COMPUTER VISION AND IMAGE UNDERSTANDING
Vol. 61, No. 1, January, pp. 38–59, 1995

Active Shape Models—Their Training and Application

T. F. COOTES, C. J. TAYLOR, D. H. COOPER, AND J. GRAHAM*

Department of Medical Biophysics, University of Manchester, Oxford Road, Manchester M13 9PT, England

Received July 29, 1992; accepted April 12, 1994

Model-based vision is firmly established as a robust approach to recognizing and locating known rigid objects in the presence of noise, clutter, and occlusion. It is more problematic to apply model-based methods to images of objects whose appearance can vary, though a number of approaches based on the use of flexible templates have been proposed. The problem with existing methods is that they sacrifice model specificity in order to accommodate variability, thereby compromising robustness during image interpretation. We argue that a model should only be able to deform in ways characteristic of the class of objects it represents. We describe a method for building models by learning patterns of variability from a training set of correctly annotated images. These models can be used for image search in an iterative refinement algorithm analogous to that employed by Active Contour Models (Snakes). The key difference is that our *Active Shape Models* can only deform to fit the data in ways consistent with the training set. We show several practical examples where we have built such models and used them to locate partially occluded objects in noisy, cluttered images. © 1995 Academic Press, Inc.

1. INTRODUCTION

We address the problem of locating examples of known objects in images. 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. In medical applications, for instance, the shape of organs can vary considerably through time and between individuals. In addition, 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 objects [3–23].

In this paper we present new methods of building and using flexible models of image structures whose shape can vary. The models are able to capture the natural variability within a class of shapes and can be used in image search to find examples of the structures that they represent. Previous approaches have allowed models to deform, but have not tailored the variability to the class of shapes concerned—the models are not specific. Our main contribution is to describe how to create models which allow for considerable variability but are still specific to the class of structures they represent.

Our technique relies upon each object or image structure being represented by a set of points. The points can represent the boundary, internal features, or even external ones, such as the center of a concave section of boundary. Points are placed in the same way on each of a training set of examples of the object. This is done manually, though tools are available to aid the user. The sets of points are aligned automatically to minimize the variance in distance between equivalent points. By examining the statistics of the positions of the labeled points a "Point Distribution Model" is derived. The model gives the average positions of the points, and has a number of parameters which control the main modes of variation found in the training set.

Given such a model and an image containing an example of the object modeled, image interpretation involves choosing values for each of the parameters so as to find the best fit of the model to the image. We describe a technique which allows an initial very rough guess for the best shape, orientation, scale, and position to be refined by comparing the hypothesized model instance with image data, and using differences between model and image to deform the shape. We have previously described how to obtain the initial guess [7]. The method has similarities with the Active Contour Models (or snakes) of Kass *et al.* [3], but differs in that global shape constraints are applied; to make this distinction clear we have adopted the term Active *Shape* Models. The key point is that instances of the models can only deform in ways found in the training set.

Our results demonstrate that the method for constructing models combined with the active matching technique provides a systematic and effective paradigm for the interpretation of complex images. In the remainder of the paper we review some of the relevant literature, describe the modeling method, and show examples of

* E-mail: bim@uk.ac.man.mb.wiau. Fax: 061 275 5145.

1077-3142/95 $6.00
Copyright © 1995 by Academic Press, Inc.
All rights of reproduction in any form reserved.

trained models. The active matching technique is described and results are given, showing how the models can be used to interpret images.

2. BACKGROUND

There is a substantial literature describing the use of flexible models or deformable templates to aid image interpretation. Such models usually have a number of parameters to control the shape and pose of all or parts of the model. We give a brief review of some of the most significant work, which relates mainly to two-dimensional images.

2.1. *"Hand Crafted" Models*

Flexible models can be built up from simple subcomponents, such as circles, lines, or arcs, which are allowed some degree of freedom to move around relative to one another, and possibly change scale and orientation. Yuille *et al.* [5] model parts of the face, such as the eyes and mouth, in this way. When attempting to fit a model to an image they first obtain an approximate fit, which they refine by changing different parts of the model, one at a time. Lipson *et al.* [6] apply a similar scheme to map ellipitical models of vertebrae onto CT images of the spine. Hill *et al.* [7] use a handcrafted model of the heart in combination with Genetic Algorithm search to find the left ventricle in echocardiograms.

Although such models can capture detailed knowledge of expected shapes, the approach lacks generality. It is necessary to design both a new model and a scheme for fitting to images for each application.

2.2. *Articulated Models*

A number of authors consider articulated models built from rigid components connected by sliding or rotating joints. Beinglass and Wolfson [8] describe a scheme for locating such objects using a Generalized Hough Transform with the point of articulation as the reference point for each subpart. Connected subparts then vote for the same reference point. Grimson [2] has extended his "interpretation tree" approach to object recognition to include some articulations, and reviews other work along the same lines. This approach is only applicable to a restricted class of variable shape problems.

2.3. *Active Contour Models ("Snakes")*

Kass *et al.* [3] describe flexible contour models which are attracted to image features. These energy minimizing spline curves are modeled as having stiffness and elasticity and are attracted toward features such as lines and edges. Constraints can be applied to ensure that they remain smooth and to limit the degree to which they can be bent.

Snakes can be considered as parameterized models, the parameters being the spline control points. They are usually free to take almost any smooth boundary with few constraints on their overall shapes. The idea of fitting by using image evidence to apply forces to the model and minimizing an energy function is effective.

Hinton *et al.* [4] describe a type of spline snake governed by a number of control points which have preferred "home" locations to give the snake a particular default shape. Deformations are caused by moving the control points away from their "home" locations. Although the average shape of an object is represented, the modes of shape variation are only coarsely defined by the number and position of control points.

2.4. *Fourier Series Shape Models*

Scott [9] proposes a method of modeling shapes by an expansion of trigonometric functions,

$$\begin{aligned} x &= x_0 + \sum_n a_n \sin(n\theta + \phi_n) \\ y &= y_0 + \sum_n b_n \sin(n\theta + \psi_n). \end{aligned} \tag{1}$$

The shape produced is a function of the parameters a_n, b_n, ϕ_n, ψ_n. By varying the parameters and the number of terms used, different shapes can be generated. Scott shows how to fit such a shape model to image data by varying the parameters so as to minimize an energy term. The model is almost infinitely deformable, and contains no prior shape information. Staib and Duncan [10] describe similar Fourier models, and use them to interpret medical images. They derive distributions for each of the parameters over a training set and while fitting the model to an image maximize a probability measure determining how likely it is that the current example is the desired object. Bozma and Duncan [11] describe how such a technique can be used to model organs in medical images. A given shape is represented by a list of values for the parameters and is deformed by varying the parameters from these values. They describe ways of incorporating relationships between several flexible objects by applying constraints to the parameters of the models.

Trigonometric basis functions are not suitable for describing general shapes; for example, using a finite number of terms, they can only approximate a square corner. The relationship between variations in shape and variations in the parameters of the trigonometric expansion is not straightforward.

2.5. *Statistical Models of Shape*

A number of workers have studied the distributions of sets of "landmark" points which mark significant positions on an object. Goodall [14] discusses the registration

of shapes in arbitrary dimensions and the use of Procrustes analysis for estimating the mean shape and the covariances between landmark point coordinates and for assessing the differences between sets of shapes.

Grenander *et al.* [12] describe a method of representing a shape as a set of boundary points connected by arcs, with a statistical model of the relationships between neighboring arcs. They show how a model of a hand outline can be manipulated to fit degraded images of hands. They do this by considering sections of the boundary, and determine their most probable positions given the rest of the boundary and the local image data. By traversing the boundary in a number of sweeps the process iterates to a solution. Grenander and Miller [13] have extended this work to include gray-level information and multiple models. Mardia *et al.* [15] do something similar, representing the boundary of a shape as a sequence of points with distributions related by a covariance matrix. They too cycle through the points to find the most likely position given the image and the current shape. Both Grenander *et al.* and Mardia *et al.* represent shapes as sets of points in the complex plane. The points can vary about their means following a normal distribution with covariance matrix **S**, where **S** is modeled using either first order conditional autoregressive (CAR) models or Toeplitz covariance matrices. In our work we use a similar underlying model, but avoid any dependence of the sequence of points, thereby capturing more global shape properties. We also use principal component analysis to simplify the structure of the covariance matrix.

2.6. *Finite Element Models*

Finite element methods can be used to model variable image objects as physical entities with internal stiffness and elasticity. Pentland [18] and Pentland and Sclaroff [19] use three dimensional models which act like lumps of elastic clay. They derive modes of vibration of a suitable base shape, such as an ellipsoid, and build up shapes from different modes of vibration. The first modes are large-scale variations of shape; the higher order modes are more localized. To model human heads they use the first 30 modes. They can fit models to range data by an interative process, and can compare different heads by comparing the parameters. Terzopoulos and Metaxas [20] present a similar idea using deformable superquadrics. Nastar and Ayache [21] use a finite element approach using the vibrational modes of an example of the shape to be modelled. Karaolani *et al.* [22, 23] use finite element methods to model two dimensional objects, giving an alternative to the 'snakes' of Kass *et al.* [3].

All these methods have the advantage that the models are relatively easy to construct and allow a compact parametric representation of a family of shapes.

