

Training Models of Shape from Sets of Examples

T.F.Cootes, C.J.Taylor, D.H.Cooper and J.Graham

Department of Medical Biophysics
University of Manchester
Oxford Road
Manchester M13 9PT
email: bim@wiau.mb.man.ac.uk

Abstract

A method for building flexible shape models is presented in which a shape is represented by a set of labelled points. The technique determines the statistics of the points over a collection of example shapes. The mean positions of the points give an average shape and a number of modes of variation are determined describing the main ways in which the example shapes tend to deform from the average. In this way allowed variation in shape can be included in the model. The method produces a compact flexible 'Point Distribution Model' with a small number of linearly independent parameters, which can be used during image search. We demonstrate the application of the Point Distribution Model in describing two classes of shapes.

1 Introduction

We have previously described a method for modelling two dimensional shape, based on the statistics of chord lengths over a set of examples [12]. Although this provided a means of automatically parameterising shape variability, the method was difficult to use, requiring an iterative procedure to reconstruct a shape given a set of parameters. The method has computational complexity $O[n^2]$ where n is the number of points used to describe the shape. In this paper we present a new method which produces a more compact representation, allows direct reconstruction of a shape from a set of parameters and offers $O[n]$ computational complexity.

Image interpretation using rigid models is well established [1,2]. However, in many practical situations objects of the same class are not identical and rigid models are inappropriate. This is particularly true in medical applications, but also many industrial applications involve assemblies with moving parts, or components whose appearance can vary. In such cases flexible models, or deformable templates, can be used to allow for some degree of variability in the shape of the imaged object.

Yuille, Cohen and Hallinan [3] and Lipson *et al* [4] use deformable templates for image interpretation. Unfortunately their templates are hand-crafted with modes of variation which have to be individually tailored for each application. Kass, Witkin and Terzopoulos [5] described 'Active Contour Models', flexible snakes which can stretch and deform to image features. These have been extended to apply constraints to their deformation by adjusting the elasticity and stiffness of the model [6,7]. Pentland and Sclaroff [8] model objects as lumps of elastic clay, generating different shapes using combinations of the modes of vibration of the clay. However this does not always lead to a very compact description of the variability within a particu-

lar class of objects. Bookstein [9] has studied the statistics of shape deformation by representing objects as sets of 'landmark points', but has not applied this to the problem of shape modelling. Mardia, Kent and Walder [10] represent the boundary of a shape as a sequence of points with distributions related by a covariance matrix. To fit a model to an image they cycle through the points to find the most likely position given the image and the current shape. The examples given seem to be local models, in that deforming one part of the boundary does not affect the rest of it until the change has been propagated round the boundary by the updating method.

In this paper we describe a new method of shape modelling based on the statistics of labelled points placed on a set of training examples. The sets of points are automatically aligned so that their mean positions and main modes of variation can be calculated. Aligning the shapes allows the positions of equivalent points in different examples to be compared simply by examining their co-ordinates. A model consists of the mean positions of the points and a number of vectors describing the modes of variation.

2 Point Distribution Models

Suppose we wish to derive a model to represent the shape of resistors as they appear on a printed circuit board, such as those shown in Figure 1. Different examples of resistor have sufficiently different shapes that a rigid model would not be appropriate. Figure 2 shows some examples of resistor boundaries which were obtained from backlit images of individual resistors. Our aim is to build a model which describes both typical shape and allowed variability, using the examples in Figure 2 as a training set.

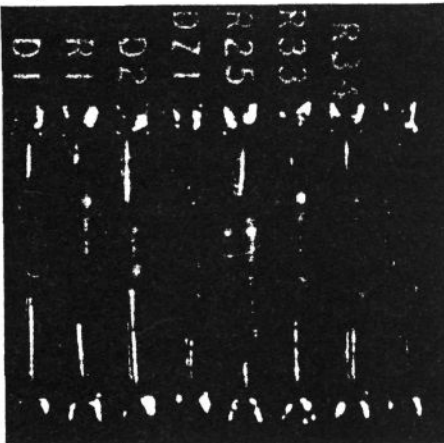


Figure 1 : Image of printed circuit board showing examples of resistors.

