

Automatic Landmark Generation for Point Distribution Models

A. Hill and C. J. Taylor

*Department of Medical Biophysics,
University of Manchester,
Oxford Road,
Manchester M13 9PT.
email ah@wiau.mb.man.ac.uk*

Abstract

Point Distribution Models (PDMs) are statistically derived flexible templates which are trained on sets of examples of the object(s) to be modelled. They require that each example is represented by a set of points (*landmarks*) and that each landmark represents the same location on each of the examples. Generating the landmarks from 2D boundaries or 3D surfaces has previously been a manual process. Here, we describe a method for automatically generating PDMs from a training set of pixelated boundaries in 2D. The algorithm is a two-stage process in which a pair-wise corresponder is first used to establish an approximate set of landmarks on each of the example boundaries; in the second phase the landmarks are refined using an iterative non-linear optimisation scheme to generate a more compact PDM. We present results for two objects – the right hand and a chamber of the heart. The models generated using the automatically placed landmarks are shown to be better than those derived from landmarks located manually.

1 Introduction

Point Distribution Models (PDMs) are statistically derived flexible templates, introduced by Cootes et al [2], for modelling the appearance of objects with variable shape. They have proved to be useful in a wide variety of 2D image interpretation problems including surveillance [1], face recognition[7], monitoring farm animals [8], hand-written character recognition [7], and medical image analysis – locating, for example, heart chambers, vertebræ, and structures in the brain [3]. Recently, the PDM approach has been extended to 3D [5] and the relationship with other flexible template systems in 2D and 3D explored [4,9].

A PDM is generated by performing a statistical analysis on a training set of the object(s) to be modelled. In each example the structures of interest are represented by a set of labelled points. These *landmark* points must be placed at equivalent locations on each of the training examples. For instance, if we were to model faces, point 32 might always represent the centre of the right pupil. If the landmarks are not placed consistently, the resulting PDM must account for a noise component in their positions as well as true shape variability; it is consequently neither as compact nor as specific as it could be. Placing the landmark points is often one of the most time-consuming aspects of building a PDM and intro-

duces an undesirable element of subjectivity. It is particularly difficult to place the landmarks appropriately when building a 3D model from volume images, such as magnetic resonance images of the brain [5].

The aim of the work described in this paper is to automate the process of landmark placement. Training is normally a two-stage process: first, the example images are interactively segmented to identify the important 2D boundaries or 3D surfaces; secondly, the landmark points are placed on these boundaries/surfaces – some at uniquely identifiable locations, others equally spaced around the boundary/surface to define its locus. The first step is essential as it allows the user to define the interpretation required. The second step is required for purely technical reasons; it can be difficult and time-consuming and relies on the skill and understanding of the user to obtain satisfactory results.

Our objective is to automate this second step, reducing the effort involved in training a PDM and optimising the placement of landmarks so as to give a compact and specific model. The work we present here is our first attempt and is restricted to considering closed boundaries of 2D objects.

2 Overview of Method

In order to generate the optimal PDM for a given class of objects, from a set of example boundaries, we must select the pixel location of each landmark on the boundary of each example so as to make the PDM as compact as possible. However, there may be a large number of landmarks on each example, a large number of examples in the training set and a large number of pixels on the boundary of each example. The combinatorics seem to preclude a direct search for the best set of landmark locations.

In order to overcome this problem the method we have developed is a two-stage process. In the first stage an approximate set of landmarks is generated for each member of the training set. This is accomplished by establishing correspondences between pairs of examples in the training set using a dynamic programming algorithm which matches the curvature of the boundaries. By merging the matched pairs and applying the matching algorithm iteratively a mean shape is generated to which each member of the training set can be corresponded. Any set of landmarks positioned on the mean shape can then be projected onto each of the examples. In the second stage we refine the locations of the landmarks generated in the first stage. A non-linear optimiser adjusts the positions of the landmarks in an iterative scheme so as to produce a more compact model.