2.7. *The Need for Better Models*

We have described a number of existing methods for creating and using deformable models to interpret images containing structures of variable form. It is important to explain at this point why we believe a new method is required. We argue that the key issue is one of model specificity. It is necessary that a deformable model should be able to accommodate the range of variation found in the objects it is used to represent, but not sufficient. The principal role of a model is to facilitate robust automatic interpretation even in images which are noisy or cluttered or where parts of the objects of interest may be occluded. If the model is nonspecific, in the sense that it is able to deform so as to represent objects which are not valid examples of the class to be recognized, then this robustness is compromised. Our objective has been to develop models which can *only* deform in ways which are characteristic of the objects they represent. In general, the mechanisms which give rise to variability are insufficiently well understood to allow a theoretical model of deformability to be proposed. The only realistic approach is to "learn" specific patterns of variability from a representative training set of the structures to be modeled. Others have attempted something similar by placing limits on model parameters on the basis of their distributions determined from a training set. If, as is generally the case, the model parameters are correlated over the training set, this approach does not effectively restrict the shapes which can be generated to ones similar to those found in the original training set (see Fig. 1). Our approach is to find a basis for shape representation in which the shape parameters are uncorrelated over the training set. In this case simple limits on each parameter constrain the model to generate shapes similar to those in the training set. We also show that it is straightforward to use these models in image interpretation.

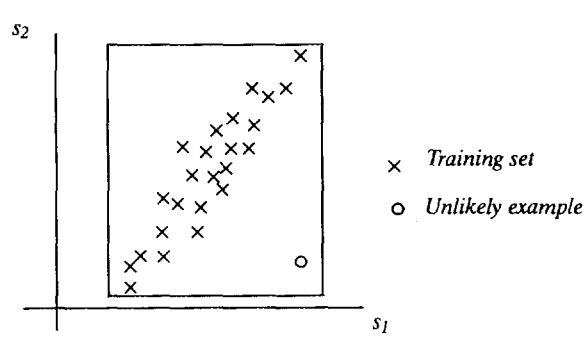

FIG. 1. If two or more shape parameters (s_1 and s_2) are correlated over a set of shapes then simple ranges to the parameters do not restrict shapes to ones similar to those in the original set.

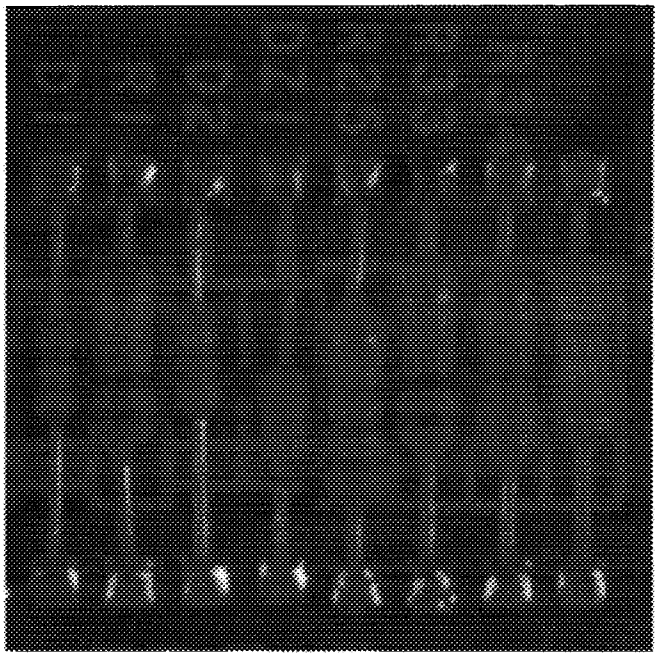

FIG. 2. Image of printed circuit board showing examples of resistors.

3. POINT DISTRIBUTION MODELS

Suppose we wish to derive a model to represent the shapes of resistors as they appear on printed circuit boards, as shown in Fig. 2. Different examples of resistor have sufficiently different shapes so that a rigid model would not be appropriate. Figure 3 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 typical variability, using the examples in Fig. 3 as a training set. We achieve this by representing each example as a set of labeled 'landmark' points, calculating the mean positions of the points and the main ways in which the points from each example tend to vary from the mean.

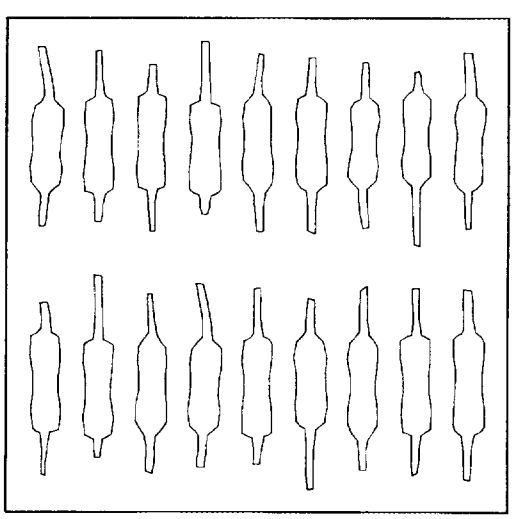

FIG. 3. Examples of resistor shapes from a training set.

3.1. *Labeling 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 Fig. 4. This must be done for each shape in the training set. The labeling of the points is important. Each labeled 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 modeling how different labeled points tend to move together as the shape varies. If the labeling is incorrect, with a particular point placed at different sites on each training shape, the method will fail to capture shape variability reliably. In the examples shown below the points were either placed manually on each image, or tools were used to mark points on boundaries segmented by hand. It is worth noting that the points are only placed manually during the training phase; it is *not* necessary to find these points in advance when the models are used for image interpretation—we describe later how this is achieved implicitly using an automatic method.

Bookstein [16, 17] labeled significant points in images of biological and medical specimens in order to examine and measure shape changes which could be correlated with other factors. We use representative points to capture shape constraints and build models which may be used to construct plausible new examples of the shape for use in image interpretation. Bookstein calls his representative points "landmark points" and describes them in terms of their usefulness. For our purposes they can be reduced to three different types:

1. points marking parts of the object with particular application-dependent significance, such as the center of an eye in the model of a face or sharp corners of a boundary;

2. points marking application-independent things, such as the highest point on an object in a particular orientation, or curvature extrema;

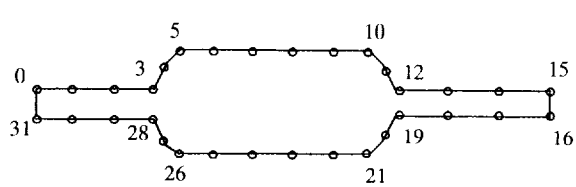

FIG. 4. Thirty-two point model of the boundary of a resistor.

3. other points which can be interpolated from points of type 1 and 2; for instance, points marked at equal distances a round a boundary between two type 1 landmarks.

On the resistor shown in Fig. 4 points 0, 3, 5, 10 and so on mark easily identified features and so are points of type 1. The other points are equally spaced along the boundaries between, and so are of type 3. Landmark points of type 1 are preferable to those of type 2, since they are in general easier to identify precisely. However, points of type 2 and 3 are almost always necessary to define the boundary of a flexible shape in sufficient detail to be useful.

It is important to note that landmark points can be used to describe single objects or sets of spatially related objects—the points can come from several different components of a structure. Typically we use boundary points, and associate boundary segments with appropriate pairs of landmark points. Although we have implemented and investigated the use of interpolating splines to generate boundaries using a minimal set of landmarks we find that simply using enough type 3 points to describe the curve to sufficient accuracy is as effective and is computationally more efficient.

3.2. *Aligning the Training Set*

Our modeling method works by examining the statistics of the coordinates of the labeled points over the training set. In order to be able to compare equivalent points from different shapes, they must be aligned with respect to a set of axes. We achieve the required alignment by scaling, rotating, and translating the training shapes so that they correspond as closely as possible. We aim to minimize a weighted sum of squares of distances between equivalent points on different shapes. This is a modification of the Procrustes method [24].

We first consider aligning a pair of shapes. Let $\mathbf{x}_i$ be a vector describing the n points of the ith shape in the set:

$$\mathbf{x}_i = (x_{i0}, y_{i0}, x_{i1}, y_{i1}, \ldots, x_{ik}, y_{ik}, \ldots, x_{in-1}, y_{in-1})^T. \quad (2)$$

Let $M(s, \theta)[\mathbf{x}]$ be a rotation by θ and a scaling by s. Given two similar shapes, $\mathbf{x}_i$ and $\mathbf{x}_j$ we can choose θ_j, s_j and a translation (t_{xj}, t_{yj}) mapping $\mathbf{x}_i$ onto $M(s_j, \theta_j)[\mathbf{x}_j] + \mathbf{t}_j$ so as to minimize the weighted sum

$$E_j = (\mathbf{x}_i - M(s_j, \theta_j)[\mathbf{x}_j] - \mathbf{t}_j)^T \mathbf{W} (\mathbf{x}_i - M(s_j, \theta_j)[\mathbf{x}_j] - \mathbf{t}_j), \quad (3)$$

where

$$M(s, \theta)\begin{bmatrix} x_{jk} \\ y_{jk} \end{bmatrix} = \begin{pmatrix} (s\cos\theta)x_{jk} - (s\sin\theta)y_{jk} \\ (s\sin\theta)x_{jk} + (s\cos\theta)y_{jk} \end{pmatrix}, \quad (4)$$

$$\mathbf{t}_j = (t_{xj}, t_{yj}, \ldots, t_{xj}, t_{yj})^T, \text{ and} \quad (5)$$

$\mathbf{W}$ 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 kth point using

$$w_k = \left(\sum_{l=0}^{n-1} V_{R_{kl}} \right)^{-1}. \quad (6)$$

If a point tends to move around a great deal 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.

We use the following algorithm to align a set of N shapes;

- Rotate, scale, and translate each shape to align with the first shape in the set.
- **Repeat**
  - Calculate the mean shape from the aligned shapes.
  - Normalize the orientation, scale and origin of the current mean to suitable defaults.
  - Realign every shape with the current mean.
- **Until** the process converges.