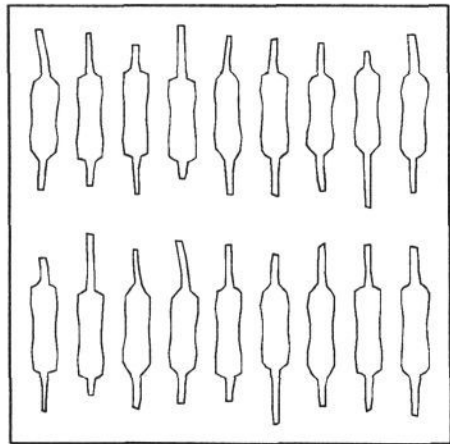


Figure 2 : Examples of resistor shapes from a training set.

2.1 Labelling The Training Set

In order to model a shape, we represent it by a set of points. For the resistors we have chosen to place points around the boundary, as shown in Figure 3. This must be done for each shape in the training set. The labelling of the points is important, each la-

belled point represents a particular part of the object or its boundary. For instance, in the resistor model, points 0 and 31 always represent the ends of a wire, points 3, 4 and 5 represent one end of the body of the resistor and so on. The method works by modelling how different labelled points tend to move together as the shape varies. If the labelling is incorrect, with a particular point placed at different sites on each training shape, the method will fail to capture shape variability.

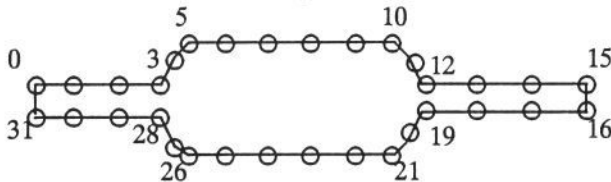


Figure 3 : 32 point model of the boundary of a resistor.

It is important that the points are placed correctly on each example image. This will usually require someone familiar with the application to choose the most appropriate set of points and to be able to reproducibly place them on different examples. This procedure can be time consuming, though we are developing tools to speed up the process. It should be noted that though the labelling of the training set is done manually, finding the mean shape and main modes of variation is automatic. Deducing such a set of modes would be very difficult by hand, particularly for more complex biological shapes.

2.2 Aligning The Training Set

Our modelling method works by examining the statistics of the co-ordinates of the labelled points over the training set. In order to be able to compare equivalent points from different shapes, they must be aligned in the same way with respect to a set of axes. If they are not, we would not be comparing like with like and any statistics derived would be meaningless. We achieve the required alignment by scaling, rotating and translating the training shapes so that they correspond as closely as possible. We aim to minimise a weighted sum of squares of distances between equivalent points on different shapes. This is a form of Generalised Procrustes Analysis [11].

We will first consider aligning a pair of shapes. Let \mathbf{x}_i be a vector describing the n points of the i^{th} shape in the set;

$$\mathbf{x}_i = (x_{i0}, y_{i0}, x_{i1}, y_{i1}, \dots, x_{ik}, y_{ik}, \dots, x_{in-1}, y_{in-1})^T$$

Let $M_j[\mathbf{x}_j]$ be a rotation by θ_j and a scaling by s_j . Given two similar shapes, \mathbf{x}_i and \mathbf{x}_j we can choose θ_j , s_j and a translation $(t_x, t_y)_j$ mapping \mathbf{x}_i onto $M_j[\mathbf{x}_j]$ so as to minimise the weighted sum

$$E_j = (\mathbf{x}_i - M_j(\mathbf{x}_j))^T \mathbf{W} (\mathbf{x}_i - M_j(\mathbf{x}_j)) \quad (1)$$

$$\text{where } M_j \begin{pmatrix} x_{jk} \\ y_{jk} \end{pmatrix} = \begin{pmatrix} (s_j \cos \theta) x_{jk} - (s_j \sin \theta) y_{jk} + t_{jx} \\ (s_j \sin \theta) x_{jk} + (s_j \cos \theta) y_{jk} + t_{jy} \end{pmatrix} \quad (2)$$

and \mathbf{W} is a diagonal matrix of weights for each point.

Details are given in Appendix A.

The weights can be chosen to give more significance to those points which tend to be most 'stable' over the set – the ones which move about least with respect to the other points in a shape. We have used a weight matrix defined as follows: let R_{kl} be the distance between points k and l in a shape; let $V_{R_{kl}}$ be the variance in this distance over the set of shapes; we can choose a weight, w_k , for the k^{th} point using

$$w_k = \left(\sum_{l=0}^{n-1} V_{R_{kl}} \right)^{-1} \quad (3)$$

If a point tends to move around a lot with respect to the other points in the shape, the sum of variances will be large, and a low weight will be given. If, however, a point tends to remain fixed with respect to the others, the sum of variances will be small, a large weight will be given and matching such points in different shapes will be a priority.