3 Approximate Landmark Generation

Let us assume that the object we wish to model can be represented as a closed boundary within an image. The training set consists of a set of images containing one or more examples of the object to be modelled. The objects can have various orientations, positions, scales and shapes within the images. In order to produce a set of landmarks on the boundary of each example, to train a PDM, we must first be able to identify corresponding points on different examples of the object.

Let us assume that a method of establishing correspondence exists which can take a pair of object boundaries and generate a pixel-to-pixel mapping. Further

assume that a metric exists which describes how well the two shapes correspond. We can use the pair-wise corresponder to generate the mean shape for the set of examples using the following algorithm :

- 1) Construct a matrix of correspondence values, one for each pair of shapes, $\Phi(i, j)$ ($1 \leq i, j \leq N$, $i \neq j$).
- 2) Find the member of the training set which is most difficult to match. This is achieved by finding the best partner for each example (i.e. the best $\Phi(i, j)$ for a given i or j) and identifying the worst Φ value associated with these best-partner pairs.
- 3) Remove the example which is most difficult to match, together with its best-match partner, from the training set and apply steps (1-3) repeatedly until all members are matched.
- 4) For each matched pair, construct a mean shape. The mean shape is constructed by averaging the position of each of the corresponding points on the two boundaries as indicated by the pair-wise corresponder.
- 5) Regard the mean shapes generated in step (4) as a new training set and apply steps (1-5) repeatedly until only one mean shape remains.

This algorithm can be represented as a binary tree as shown in figure 1. The leaves of the tree are the individual members of the training set and the mean shape is the root. A set of landmarks can now be generated automatically on the mean shape. Any sensible algorithm can be used for this purpose e.g. a set of landmarks positioned according to equal path-length along the boundary.

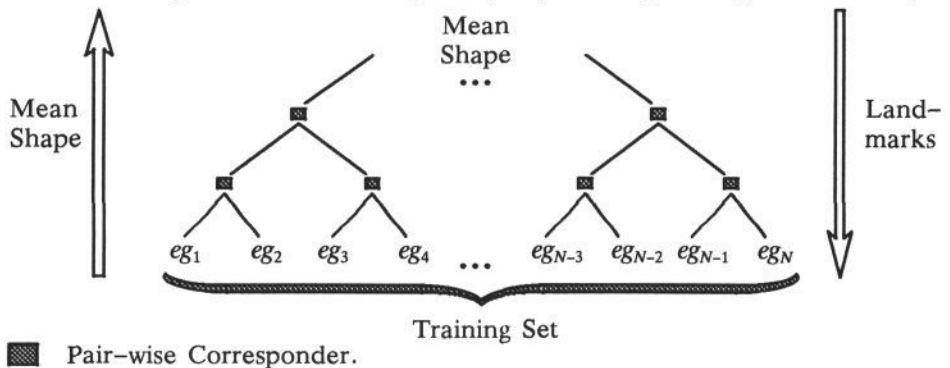


Figure 1 : Generating the Mean Shape and Approximate Landmarks.

Once the landmarks have been placed on the mean shape they can be projected back along the branches of the tree towards the leaves. This can be accomplished using the pair-wise corresponder. Each parent node in the binary tree has two children. The pair-wise corresponder can be used to match the parent to each of its two children. Using the pixel-matches generated by the corresponder, the landmarks of the parent can be mapped onto the children. This procedure is applied from the root to the leaves to generate the landmarks for each member of the training set.

Clearly, this algorithm requires a reliable method of pair-wise correspondence. We have employed a method which uses the curvature of the boundary of the

objects as a basis for a dynamic programming matching algorithm[6]. The dynamic programming algorithm attempts to match points of high curvature on the two boundaries so as to minimise the discrepancy in the curvature over the entire length of the boundary. We are also investigating the method of Shapiro & Brady [12,13] and the method of Sclaroff & Pentland [10,11]. In the Shapiro & Brady scheme a modal analysis of the chord-lengths $r_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$

($1 \leq i, j \leq n$) is performed for each of two shapes and the resulting modes are used to find the correspondences between the two sets of points. In the Sclaroff & Pentland method a finite element model (FEM) is constructed from the boundaries of each shape. A modal analysis of the FEM generates vibrational modes of variation which are used to match the two shapes.