Normalizing the mean to a default scale and pose during each iteration is required to ensure that the algorithm converges. Without this there are in effect $4(N-1)$ constraints on $4N$ variables (θ, s, t_x, t_y for each of the N shapes) and the algorithm is ill-conditioned—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 center of gravity, an orientation such that a particular part of the shape is at the top, and a scale such that the distance between two selected points is one unit. Note that normalizing the current mean shape and then aligning the shapes to match is not the same as normalizing each individual shape. If every shape were normalized in scale by setting the distance between a particular two points to be one unit, artificial correlations might be forced upon the set, distorting the model. However, if each shape is aligned with the mean, each will have a scale similar to

that of the mean. In this case the landmark point positions will be chosen to best match the mean, rather than rigidly imposed. This leads to better models.

The convergence condition in the alignment procedure 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 show 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. We have considered direct methods of solution but have found problems with numerical stability. Since computational efficiency is not an issue during model construction the iterative method is adequate for our purposes.

3.3. *Capturing the Statistics of a Set of Aligned Shapes*

In Fig. 5 the coordinates of the some of the vertices of the aligned resistor shapes are plotted, with the mean shape overlaid. It can be seen that some of the vertices show little variability over the training set, while others form more diffuse "clouds." The Point Distribution Model (PDM) seeks to model the variation of the coordinates within these clouds. However, it must be remembered that landmarks do not move about independently—their positions are partially correlated.

Each example in the training set, when aligned, can be represented by a single point in a $2n$ dimensional space (see Eq. (2)). Thus a set of N example shapes gives a cloud of N points in this $2n$ dimensional space. We assume that these points lie within some region of the space, which we call the "Allowable Shape Domain," and that the points give an indication of the shape and size of this region. Every $2n$-D point within this domain gives a set of landmarks whose shape is broadly similar to that of those in the original training set. Thus by moving about the Allowable Shape Domain we can generate new shapes in a systematic way. The approach given below attempts to model the shape of this cloud in a high dimensional space, and hence to capture the relationships between the positions of the individual landmark points. We make the assumption that the cloud is approximately ellipsoidal, and proceed to calculate its center (giving a mean shape) and its major axes, which give a way of moving around the cloud. Later we will discuss the implications of this ellipsoid assumption breaking down.

Given a set of N aligned shapes, the mean shape, $\bar{\mathbf{x}}$ (the center of the ellipsoidal Allowable Shape Domain), is calculated using

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i \tag{7}$$

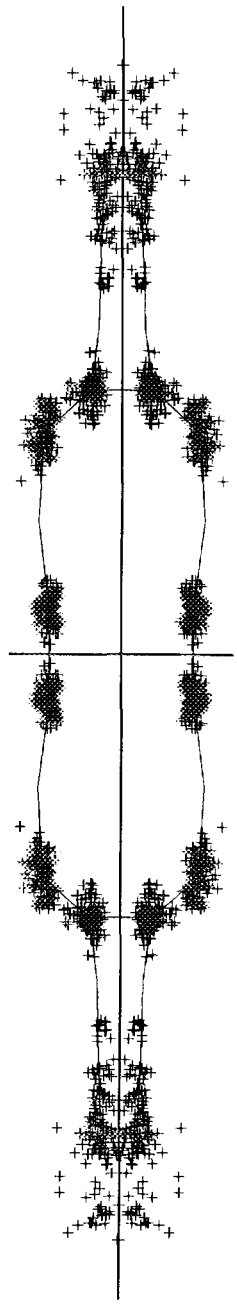

FIG. 5. Scatter of some points from aligned set of resistor shapes, with the mean shape overlaid.

The principal axes of a $2n$-D ellipsoid fitted to the data can be calculated by applying a principal component analysis (PCA) to the data [25]. Each axis gives a "mode of variation," a way in which the landmark points tend to move together as the shape varies. For each shape in the training set we calculate its deviation from the mean, $d\mathbf{x}_i$, where

$$d\mathbf{x}_i = \mathbf{x}_i - \bar{\mathbf{x}}. \tag{8}$$

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

$$\mathbf{S} = \frac{1}{N}\sum_{i=1}^{N} d\mathbf{x}_i \, d\mathbf{x}_i^{\mathrm{T}}. \tag{9}$$

The principal axes of the ellipsoid, giving the modes of variation of the points of the shape, are described by $\mathbf{p}_k$ ($k = 1, \ldots, 2n$), the unit eigenvectors of $\mathbf{S}$ such that

$$\mathbf{S}\mathbf{p}_k = \lambda_k \mathbf{p}_k \tag{10}$$

(where λ_k is the kth eigenvalue of $\mathbf{S}$, $\lambda_k \geq \lambda_{k+1}$),

$$\mathbf{p}_k^{\mathrm{T}}\mathbf{p}_k = 1. \tag{11}$$

It can be shown that the eigenvectors of the covariance matrix corresponding to the largest eigenvalues describe the longest axes of the ellipsoid, and thus the most significant modes of variation in the variables used to derive the covariance matrix. The variance explained by each eigenvector is equal to the corresponding eigenvalue [25]. Most of the variation can usually be explained by a small number of modes, t. This means that the $2n$ dimensional ellipsoid is approximated by a t dimensional ellipsoid, where t is chosen so that the original ellipsoid has a relatively small width along axes $t + 1$ and above. One method for calculating t is to chose the smallest number of modes such that the sum of their variances explains a sufficiently large proportion of λ_T, the total variance of all the variables, where

$$\lambda_T = \sum_{k=1}^{2n} \lambda_k. \tag{12}$$

Any point in our Allowable Shape Domain (i.e., any allowable shape) can be reached by taking the mean and adding a linear combination of the eigenvectors. The kth eigenvector affects point l in the model by moving it along a vector parallel to (dx_{kl}, dy_{kl}), which is obtained from the lth pair of elements in $\mathbf{p}_k$:

$$\begin{aligned}\mathbf{p}_k^{\mathrm{T}} = (&dx_{k0}, dy_{k0}, \ldots, \underline{dx_{kl}, dy_{kl}}, \ldots, \\ &dx_{kn-1}, dy_{kn-1}).\end{aligned} \tag{13}$$

An 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:

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b}, \quad \text{where} \tag{14}$$

$\mathbf{P} = (\mathbf{p}_1\, \mathbf{p}_2 \ldots \mathbf{p}_t)$ is the matrix of the first t eigenvectors, and $\mathbf{b} = (b_1\, b_2 \ldots b_t)^{\mathrm{T}}$ is a vector of weights.

The above equations allow us to generate new examples of the shapes by varying the parameters (b_k) within suitable limits, so the new shape will be similar to those in the training set. The parameters are linearly independent, though there may be nonlinear dependencies still present. The limits for b_k are derived by examining the distributions of the parameter values required to generate the training set. Since the variance of b_k over the training set can be shown to be λ_k, suitable limits are typically of the order of

$$-3\sqrt{\lambda_k} \leq b_k \leq 3\sqrt{\lambda_k}, \tag{15}$$

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

Alternatively, one can choose sets of parameters $\{b_1, \ldots, b_t\}$ such that the Mahalanobis distance (D_m) from the mean is less than a suitable value, D_{max}:

$$D_m^2 = \sum_{k=1}^{t} \left(\frac{b_k^2}{\lambda_k}\right) \leq D_{max}^2. \tag{16}$$

If each shape parameter is normally distributed then D_m will be chi-squared distributed, and D_{max} can be chosen to include a suitably large proportion of possible realizations.

3.4. *Practical Examples*

The techniques described above have been used to generate point distribution models (PDMs) for both manmade and biological objects. We present results for the set of resistor shapes shown in Fig. 3, a set of heart ventricle shapes, and a set of hand shapes. Other examples have been described elsewhere [33, 34].

3.4.1. *Resistor Model.* The resistor shapes illustrated in Fig. 3 were aligned using the method described above, with the mean shape scaled so the average distance of the points from their center of gravity was one unit. Figure 5 shows the mean shape. The most significant eigenvalues of the covariance matrix derived are shown in Table 1.

Figure 6 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 nonlinear 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 Figs. 7–9. Each of the parameters represents a mode of variation of the shape which can frequently be associated with an intuitive de-

TABLE 1
Eigenvalues of the Covariance Matrix Derived from a Set of Resistor Shapes

Eigenvalue	$\frac{\lambda_i}{\lambda_T} \times 100\%$
λ_1	66%
λ_2	8%
λ_3	5%
λ_4	4%
λ_5	3%
λ_6	3%

scription of the deformation. Compare Figs. 7–9 with Fig. 3. 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 present in the training set. Note that the apparently large variability in the positions of individual points in Fig. 3 is in fact highly constrained, and the overall variation in shape can be described by a small number of modes. This model has been used to locate resistors in images (see below).

3.4.2. *Heart Model.* Figure 10 shows examples from a set of 66 heart chamber boundaries obtained by asking a cardiologist to draw over echocardiogram images. Each

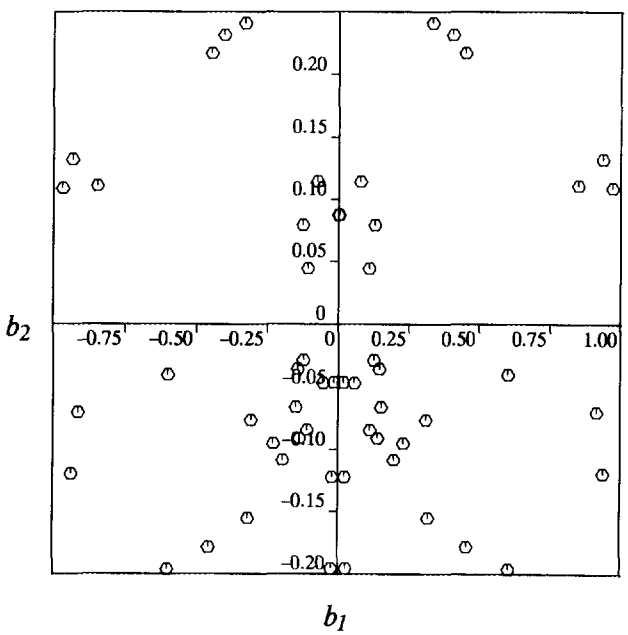

FIG. 6. Plot of b_1 vs b_2 for a training set of resistor shapes.