In order to align all the shapes in a set we use the following algorithm.

- 1) Rotate, scale and translate each of the shapes in the set to align to the first shape.

Repeat

- 2) Calculate the mean of the transformed shapes
- 3) Either
 - a) Adjust the mean to a default scale, orientation and origin,
 - b) Rotate, scale and translate the mean to align to the first shape
- 4) Rotate, scale and translate each of the shapes again to match to the adjusted mean.

Until convergence.

Stage 3 inside the iteration loop is required to renormalise the mean. Without this the algorithm is ill-conditioned – there are in effect $4(N_s-1)$ constraints on $4N_s$ variables (θ , s , t_x , t_y for each shape) – and will not converge – the mean will shrink, rotate or slide off to infinity. Constraints on the pose and scale of the mean allow the equations to have a unique solution. Either the mean is scaled, rotated and translated so it matches the first shape, or an arbitrary default setting can be used, such as choosing an origin at its centre of gravity, an orientation so that a particular part of the shape is at the top and a scale so that the distance between two points is one unit.

The convergence condition can be tested by examining the average difference between the transformations required to align each shape to the recalculated mean and the identity transformation. Experiments suggest that the method converges to the same result independent of which shape is aligned to in the first stage, though a formal proof of convergence has yet to be devised.

2.3 Capturing the Statistics of a Set of Aligned Shapes

Once a set of aligned shapes is available the mean shape and variability can be found. The mean shape, \bar{x} , is calculated using

$$\bar{x} = \frac{1}{N_s} \sum_{i=1}^{N_s} x_i \quad (4)$$

The modes of variation, the ways in which the points of the shape tend to move together, can be found by applying principal component analysis to the deviations from the mean as follows.

For each shape in the training set we calculate its deviation from the mean, dx_i , where

$$dx_i = x_i - \bar{x} \quad (5)$$

We can then calculate the $2n \times 2n$ covariance matrix, S , using

$$S = \frac{1}{N_s} \sum_{i=1}^{N_s} dx_i dx_i^T \quad (6)$$

The modes of variation of the points of the shape are described by the unit eigenvectors of S , p_i ($i = 1$ to $2n$) such that

$$S p_i = \lambda_i p_i \quad (7)$$

(where λ_i is the i 'th eigenvalue of S , $\lambda_i \geq \lambda_{i+1}$)

$$p_i^T p_i = 1 \quad (8)$$

It can be shown that the eigenvectors of the covariance matrix corresponding to the largest eigenvalues describe the most significant modes of variation in the variables used to derive the covariance matrix, and that the proportion of the total variance explained by each eigenvector is equal to the corresponding eigenvalue [13]. Most of the variation can usually be explained by a small number, t , modes. One method for calculating t would be to chose the smallest number of modes such that the sum of variance explained was a sufficiently large proportion of λ_T , the total variance of all the variables, where

$$\lambda_T = \sum_{i=1}^{2n} \lambda_i \quad (9)$$

The i 'th eigenvector affects point k in the model by moving it along a vector parallel to (dx_{ik}, dy_{ik}) , which is obtained from the k 'th pair of elements in p_i ,

$$(dx_{i0}, dy_{i0}, \dots, \underbrace{dx_{ik}, dy_{ik}}_{}, \dots, dx_{in-1}, dy_{in-1}) \quad (10)$$

Any shape in the training set can be approximated using the mean shape and a weighted sum of these deviations obtained from the first t modes

$$x = \bar{x} + P b \quad (11)$$

where $P = (p_1 \ p_2 \ \dots \ p_t)$ is the matrix of the first t eigenvectors,

$\mathbf{b} = (b_1 \ b_2 \ \dots \ b_i)^T$ is a vector of weights for each eigenvector

the eigenvectors are orthogonal, $\mathbf{P}^T\mathbf{P} = \mathbf{I}$ so

$$\mathbf{b} = \mathbf{P}^T(\mathbf{x} - \bar{\mathbf{x}}) \quad (12)$$

The above equations allow us to generate new examples of the shapes by varying the parameters (b_i) within suitable limits. The parameters are linearly independent, though there may be non-linear dependencies still present. The limits for b_i are derived by examining the distributions of the parameter values required to generate the training set. Since the variance of b_i over the training set can be shown to be λ_i , suitable limits are likely to be of the order of

$$-3\sqrt{\lambda_i} \leq b_i \leq 3\sqrt{\lambda_i} \quad (13)$$

since most of the population lies within three standard deviations of the mean.