4 Generating an Initial PDM

Given a set of approximate landmarks $\{\mathbf{x}_i = (x_{i,1}, y_{i,1}, \dots, x_{i,n}, y_{i,n})^T, (1 \leq i \leq N)\}$ we can construct a PDM as follows (see [2] for details) :

- Calculate the mean shape $\mathbf{x}_m = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i$.
- Calculate the deviations from the mean $\delta \mathbf{x}_i = \mathbf{x}_i - \mathbf{x}_m$ ($1 \leq i \leq N$).
- Calculate the covariance matrix $\mathbf{S} = \frac{1}{N} \sum_{i=1}^N \delta \mathbf{x}_i \delta \mathbf{x}_i^T$.
- The modes of variation are given by \mathbf{p}_k ($k = 1..2n$), the unit eigenvectors of \mathbf{S} such that $\mathbf{S} \mathbf{p}_k = \lambda_k \mathbf{p}_k$ where λ_k is the k^{th} eigenvalue.
- Select the t ($< 2n$) major modes of variation which capture the required variability. In the examples we present below we select the t modes which model 90% of the variance in the training set.

This procedure results in a model with a set of parameters $\mathbf{b} = (b_1, b_2, \dots, b_t)^T$ which act as weights for the t major eigenvectors of the co-variance matrix. These weights can be manipulated to create new instances from the class of objects modelled :

$$\mathbf{x} = \mathbf{x}_m + \mathbf{P} \mathbf{b} \quad \text{where } \mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_t] \quad (1)$$

5 Refining the Landmark Positions

The set of landmarks generated by the algorithm described in section 3 will not, in general, be the best set of landmarks from which to build a PDM. This is because the criterion used by the corresponder to match two shapes does not relate directly to the compactness of the model. We are interested in generating the most compact representation possible for the PDM. We can achieve this by attempting to reduce the total absolute variance of the model whilst trying to explain as much of the variance using as few modes of variation as possible.

In order to generate a more compact PDM we consider each member of the training set in turn and attempt to find a new set of landmarks on its boundary

which will lead to a more compact model. This is achieved by varying the pose (orientation and scale – the origin is fixed at the centroid of the object) and shape (\mathbf{b} vector) of the landmarks associated with the given member of the training set in order to minimise the following expression :

$$f(\theta, s, \mathbf{b}) = D^2 + \alpha(E^2 - E_0^2) \quad (2)$$

where :

$$\left. \begin{aligned} D^2 &= \sum_{k=1}^t \frac{b_k^2}{\lambda_k}, & \alpha &= \frac{2}{\lambda_t} \\ E^2 &= \sum_{i=1}^n [(x_i - x_i')^2 + (y_i - y_i')^2] \\ \mathbf{x} &= (x_1, y_1, x_2, y_2, \dots, x_n, y_n)^T = \mathbf{M}(\mathbf{x}_m + \mathbf{Pb}) \\ E_0^2 &= E^2 \text{ measured at } \mathbf{x} = \mathbf{x}_0 \end{aligned} \right\} \quad (3)$$

\mathbf{M} is the diagonal pose transformation matrix given by :

$$\mathbf{M} = \begin{bmatrix} \mathbf{Q} & \mathbf{0} & \cdot \\ \mathbf{0} & \mathbf{Q} & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} s \cos(\theta) & -s \sin(\theta) \\ s \sin(\theta) & s \cos(\theta) \end{bmatrix} \quad (4)$$

The term $\mathbf{x}' = \{x_1', y_1', \dots\}$ in equation (3) represents the set of points on the boundary of the example which are closest to the set of points \mathbf{x} generated from the current set of (θ, s, \mathbf{b}) values (see below). The term \mathbf{x}_0 is the set of landmarks in the training set representing the current example object. We now consider each of the terms in equation (2) in more detail.