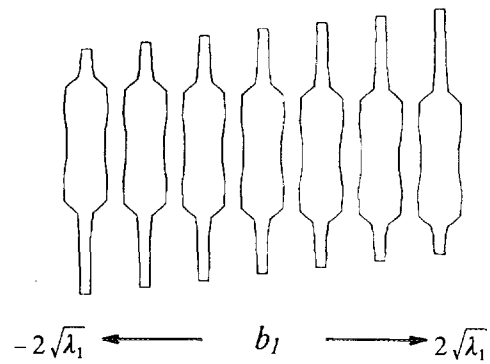

FIG. 7. Effects of varying the first parameter of the resistor model.

structure is represented by 96 points. This example shows how a single model can represent several shapes *and* the spatial relationships between them. The shape variation arises from two sources: the training set was derived from several individuals, and in each case images were taken from different stages in the cardiac cycle, during which the sizes and shapes of the heart chambers can change considerably. The points represent the boundary of the left ventricle, part of the boundary of the right ventricle, and part of the boundary of the left atrium (below the ventricle in the figures). Table 2 shows the eigenvalues of the covariance matrix obtained for the training set. Figure 11 suggests that b_1 and b_2 are again independent, and Fig. 12 shows reconstructed shapes obtained by varying the first four model parameters in turn. The first parameter varies the width of the shape. The second parameter varies the appearance of the septum (the wall separating the left from the right ventricle). The third and fourth parameters vary the shape of the left ventricle and the modeled part of the atrium below. It should be emphasized that these modes are derived entirely automatically, and arise from a statistical analysis of the variation in the data. This model has been used to locate the boundary of a

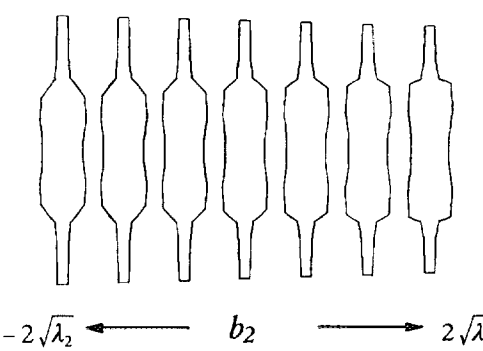

FIG. 8. Effects of varying the second parameter of the resistor model.

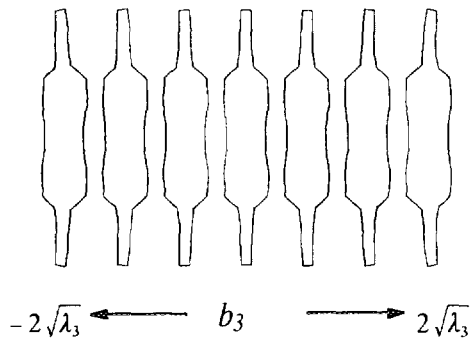

FIG. 9. Effects of varying the third parameter of the resistor model.

TABLE 2
Eigenvalues of the Covariance Matrix Derived from a Set of Heart Ventricle Shapes

Eigenvalue	$\frac{\lambda_i}{\lambda_T} \times 100\%$
λ_1	37%
λ_2	17%
λ_3	13%
λ_4	7%
λ_5	6%
λ_6	4%

heart ventricle in echocardiograms (see below, and also [26]).

3.4.3. *Hand Model.* A set of 18 hand shapes was generated from images of the right hand of one of the authors (Fig. 13). Each was represented by 72 points around the boundary. These were planted on the examples by locating 12 landmark 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 Fig. 14, and consist of combinations of movements of the fingers. Again, a compact parameterized model has been generated, which has been used to locate hands in images (see below).

3.4.4. *Worm Model—Limitations of the PDM.* We have found that the linear models described above are effective in a very broad range of applications. There are, however, some situations where the method breaks down.

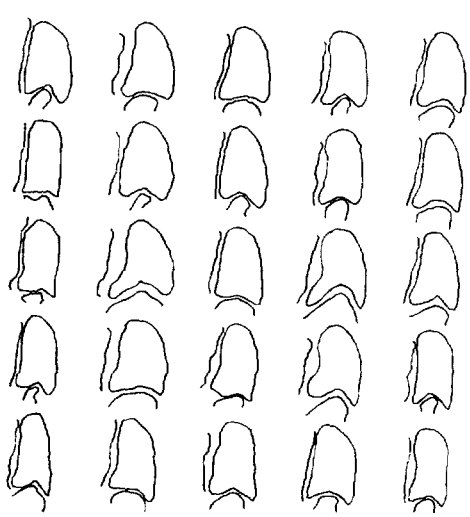

FIG. 10. Examples of heart ventricle shapes, each containing 96 points.

Each mode of variation in a PDM moves the landmark points along straight lines relative to the local coordinate system. In some cases the variations in a class of shapes would be better represented by moving points along curves. This can be especially important when bending or relative rotational effects occur in the class of example shapes. Consider, for instance, the examples from the set of "worm" shapes shown in Fig. 15. The set consists of 84 artificially generated shapes which have a fixed width but varying curvature and length, each represented by 12 labeled points (Fig. 16). Figure 17 shows the scatter of some of the points once the shapes have been aligned. The varying curvature leads to the points at the ends of the shape forming a curved cloud.

The shape formed by the mean positions of the labeled points does not have a constant width and is shorter than the aligned versions of the training shapes. The end points,

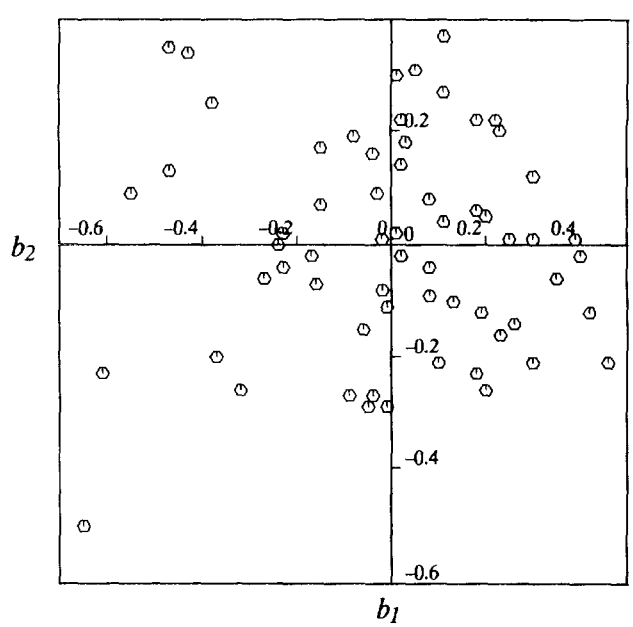

FIG. 11. b_1 vs b_2 for the training set of heart ventricle examples.

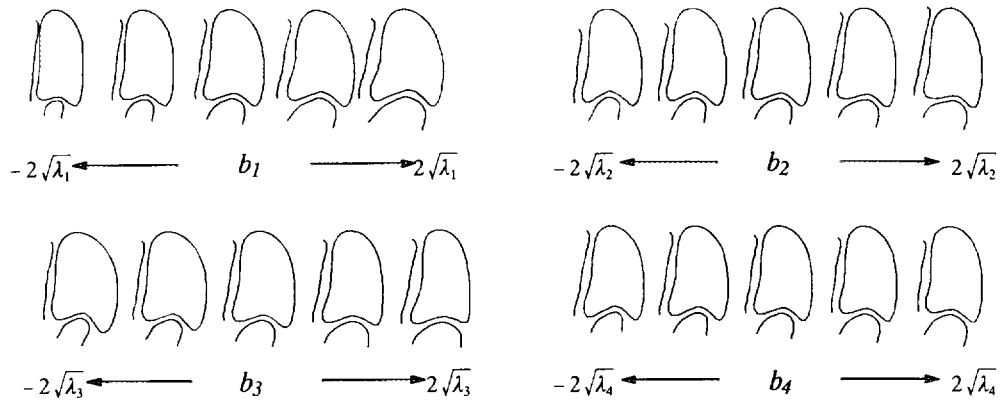

FIG. 12. Effects of varying each of the first four parameters of the heart ventricle model individually.

0 and 6, form curved clouds, the centroids of which do not lie inside the clouds. The mean shape generated in this way is thus not sufficiently similar to the training set to give a satisfactory model. The first three modes of variation of a PDM trained on this data are shown in Fig. 18. Ideally one would expect a model to have the first and second order curvature as its first two modes. The first mode of the PDM is an approximation to bending, generated by fitting straight lines to the curved "clouds" of points. The second mode gives the corrections required because the linear approximation is poor. The third mode of the model gives an approximation to second order bending. Figure 19 shows the relationship between the first two parameters b_1 and b_2. Though they are *linearly* independent, there are clearly nonlinear relationships present. One cannot choose the parameters independently and ex-

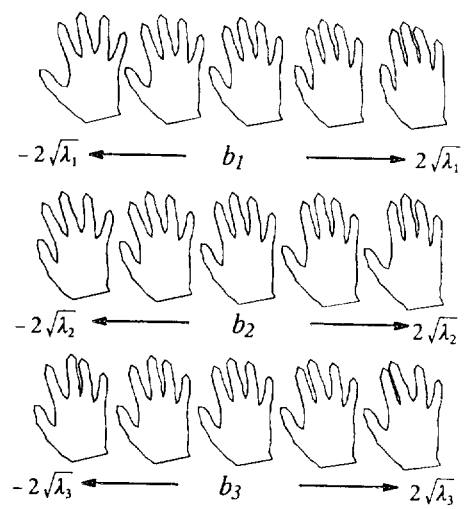

FIG. 14. Effects of varying each of the first three parameters of the hand model individually.