3 Practical Examples

The techniques described above have been used to generate shape models for both manufactured and biological objects. We present results for the set of resistor shapes shown in Figure 2 and a set of hand shapes.

3.1 Resistor Example

The resistor shapes were aligned using the method described above, arranging the mean shape to be horizontal and scaling so the average distance of each point of the mean from its centre of gravity is one unit. The most significant eigenvalues of the covariance matrix derived are shown in Table 1.

Table 1 : Eigenvalues of the covariance matrix derived from a set of resistor shapes.

Eigenvalue	λ_i	$\frac{\lambda_i}{\lambda_r} \times 100\%$	$\sqrt{\lambda_i}$
λ_1	0.207	66%	0.46
λ_2	0.026	8%	0.16
λ_3	0.017	5%	0.13
λ_4	0.013	4%	0.11
λ_5	0.010	3%	0.10
λ_6	0.008	3%	0.09

Figure 4 shows the plot of b_1 against b_2 for the training set. The lack of structure in the scatter plot suggests that the parameters can be treated as independent. We are currently working on deriving more formal tests of independence. Any dependencies between the parameters would imply non-linear relationships between the original point positions and would result in some combinations of parameters generating 'illegal' shapes. By varying the first three parameters separately we can generate examples of the shape as shown in Figures 5–7. Each parameter 'represents' a mode of variation of the shape which can frequently be associated with an intuitive

description of the deformation. Compare Figures 5–7 with Figure 2. Varying the first parameter (b_1) adjusts the position of the body of the resistor up and down the wire. The second parameter varies the shape of the ends of the main body of the resistor, between tapered and square. The third parameter affects the curvature of the wires at either end. Subsequent parameters have smaller effects, including the wires bending in opposite directions. These modes of variation effectively capture the variability which was present in the training set.

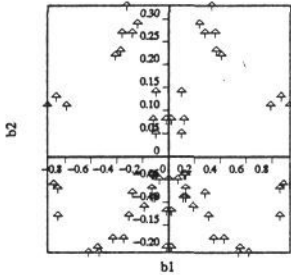


Figure 4 : Plot of b_1 vs b_2 for a training set of resistor shapes.

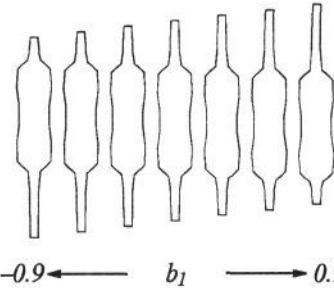


Figure 5 : Effects of varying the first parameter of the resistor model.

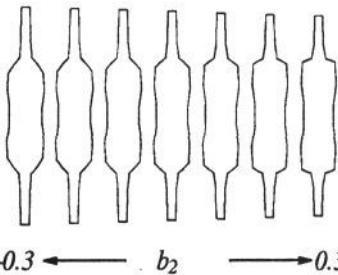


Figure 6 : Effects of varying the second parameter of the resistor model.

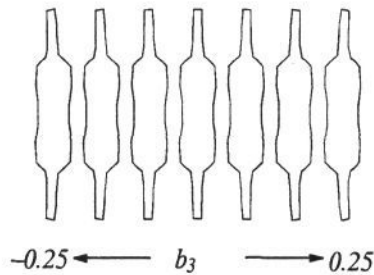


Figure 7 : Effects of varying the third parameter of the resistor model.

3.2 Hand Example

A set of 18 hand shapes was generated from images of the right hand of one of the authors (Figure 8). Each was represented by 72 points around the boundary. These were planted on the examples by locating 12 control points at the ends and joints of the fingers and filling in the rest equally along the connecting boundaries. A model was trained on the data, and it was found that 96% of the variance could be explained by the first 6 modes of variation. The first three modes are shown in Figure 9, and consist of combinations of movements of the fingers. Again, a compact parameterised model has been generated.