The term D^2 in (2) is the squared Mahalanobis distance which indicates the distance of the vector \mathbf{b} from the origin in the parameter space of the model. The origin of the parameter space ($\mathbf{b} = 0$) is the mean set of landmark positions. The Mahalanobis distance can be reduced in two ways :

- by decreasing the magnitude of the shape parameters, b_i , associated with the member of the training set they represent i.e. finding a new vector, \mathbf{b}_{new} , which represents a set of landmarks more similar to the mean set of landmarks but which can still be represented by the boundary of the member of the training set under consideration; by attempting to re-position the landmarks around the boundary so as to be closer to the mean set of landmarks, the total variance, λ_T , is reduced;
- by reducing the b_j values associated with the higher values of j , the value of D^2 is reduced; the variances, λ_j , (the eigenvalues associated with the covariance matrix – see section 4) decrease rapidly with increasing j ; the model becomes more compact in the sense that the first few modes of variation explain a greater proportion of the total variance.

Clearly, attempting to minimise the Mahalanobis distance alone would be unsuccessful; the set of landmarks for each example would collapse onto the mean set of landmarks ($\mathbf{b} = 0$). We prevent this happening by introducing a *representen-*

tation error (E^2) associated with each shape. The representation error measures the discrepancy between a desired set of landmarks (generated from a particular set of (θ, s, \mathbf{b}) values) and the ability to represent the suggested landmarks on a given boundary. The representation error is measured by locating the point on the boundary closest to each of the landmarks and summing the inter-point distances (see figure 2).

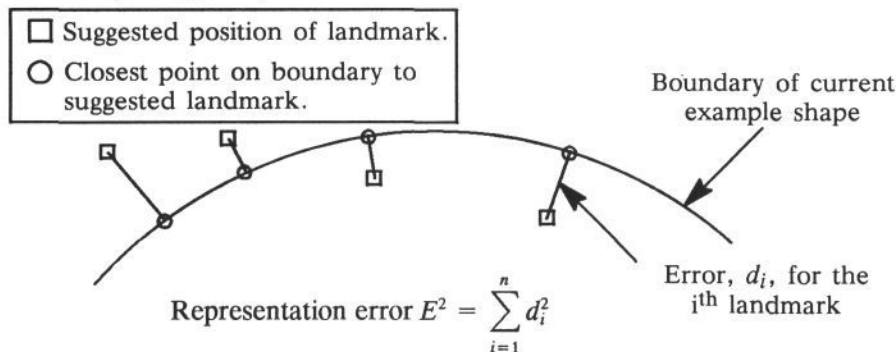


Figure 2: Calculating the Representation Error for a given set of Landmarks and a given Boundary.

The set of landmarks representing each member of the training set has an implicit representation error due to model truncation. In equation (2) this is the term E_0^2 . For a given set of (θ, s, \mathbf{b}) values and a given boundary the representation error is E^2 (see figure 2). The term $(E^2 - E_0^2)$ in equation (2) represents the increase in the representation error associated with a change in the pose and shape of the set of landmarks for a particular member of the training set. This is a squared distance term which is normalised using the value $2/\lambda_r$ in order to compare it with the dimensionless Mahalanobis distance. What equation (2) says, therefore, is that we will only accept a decrease in the Mahalanobis distance if the improvement is not outweighed by an increase in the representation error. In order to minimise equation (2) we have employed the Simplex method of non-linear optimisation (NAG[†] E04CCF).

The refinement stage can be summarised as follows :

- i) Generate the PDM from the current training set;
- ii) for each member of the training set re-position the landmarks by minimisation of equation (2); place the new set of landmarks for each member of the training set into a new training set;
- iii) if no improvement has been found, stop; otherwise, replace the training set with the new training set and return to step (i).