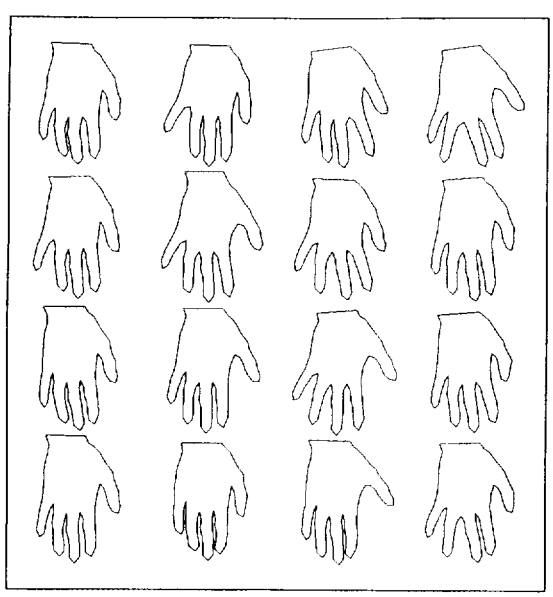

FIG. 13. Training set of hand shapes, each defined by 72 points.

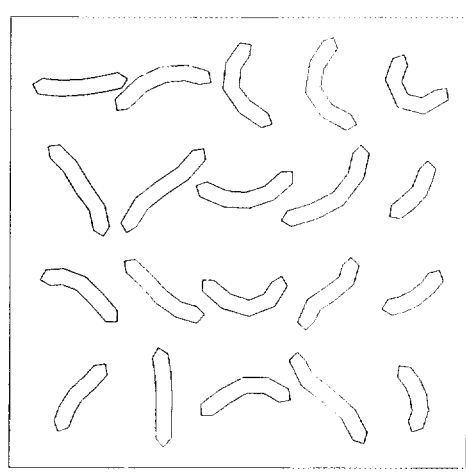

FIG. 15. Examples from a set of "worm" shapes.

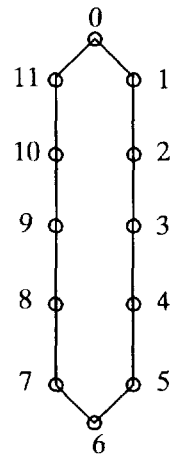

FIG. 16. Labeling of points in "worm" shapes.

pect to get a shape similar to those in the training set. We discuss ways in which this problem might be overcome at the end of the paper.

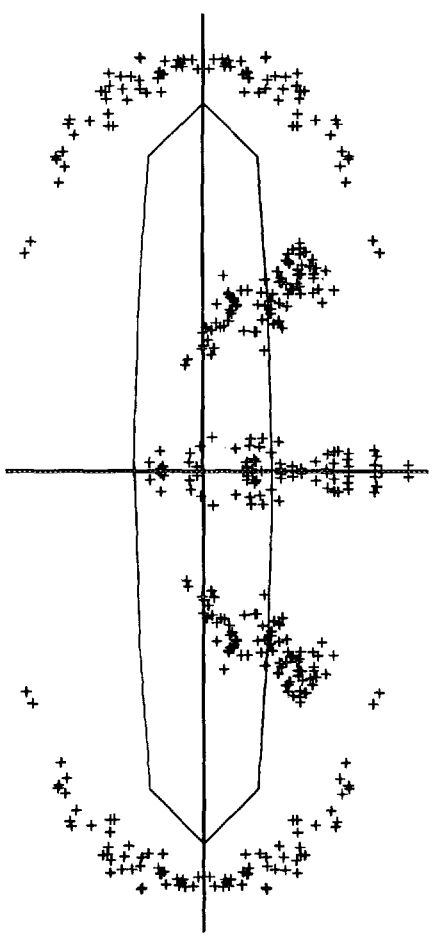

FIG. 17. Scatter of points 0, 2, 3, 4, and 6 from the aligned set of "worms," with the mean shape overlaid.

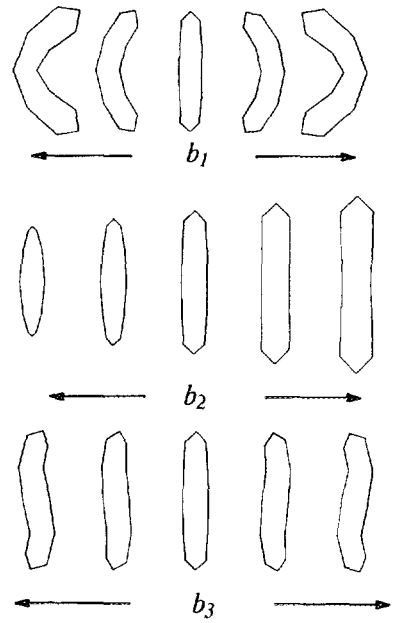

FIG. 18. Effects of varying the first three parameters (b_1, b_2, b_3) of the "worm" model individually.

4. USING POINT DISTRIBUTION MODELS IN IMAGE SEARCH—ACTIVE SHAPE MODELS

Having generated flexible models, we would like to use them in image search, to find new examples of modeled objects in images. This involves finding the shape and pose parameters which cause the model to coincide with

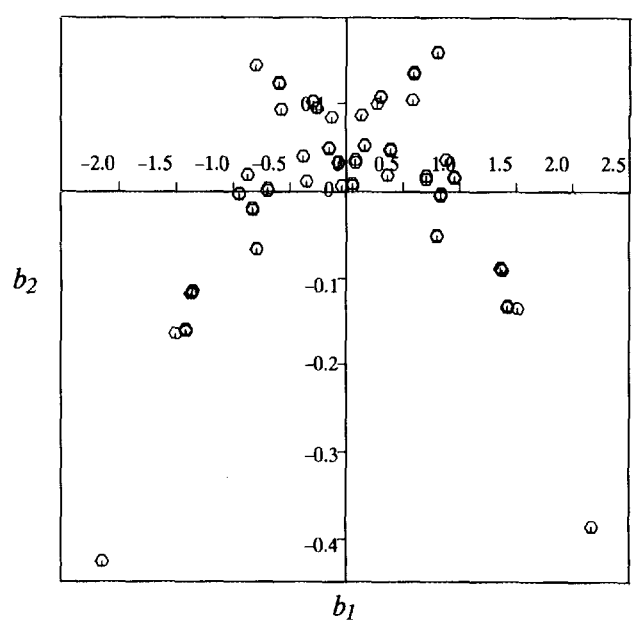

FIG. 19. b_1 vs b_2 for the training set of "worm" examples.

the structures of interest in the image. An instance of the model is given by

$$\mathbf{X} = M(s, \theta)[\mathbf{x}] + \mathbf{X}_c, \quad \text{where}$$
$$\mathbf{X}_c = (X_c, Y_c, X_c, Y_c, \ldots, X_c, Y_c)^{\mathrm{T}}$$
$$M(s, \theta)[\ ] \text{ is a rotation by } \theta \text{ and a scaling by } s, \text{ and} \quad (17)$$
$$(X_c, Y_c) \text{ is the position of the centre}$$
$$\text{of the model in the image frame.}$$

In this section we describe an iterative method for finding the appropriate $\mathbf{X}$ given a very rough starting approximation. Hill *et al.* have described elsewhere how Genetic Algorithm search can be used to find a good starting approximation quite rapidly [26, 7, 27]; this is applicable if there is no prior knowledge of the expected location of objects of interest. In practice, the starting value of $\mathbf{X}$ does not need to be very close to the final solution, so that, for many practical applications, the method below can be used on its own.

The idea of the iterative scheme is to place the current estimate of $\mathbf{X}$ into the image and examine a region of the image around each model point to determine a displacement which moves it to a better location. These local deformations are transformed into adjustments to the pose, scale, and shape parameters of the PDM. By enforcing limits on the shape parameters, global shape constraints can be applied ensuring the shape of the model example remains similar to those of the training set. The procedure is repeated until no significant changes result. Because the models attempt to deform to better fit the data, but only in ways which are consistent with the shapes found in the training set, we call them "Active Shape Models" or "Smart Snakes."

4.1. *Calculating a Suggested Movement for Each Model Point*

Given an initial estimate of the positions of a set of model points which we are attempting to fit to an image

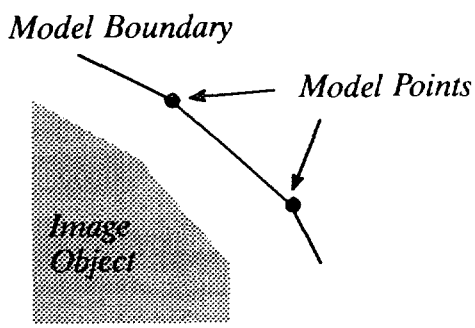

FIG. 20. Part of a model boundary approximating to the edge of an image object.

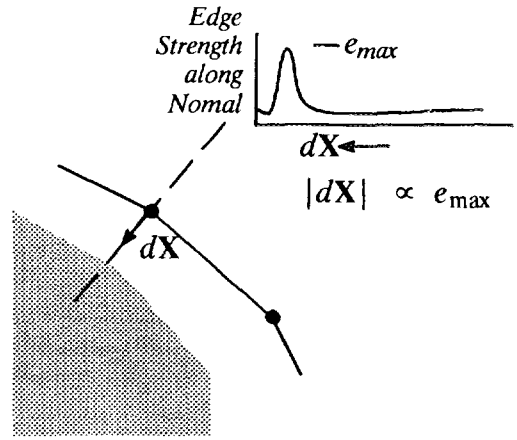

FIG. 21. Suggested movement of point is along normal to boundary, proportional to maximum edge strength on normal.

object we need to find a set of adjustments which will move each point toward a better position. When the model points represent the boundaries of objects (Fig. 20) this involves moving them toward the image edges. There are various approaches that could be taken. In the examples we describe below we use an adjustment along a normal to the model boundary toward the strongest image edge, with magnitude proportional to the strength of the edge (Fig. 21).

