1995.

[3] T. F. Cootes, C. J. Taylor, and A. Lanitis. Active shape models : Evaluation of a multi-resolution method for improving image search. In E. Hancock, editor, 5^{th} *British Machine Vison Conference*, pages 327–336, York, England, Sept. 1994. BMVA Press.

[4] C. Goodall. Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society B*, 53(2):285–339, 1991.

[5] T. Heap and D. Hogg. Automated pivot location for the cartesian-polar hybrid point distribution model. In R. Fisher and E. Trucco, editors, 7^{th} *British Machine Vison Conference*, pages 97–106, Edinburgh, UK, Sept. 1996. BMVA Press.

[6] T. Heap and D. Hogg. Improving specificity in pdms using a hierarchical approach. In A. F. Clark, editor, 8^{th} *British Machine Vison Conference*, pages 80–89, University of Essex, UK, Sept. 1997. BMVA Press.

[7] A. Hill and C. J. Taylor. A method of non-rigid correspondence for automatic landmark identification. In 7^{th} *British Machine Vison Conference*, pages 323–332. BMVA Press, Sept. 1996.

[8] G. McLachlan and K.E.Basford. *Mixture Models: Inference and Applications to Clustering*. Dekker, New York, 1988.

[9] B. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, London, 1986.

[10] P. Sozou, T. F. Cootes, C. J. Taylor, and E. DiMauro. Non-linear point distribution modelling using a multi-layer perceptron. In D. Pycock, editor, 6^{th} *British Machine Vison Conference*, pages 107–116, Birmingham, England, Sept. 1995. BMVA Press.

[11] P. Sozou, T. F. Cootes, C. J. Taylor, and E. D. Mauro. A non-linear generalisation of point distribution models using polynomial regression. *Image and Vision Computing*, 13(5):451–457, June 1995.