6 Results

We have used the method above to generate PDMs for two classes of object : hands and heart chambers (left ventricles). In each case the objects of interest were segmented from a set of images containing examples (18 for the hand and [†]NAG is a registered trademark.

66 for the heart). The segmented pixel boundaries of the objects formed the training set. For both the hand and the heart we generated three models :

- a model generated by manually placing a single landmark on the boundary of each example; extra landmarks were equally-spaced around the boundary.
- a model generated by manually placing a small set of landmarks on the boundary of each example; extra landmarks were generated at equally-spaced intervals along the boundary between the landmarks selected by hand; this is the technique we have used previously to generate PDMs; for example, on the hand 11 landmarks were placed, one at the tip of each finger and the thumb, two at the wrist and one at each of the junctions between the fingers.
- a model generated automatically as described above.

These models are shown in figures 3 and 4. From these results it is clear that the simple approach of placing a single point on the boundary at a single key landmark position and generating further landmarks by equally spacing points

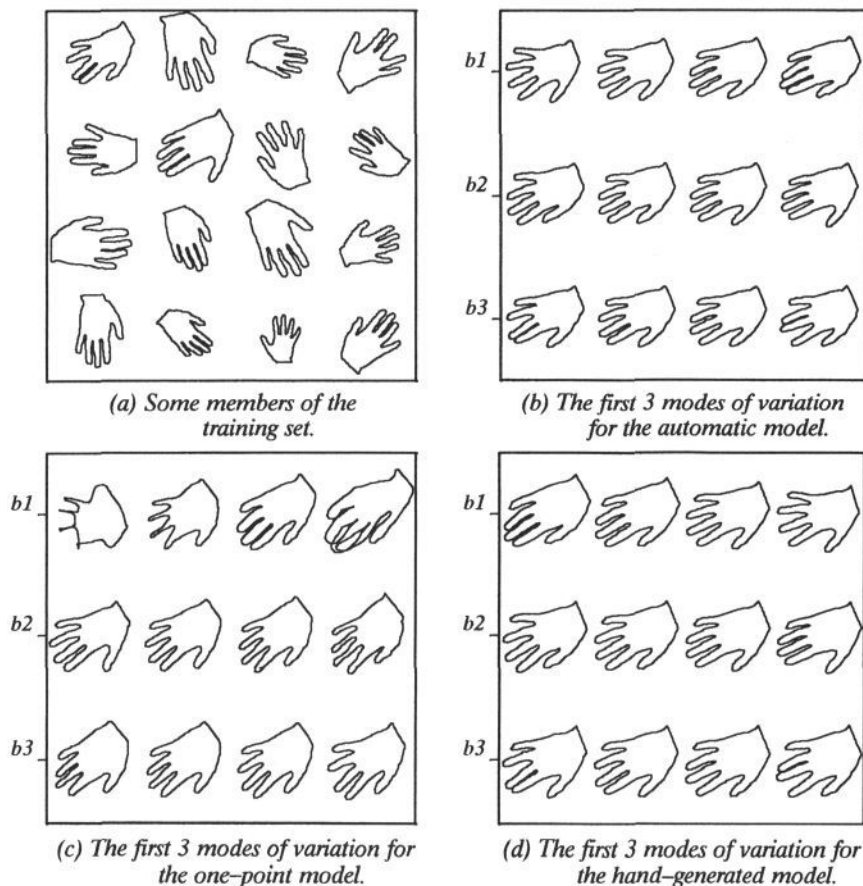
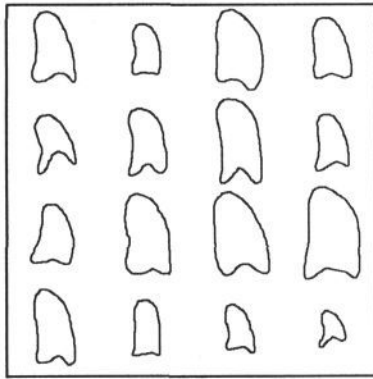
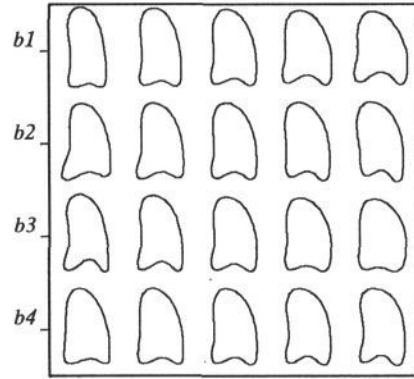