4 Discussion and Conclusions

The method outlined above allows a compact, flexible shape model to be built, representing a class of shapes by the mean positions of a set of labelled points and a small number of modes of variation about the mean. The model points do not have to lie only on the boundary of objects, they can represent internal features, and even sub-components of a complex assembly. In the latter case the model describes both

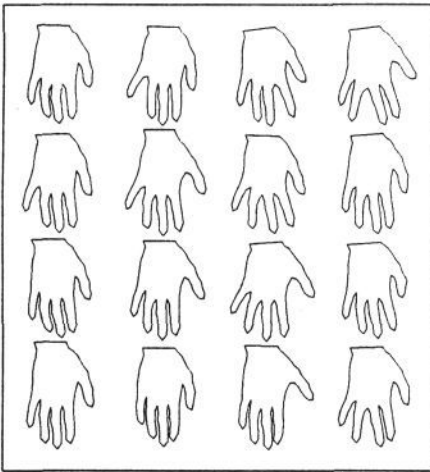


Figure 8 : Training set of hand shapes, each defined by 72 points.

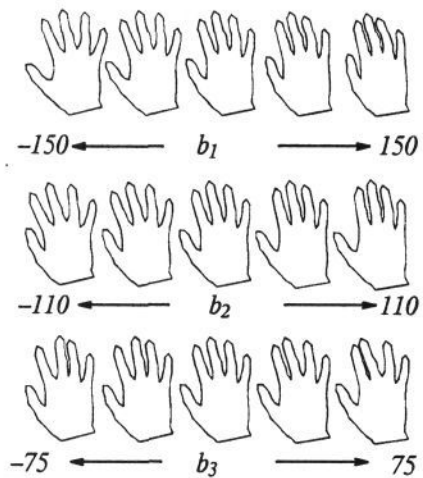


Figure 9 : Effects of varying each of the first three parameters of the hand model individually.

the variations in the shapes of the sub-components and the geometric relationships between components. Such a model, representing a section through the ventricles in the brain in MR scans is described by Hill *et al* in [16].

It is important to arrange that all the examples used to train the model are similarly aligned with respect to a set of axes, to ensure that the labelled points in different shapes are being compared correctly. In some cases an obvious alignment is apparent, but in others, particularly medical cases where the shapes of organs are very flexible, the automatic least squares alignment method is essential. The method has been used successfully to model a variety of objects from both industrial and biological domains.

The models we build are linear. Varying each parameter individually moves the points along straight lines. The method is inefficient at modelling non-linear effects such as bending or rotation of one sub-component about another. To deal with such cases a non-linear model of the modes of variation would be required. We have begun experimenting with a system which represents each mode using a polynomial curve rather than a straight line [14]. Some promising results have been produced which will be the subject of a further paper.

Point Distribution Models have been used in image search. A local optimiser called the Active Shape Model has been developed [15] which provides a way of iteratively improving an initial estimate of the position, pose and shape parameters of a model fitted to image data. The model has also been used in conjunction with a generate and test strategy based around Genetic Algorithms [16]. The hand and resistor models described above have been successfully used to find examples in images with both techniques.

The model can also be used in a classifier. Given an example of a shape, an estimate can be made of how likely that example is to be a member of the class of shapes described by a model. If labelled points are placed on the example and the point set aligned with the mean shape, Equation 12 can be used to calculate the model para-

mers required to generate the example. The distributions of the parameters can be estimated from the training set, allowing probabilities to be assigned. This technique has been successfully used in a simple handwritten character recognition application [17].

The models are compact and easy to use. Given a set of parameters an example of the model can be calculated rapidly. The models are well suited to generate-and-test image search strategies in many domains.

Acknowledgements

This work is funded by SERC under the IEATP Initiative (Project Number 3/2114). The authors would like to thank the other members of the Wolfson Image Analysis Unit for their help and advice, particularly D.Bailes and A.Hill.

Appendix : Aligning A Pair of Shapes

Given two similar shapes, x_1 and x_2 we would like to choose a rotation, θ , a scale s and a translation (t_x, t_y) mapping x_2 onto $M(x)$ so as to minimise the weighted sum

$$E = (x_1 - M(x_2))^T W (x_1 - M(x_2)) \quad (1)$$

$$\text{where } M \begin{pmatrix} x_{jk} \\ y_{jk} \end{pmatrix} = \begin{pmatrix} (s \cos \theta)x_{jk} - (s \sin \theta)x_{jk} + t_x \\ (s \sin \theta)x_{jk} + (s \cos \theta)x_{jk} + t_y \end{pmatrix} \quad (2)$$

and W is a diagonal matrix of weights for each point.
If we write

$$a_x = s \cos \theta \quad a_y = s \sin \theta$$

then least squares approach (differentiating with respect to each of the variables a_x , a_y , t_x , t_y) leads to a set of four linear equations;