An alternative approach is to generate potential images such as those described by Kass *et al.* [3], possibly one for each model point, describing the likelihood of each point in the image being the model point. Adjustments to each point position can then be derived from the gradient of the potential image at the current estimate of the point's position.

However they are obtained, we denote the set of adjustments (Fig. 22) as a vector $d\mathbf{X}$, where

$$d\mathbf{X} = (dX_0, dY_0, \ldots, dX_{n-1}, dY_{n-1})^{\mathrm{T}}.$$

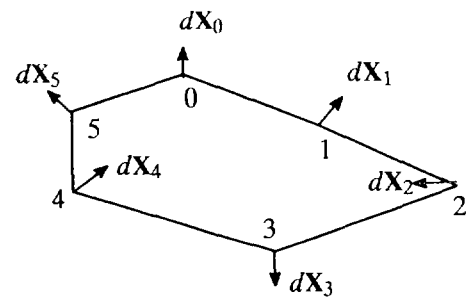

FIG. 22. Adjustments to a set of points.

4.2. *Computing Changes in the Pose and Shape Parameters*

We aim to adjust the pose and shape parameters of the model to move the points from their current locations in the image frame, $\mathbf{X}$, to be as close to the suggested new locations ($\mathbf{X} + d\mathbf{X}$) as can be arranged while still satisfying the shape constraints of the model. If the current estimate of the model is centered at (X_c, Y_c) with orientation θ and scale s we would like first to calculate how to update these parameters to better fit the image. This is achieved by finding the translation (dX_c, dY_c), rotation $d\theta$ and scaling factor $(1 + ds)$ which best map the current set of points, $\mathbf{X}$, onto the set of points given by ($\mathbf{X} + d\mathbf{X}$) using the method given in Appendix A.

Having adjusted the pose variables there remain residual adjustments which can only be satisfied by deforming the shape of the model. We wish to calculate the adjustments, $d\mathbf{x}$, in the local coordinate frame required to cause the points $\mathbf{X}$ to move by $d\mathbf{X}$ when combined with the effect of the new scale, rotation and translation parameters.

The initial position of the points in the image frame is given by Eq. (17),

$$\mathbf{X} = M(s, \theta)[\mathbf{x}] + \mathbf{X}_c.$$

We wish to calculate a set of residual adjustments $d\mathbf{x}$ in the local model coordinate frame such that

$$M(s(1 + ds), (\theta + d\theta)[\mathbf{x} + d\mathbf{x}] + (\mathbf{X}_c + d\mathbf{X}_c) = (\mathbf{X} + d\mathbf{X}). \tag{18}$$

Thus

$$M(s(1 + ds), \theta + d\theta)[\mathbf{x} + d\mathbf{x}] = (M(s, \theta)[\mathbf{x}] + d\mathbf{X}) - (\mathbf{X}_c + d\mathbf{X}_c)$$

and since

$$M^{-1}(s, \theta)[\ ] = M(s^{-1}, -\theta)[\ ]$$

we obtain

$$d\mathbf{x} = M((s(1 + ds))^{-1}, -(\theta + d\theta))[\mathbf{y}] - \mathbf{x}, \quad \text{where } \mathbf{y} = M(s, \theta)[\mathbf{x}] + d\mathbf{X} - d\mathbf{X}_c. \tag{19}$$

Equation (19) gives a way of calculating the suggested movements to the points $\mathbf{x}$ in the local model coordinate frame. These movements are not in general consistent with our shape model. In order to apply the shape constraints we transform $d\mathbf{x}$ into model parameter space, giving $d\mathbf{b}$, the changes in model parameters required to adjust the model points as closely to $d\mathbf{x}$ as is allowed. Equation (14) gives

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b}.$$

We wish to find $d\mathbf{b}$ such that

$$\mathbf{x} + d\mathbf{x} \approx \bar{\mathbf{x}} + \mathbf{P}(\mathbf{b} + d\mathbf{b}). \tag{20}$$

Since there are only t ($<2n$) modes of variation available and $d\mathbf{x}$ can move the points in $2n$ different degrees of freedom, we can only achieve an approximation to the deformation required.

Subtracting (14) from (20) gives

$$d\mathbf{x} \approx \mathbf{P}(d\mathbf{b})$$

so

$$d\mathbf{b} = \mathbf{P}^{\mathrm{T}} d\mathbf{x} \tag{21}$$

since $\mathbf{P}^{\mathrm{T}} = \mathbf{P}^{-1}$, as the columns of $\mathbf{P}$ are mutually orthogonal and of unit length.

It can be shown that Eq. (21) is equivalent to using a least-squares approximation to calculate the shape parameter adjustments, $d\mathbf{b}$.

4.3. *Updating the Pose and Shape Parameters*

The equations above allow us to calculate changes to the pose variables and adjustments, dX_c, dY_c, $d\theta$, and ds, to the shape parameters $d\mathbf{b}$ required to improve the match between an object model and image evidence. We apply these to update the parameters in an iterative scheme,

$$X_c \rightarrow X_c + w_t\, dX_c \tag{22}$$

$$Y_c \rightarrow Y_c + w_t\, dY_c \tag{23}$$

$$\theta \rightarrow \theta + w_\theta\, d\theta \tag{24}$$

$$s \rightarrow s(1 + w_s\, ds) \tag{25}$$

$$\mathbf{b} \rightarrow \mathbf{b} + \mathbf{W}_b\, d\mathbf{b}, \tag{26}$$

where w_t, w_s, and w_θ are scalar weights, and $\mathbf{W}_b$ is a diagonal matrix of weights, one for each mode. This can be the identity, or each weight can be proportional to the standard deviation of the corresponding shape parameter over the training set. The latter allows more rapid movement in modes in which there tend to be larger shape variations. We can ensure that the model only deforms into shapes consistent with the training set by placing limits on the values of b_k. A shape can be considered acceptable if the Mahalanobis distance D_{m} is less than a suitable constant, $D_{\max}$, say 3.0 (See Eq. (16)). This limit is calculated so that almost all the training examples satisfy Eq. (16).

The vector **b** should lie within a hyperellipsoid about the origin. If updating **b** using (26) leads to an implausible shape, i.e., $D_m > D_{max}$ and the point lies outside the ellipsoid, **b** can be rescaled to lie on the closest point of the allowed volume using

$$b_k \rightarrow b_k \cdot \left(\frac{D_{max}}{D_m}\right) (k = 1, \ldots, t). \quad (27)$$

Note that we have already applied implicit limits of zero to the weights of the eigenvectors truncated from our representation (i.e., $b_i = 0 \; \forall \; i > t$). Once the parameters have been updated, and limits applied where necessary, the updated positions of the model points can be calculated, and new suggested movements derived for each point. The procedure is repeated until no significant change results.

4.4. EXAMPLES USING ACTIVE SHAPE MODELS

The techniques described above have been used successfully in a number of applications, both industrial and medical [26, 27, 33]. Here we show results using the resistor, heart, and hand models described above.

In each case initial estimates of the position, orientation, and scale are made, and the shape parameters of the Active Shape Model (ASM) are initialized at zero ($b_i = 0 \; (i = 1, \ldots, t)$). Suggested movements for each model point are calculated by finding the strongest edge (of the correct polarity) along the normal to the boundary at the point (see 4.1 and Fig. 21). Adjustments to the parameters are calculated and applied, and the process is repeated.

4.4.1. *Locating Resistors.* We have constructed a Point Distribution Model of a resistor, representing its boundary using 32 points (Section 3.4.1). Figure 23 shows an image of part of a printed circuit board with the resistor boundary model superimposed as it iterates toward a component in the image. We interpolate an additional 32 points, one between each pair of model points around the boundary, and calculate adjustments to each point by finding the strongest edge along profiles 20 pixels long centred at each point. We use a shape model with 5 degrees of freedom. Each iteration of the ASM takes about 0.015 s on a Sun Sparc 10 Workstation.

The method is effective in maintaining the global shape constraints of the model and works well, given a sufficiently good starting approximation; we discuss methods of obtaining such initial hypotheses elsewhere [26, 27].

4.4.2. *Locating Heart Ventricles.* Figure 24a shows an example of an echocardiogram. The left ventricle is in the top right of the imaged region. Figure 24b shows the initial placement of an instance of the 96 point heart chamber model described above (Section 3.4.2). Figure 24c shows the ASM after 80 iterations. After 200 iterations (Fig. 24d) the model gives a good fit to the data. The shape model used has 12 degrees of freedom. The adjustments to each point are calculated using the strongest edge in a smoothed image along a profile 40 pixels long centered on the point. Each ASM iteration takes about 0.03 s on a Sun Sparc 10 workstation. In this example the model is able to infer the position of the parts of the boundary where there are missing data (for example, the top of the ventricle) by using the knowledge of the expected shape combined with information from the areas of the image where good evidence for the ventricle wall can be found. Without the prior knowledge of the shape given by the model it would not be possible to delineate the ventricle boundary accurately. Further medical applications of the method are described in [33].

4.4.3. *Locating Hands.* We have constructed a Point Distribution Model of a hand, representing the boundary using 72 points (Section 3.4.3). Figure 25 shows an image of one of the author's hands amid some clutter and occlusion, and an example of the model iterating towards it. We calculate adjustments to each point by finding the strongest edge on a profile 35 pixels long centred on the point. The shape model has 8 degrees of freedom, and each ASM iteration takes about 0.02 seconds on a Sun Sparc 10 Workstation. The result demonstrates that the method can deal with clutter and limited occlusion.