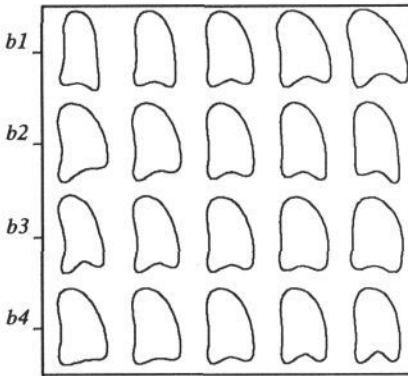
Figure 3: Training Set and PDMs for the Hand.



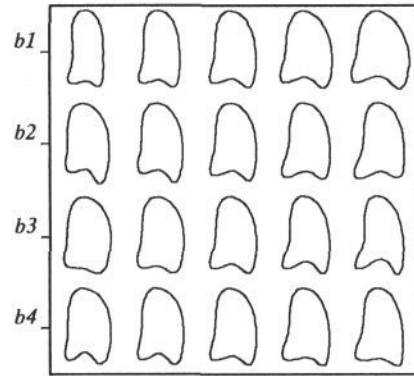
(a) Some members of the training set.



(b) The first 3 modes of variation for the automatic model.



(c) The first 3 modes of variation for the one-point model.



(d) The first 3 modes of variation for the hand-generated model.

Figure 4: Training Set and PDMs for the Heart.

around the boundary fails to generate a PDM with the required specificity. This is especially true of the hand model (see figure 3c). As the hand changes shape the relative lengths of the sub-parts change e.g. if the fingers are curled slightly, they appear shorter than when fully extended. This means that points change their position on the boundary of the hand according to how extended the fingers are rather than consistently representing the same features. Consequently, several of the modes associated with this model are concerned with shuffling points around the boundary in order to compensate for other modes which do not capture the variation correctly (the 2nd mode is a compensation for the 1st mode in figure 3c).

It can also be seen from figures 3 and 4 that the models generated automatically are qualitatively very similar to those generated by hand – see figures 3 b&d and figures 4 b&d. (Note that the modes of variation are displayed left-to-right but that they may appear reversed right-to-left for different models. This is because the signs of the eigenvectors, \mathbf{p}_k , representing the modes of variation are arbitrary.)

Recall that the automated method of landmark placement described in section 2 has two phases. In the first phase the landmarks are positioned approximately and in the second phase the landmark positions are refined. The effect of the refinement process is summarised in table 1. This table shows the cumulative percentage of the total variance, λ_T , explained by each mode, together with the total variance. These values are shown for both the training set containing the approximate landmarks and the training set of landmarks generated after 5 iterations of refinement. The refinement algorithm not only reduces the total variance over the training set but also increases the compactness of the models in the sense that the first few modes of variation describe a greater proportion of the total variance. We have also included the values for the PDMs generated manually. Comparing the refined automatic models with the manual models, we see that the automatic models have less total variance and are also more compact.

Mode	Cumulative % Variance Explained					
	Hand Model			Heart Model		
	Approx	Refined	Manual	Approx	Refined	Manual
1	38.4	47.6	40.5	33.1	48.4	41.2
2	61.5	74.0	66.3	58.7	68.7	62.5
3	75.2	88.1	80.2	73.5	82.3	72.8
4	83.0	94.6	87.5	81.4	88.9	81.8
5	88.3	96.1	92.0	87.0	92.8	86.9
6	91.5	97.3	94.2	89.6	94.2	90.2
λ_T	0.95	0.67	0.89	0.70	0.39	0.40

Table 1: Refinement of Hand and Heart Models.