$$\begin{pmatrix} X_2 & -Y_2 & W & 0 \\ Y_2 & X_2 & 0 & W \\ Z & 0 & X_2 & Y_2 \\ 0 & Z & -Y_2 & X_2 \end{pmatrix} \begin{pmatrix} a_x \\ a_y \\ t_x \\ t_y \end{pmatrix} = \begin{pmatrix} X_1 \\ Y_1 \\ C_1 \\ C_2 \end{pmatrix} \quad (14)$$

where

$$X_i = \sum_{k=0}^{n-1} w_k x_{ik} \quad Y_i = \sum_{k=0}^{n-1} w_k y_{ik} \quad (15)$$

$$Z = \sum_{k=0}^{n-1} w_k (x_{2k}^2 + y_{2k}^2) \quad W = \sum_{k=0}^{n-1} w_k \quad (16)$$

$$C_1 = \sum_{k=0}^{n-1} w_k (x_{1k}x_{2k} + y_{1k}y_{2k}) \quad (17)$$

$$C_2 = \sum_{k=0}^{n-1} w_k (y_{1k}x_{2k} - x_{1k}y_{2k}) \quad (18)$$

These can be solved for a_x , a_y , t_x , and t_y using standard matrix methods.

References

- [1] R. Chin and C.R. Dyer, Model-Based Recognition in Robot Vision. Computing Surveys 1986; Vol 18, No 1
- [2] W.E.L. Grimson, Object Recognition by Computer : The Role of Geometric Constraints, The MIT Press, Cambridge, MA, USA, 1990.
- [3] A.L. Yuille, D.S. Cohen and P. Hallinan, Feature extraction from faces using deformable templates, Proc. Computer Vision and Pattern Recognition (1989) pp104–109.
- [4] P. Lipson, A.L. Yuille, D. O’Keeffe, J. Cavanaugh, J. Taaffe and D. Rosenthal, Deformable Templates for Feature Extraction from Medical Images, Proceedings of the First European Conference on Computer Vision (Lecture Notes in Computer Science, ed. O. Faugeras, pub. Springer-Verlag) 1990 pp413–417.
- [5] M. Kass, A. Witkin and D. Terzopoulos, Snakes: Active Contour Models. First International Conference on Computer Vision, pub. IEEE Computer Society Press, 1987, pp 259–268.
- [6] L.H. Staib and J.S. Duncan, Parametrically Deformable Contour Models. IEEE Computer Society conference on Computer Vision and Pattern Recognition, San Diego, 1989
- [7] D. Terzopoulos and D. Metaxas, Dynamic 3D Models with Local and Global Deformations : Deformable Superquadrics. IEEE Trans. on Pattern Analysis and Machine Intelligence 1991; Vol.13 No.7 pp703–714.
- [8] A. Pentland and S. Sclaroff, Closed-Form Solutions for Physically Based Modelling and Recognition. IEEE Trans. on Pattern Analysis and Machine Intelligence 1991; Vol.13 No.7 pp703–714.
- [9] F.L. Bookstein, Morphometric Tools for Landmark Data. Cambridge University Press, 1991.
- [10] K.V. Mardia, J.T. Kent and A.N. Walder, Statistical Shape Models in Image Analysis. Proceedings of the 23rd Symposium on the Interface, Seattle 1991, pp 550–557.
- [11] J.C. Gower, Generalized Procrustes Analysis. Psychometrika. 40, 1975, 33–51.
- [12] T.F. Cootes, D. Cooper, C.J. Taylor and J. Graham, A Trainable Method of Parametric Shape Description. Proc. BMVC 1991 pub. Springer-Verlag, pp54–61.
- [13] K. Fukunaga and W.L.G. Koontz, Application of the Karhunen-Loeve Expansion to Feature Selection and Ordering. IEEE Trans. on Computers 1970; 4.
- [14] J. Graham, T.F. Cootes, D.Cooper and C.J. Taylor, VISAGE Progress Report – Deliverable D4, Wolfson Image Analysis Unit, Manchester University 1992.
- [15] T.F. Cootes and C.J. Taylor, Active Shape Models – ‘Smart Snakes’. This Volume.
- [16] A. Hill, T.F. Cootes and C.J. Taylor, A Generic System for Image Interpretation Using Flexible Templates. This Volume.
- [17] A. Lanitis, Optical Character Recognition of Hand-written Characters using Flexible Templates. Internal Report, Wolfson Image Analysis Unit, Manchester University 1992.