5. DISCUSSION

The examples given above illustrate the main features of our approach. Using a single method, specialized only by training with an appropriate set of examples, we have been able to locate automatically a range of structures in complex, noisy, and cluttered images. Other examples reported elsewhere include faces [36], handwritten characters [36], anatomical structures in magnetic resonance images of the brain and abdomen [33], vertebrae in radiographs [33], parts of the foot in pressure images [38] and all the parts in an automobile brake assembly [34]. We discuss below some of the issues which arise from this work, including areas where further development is required.

5.1. *Point Distribution Models*

5.1.1. *Choice of Model Points and Training Examples.* It is important that landmark points be placed on the training images as accurately as possible. If a point is not in the correct position on each shape, the model will be unable to correctly represent the position of that point—it will include terms describing the noise caused by errors in point location. It is equally important to arrange that all the examples used to train the model are

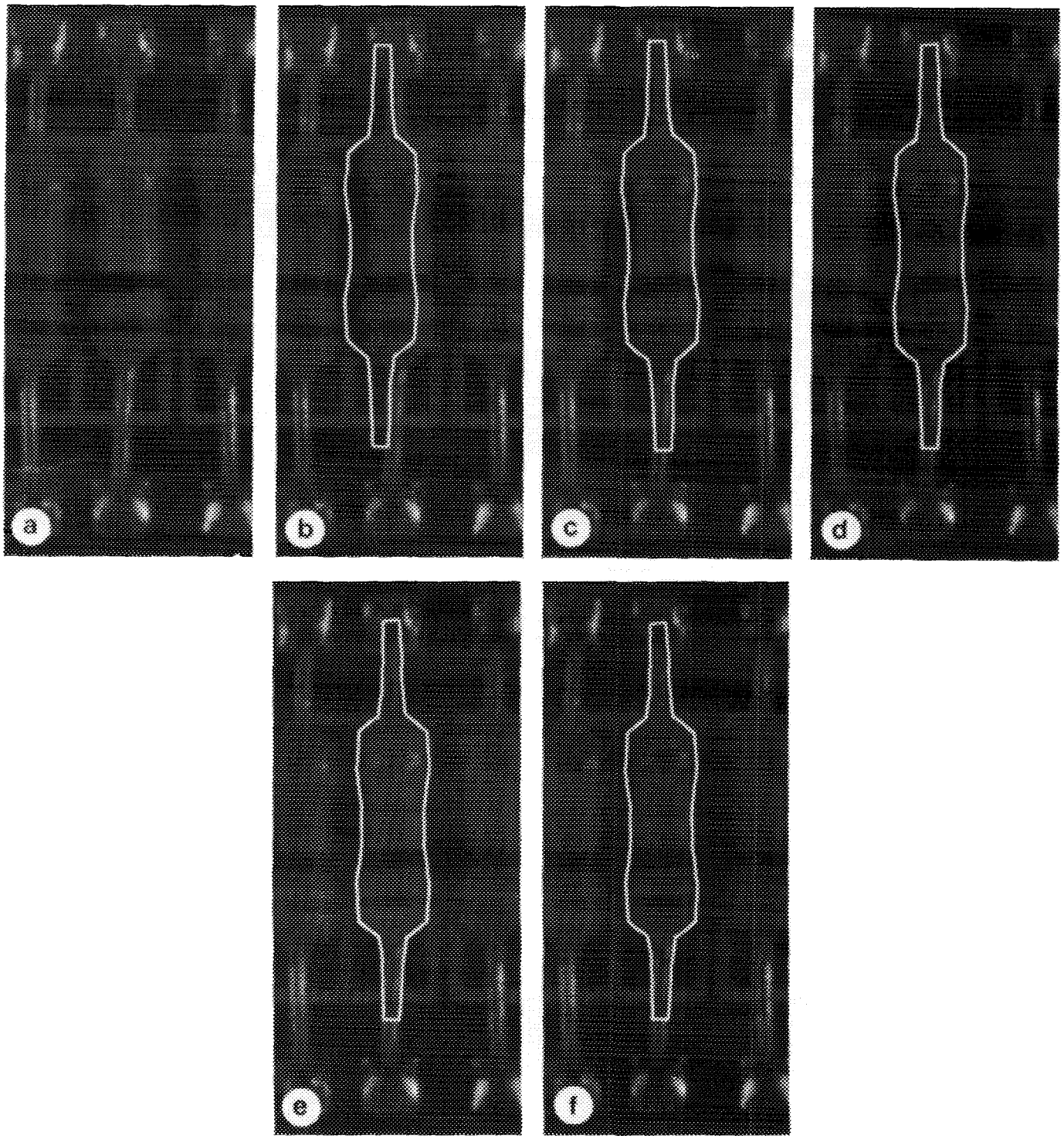

FIG. 23. Section of printed circuit board with resistor model superimposed, showing its initial position and its location after 30, 60, 90, and 120 iterations.

similarly aligned with respect to a set of axes, to ensure that the labeled points in different shapes are being compared correctly. In some cases an obvious alignment is apparent, but in others, particularly medical examples where the shapes of organs are very flexible, the automatic least-squares alignment method is essential.

Of course, placing every point by hand on every training image can be very time consuming. We are developing

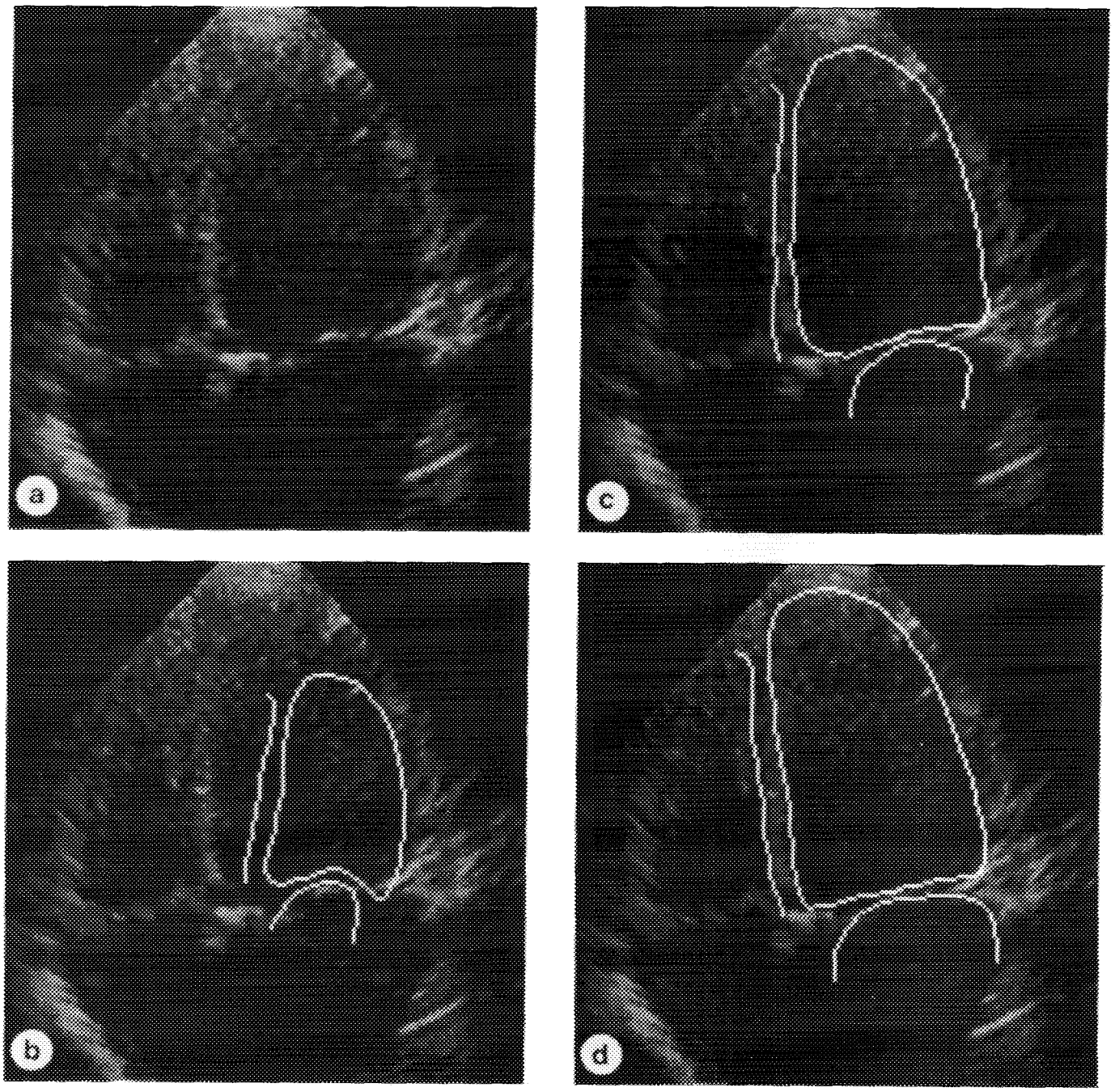

FIG. 24. Echocardiogram image with heart chamber boundary model superimposed, showing its initial position and its location after 80 and 200 iterations.

tools to ease the procedure. Techniques such as those described by Burr [29] and the Finite Element Models of Sclaroff and Pentland [30] or Nastar and Ayache [21] may be able to assist the user in locating point correspondences during this training phase.

In some cases occlusion and noise will lead to images in which some points cannot be accurately located. It is straightforward to adjust the calculation of mean shape (7) and the covariance matrix (9) to give a weighting to each point in each example in the training set. When some points are missing, the weights for known points can be set to unity; those for unknown points can be set to zero. As long as only a small proportion of points are missing in any one example, and no points are missing from all examples, it is still possible to build useful models.