7 Conclusions

We have described an algorithm which can be used to generate PDMs automatically from a training set of example shapes, where each shape is represented by a simple closed boundary in 2D. We have demonstrated the successful application of the method by both a qualitative and quantitative comparison of two PDMs generated by the automatic method with PDMs generated manually. Having shown that it is possible to generate models automatically for closed boundaries in 2D we intend to extend the method in two ways. First, the method must be able to cope with objects consisting of more than a single part. Second, the method must be extensible to surfaces in 3D.

In the first phase of the two-stage algorithm described in section 2 we employed a pair-wise correspondence algorithm to match the boundaries of two example objects. In order for this matching process to be successful when considering surfaces in 3D it would be highly advantageous if no connectivity of the 3D points were required by the correspondence algorithm. It is unlikely that the dynamic programming algorithm we have employed here for the 2D case can be extended

to 3D as the connectivity of the 2D boundary is exploited by the algorithm. However, both the Shapiro & Brady [12,13] and Sclaroff & Pentland [10,11] methods extend straightforwardly to 3D as no connectivity information is required. The reliability of these correspondence algorithms has yet to be investigated and the computational requirements of the algorithms may prove to be considerable.

8 Acknowledgements

This research was funded by the UK Science & Engineering Research Council.

9 References

- [1] **Baumberg A., Hogg D.** *Learning Flexible Models from Image Sequences*, Proc. 3rd European Conference on Computer Vision, Stockholm, Sweden, 1994, 299–308.
- [2] **Cootes T. F., Cooper D. H., Taylor C. J., Graham J.** *Training Models of Shape from Sets of Examples*, Proc. 3rd British Machine Vision Conference, Leeds, September 1992, Springer Verlag; 9–18.
- [3] **Cootes T. F., Hill A., Taylor C. J., Haslam J.** *The Use of Active Shape Models for Locating Structures in Medical Images*, Information Processing in Medical Imaging, Arizona, June 1993; 33–47.
- [4] **Cootes T. F.** *Combining Point Distribution Models with Shape Models based on Finite Elements Analysis*, these proceedings.
- [5] **Hill A., Thornham A., Taylor C. J.** *Model-Based Interpretation of 3D Medical Images*, Proc. 4th British Machine Vision Conference, Guildford, September 1993; 339–348.
- [6] **Cosquer F. J. N.** *Shape Matching using Dynamic Programming*, M.Sc. Thesis, Manchester University, 1989.
- [7] **Lanitis A.** *A Generic System for Classifying Variable Objects using Flexible Template Matching*, Proc. 4th British Machine Vision Conference, Guildford, September 1993; 329–338.
- [8] **Marchant J. A.** *Adding Grey Level Information to Point Distribution Models using Finite Elements*, Proc. 4th British Machine Vision Conference, Guildford, September 1993; 309–318.
- [9] **Martin J., Pentland A., Kikinis R.** *Shape Analysis of Brain Structures using Physical and Experimental Modes*, Technical Report No. 276, MIT AI Lab, 1994.
- [10] **Scalroff S., Pentland A.** *A Modal Framework for Correspondence and Description*, Proc. 4th International Conference of Computer Vision, Berlin, Germany, May 1993; 308–313.
- [11] **Scalroff S., Pentland A.** *Modal Matching for Correspondence and Recognition*, Technical Report No. 201, MIT AI Lab, 1993.
- [12] **Shapiro I. S., Brady M. J.** *A Modal Approach to Feature-Based Correspondence*, Proc. 2nd British Machine Vision Conference, Glasgow, September 1991, Springer Verlag; 78–85.
- [13] **Shapiro I. S.** *Towards a Vision-Based Motion Framework*, First Year Report, Department of Engineering Science, Oxford University. May 1991.