In principle it is possible to "overtrain" a model. Sup-

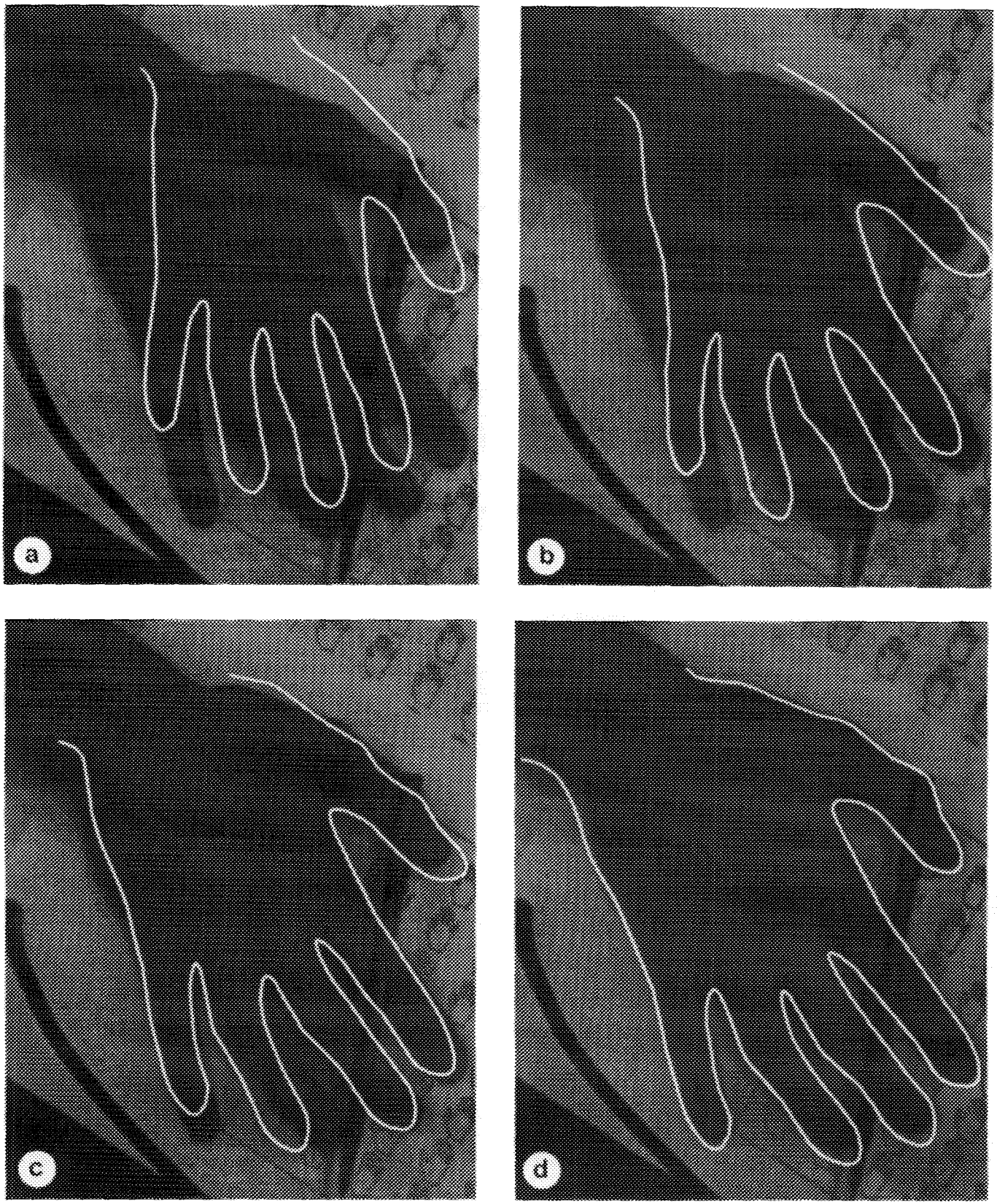

FIG. 25. Image of author's hand with hand model superimposed, showing its initial position and its location after 100, 200, and 350 iterations.

pose that a large proportion of the examples were close to the mean and there were only one or two examples demonstrating some particular form of shape variation. It is possible that when the number of modes to be used, t, is chosen, the mode which best describes the infrequent shape variation will be truncated, since it will explain only a small amount of the total variance. However, since the training examples are typically selected and labeled by hand, it is time consuming and inefficient to include many similar shapes—it is better to choose a variety of different shapes which cover the whole range of variations one is likely to observe (where such are available). It is at this stage that the expert knowledge of a human can play a part.

5.1.2. *Multipart Models.* The heart example illustrates an important fact—that the points used to construct a PDM and its derived ASM do not need to belong to a single object or shape. The connectivity of the points is not relevant to the construction of the PDM and is only used by the ASM to determine the direction of the local normal at each point during image search. The shapes of multiple subparts of a complex assembly and the spatial relationships between them can thus be represented by a single PDM. A significant advantage of handling shape and spatial relationships in a unified way is that correlations between the positions and shapes of subcomponents can be modeled; this is important, for example, in assemblies of interlinked mechanical components or in medical images where several organs are "packed" into the same cavity.

5.1.3. *Modeling Shape Variation.* We showed in Section 3.3 that each aligned shape can be considered as a single point in $2n$ dimensional space, and the whole training set as a cloud of points in this space. We attempt to model this cloud using the idea of an Allowable Shape Domain. For the search method to work effectively it is important that this domain be simply connected, and that we have a simple method of navigating around the domain. The assumption that the domain is an ellipsoid (or a box with the same axes) allows us to do this. However, under certain circumstances this is an inappropriate model. When there is a large degree of bending or relative rotation in the training set, nonlinear relationships between landmarks can give the cloud in the $2n$ dimensional space a "banana" shape or worse. Under these circumstances, as was demonstrated in Section 3.4.4, the ellipsoidal assumption gives a shape model which can generate shapes badly distorted when compared with those from the training set. The model is not as *specific* as one would like, and only a subset of the shapes it can generate would be considered "legal." In some situations this is not disastrous. For instance, the worm model given can be used successfully to locate examples of worms in images, but the models are more susceptible to being distorted by noise or clutter than a more specific model would be.

A more general model of the allowable shape domain could lead to more specific shape models. We have experimented using polynomials, rather than straight lines, for the axes of the domain with encouraging results. Instead of each mode defining straight line motion for each point, the points follow polynomial curves as the parameter varies. Results will be presented in a further paper.

5.1.4. *Dealing with Small Numbers of Examples.* If there are fewer training examples, N, than point coordinates ($2n$), as is often the case, particularly for complex models, there can be no more than $N - 1$ degrees of freedom in the model. The principal component analysis required for the method uses the eigenvectors of the $2n \times 2n$ matrix $\mathbf{S}$ (Eqs. 9, 10). When $N < 2n$ this matrix has no more than $N - 1$ nonzero eigenvalues. Calculating all $2n$ eigenvectors in this case is unnecessary. An efficient way of calculating the eigenvectors associated with nonzero eigenvalues is given in Appendix B.

5.1.5. *Extensions to the Model.* Rather than have one "flat" PDM representing a complex assembly, it is possible to build a hierarchical PDM in which the top layer controls the position, scale, orientation, and shape parameters of the layer below. The bottom layer can consist of a number of subcomponents, each represented by a "flat" PDM. Varying the parameters of the top layer varies the pose, scale, and shape of the various components below. This avoids problems with the PDM due to rotating subcomponents—their orientation relative to the rest of the assembly can be modeled explicitly, rather than implicitly in a single-layer linear PDM.

It is also easy to extend the Point Distribution Model to deal with three dimensional data, for example, 3D medical images. We have recently described a successful system for automated interpretation of 3D Magnetic Resonance images of the brain using a 3D PDM [35].

5.1.6. *The Chord Length Distribution.* Elsewhere we have described how to derive a shape model from a training set using the distances between pairs of points—a Chord Length Distribution Model [31]. The distance, R_{ij}, between every pair of points i, j in each example of the training set is calculated, and the way these chord lengths vary is modeled by calculating their mean and covariances and applying a Principal Component Analysis. A model with several parameters is obtained, which returns sets of interpoint distances, R_{ij}, from which a new shape can be constructed. Varying the parameters varies the distances, which causes the shape to change. Such a system is able to model the rigid parts of an object regardless of their orientation, since it relies only on internal distances. Though this technique is sometimes better than the linear PDM at representing objects which can bend (such as

the "worms"), the reconstruction of the shape from the distances between points is iterative and slow. A refinement technique using such a model would be complex, leading us to favour the PDM for practical applications.

5.1.7. *Use as a Classifier.* A PDM 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 labeled points are placed on the example and the point set is aligned with the mean shape, we can calculate the model parameters required to generate the example. The distributions of the parameters can be estimated from the training set, allowing probabilities to be assigned. Alternatively, classification of unknown objects in images can be made by training a model for each class to be considered. Given a new image, the ASM technique is used to fit each model to the image data. The one which gives the best fit to the image is chosen as the result [36].

5.2. *Using PDMs in Image Search*

We have shown that ASMs are effective at locating known objects in images given initial estimates of position, scale, and orientation. How good an estimate is required will depend on how cluttered the image is and how well the model describes the objects in the image. For instance, unless some points in the model are close enough to the image example for their search profiles to overlap appropriate edges on the target object, the model instance cannot "see" the target, and will not be able to move towards it. We have found that techniques using Genetic Algorithms [26, 7, 27] provide a suitably close initial estimate to achieve successful refinement with ASMs.

5.2.1. *Occlusion and Clutter.* The hand example demonstrates that ASMs can deal successfully with occlusion and clutter. The heart example also shows that the method works with noisy images and missing data. As noise, clutter, and occlusion increase it becomes more likely that the models will latch onto the wrong edges, although the constraints applied by the PDM ensure that the shape of the final result is "sensible," even when it does not locate all the edges correctly. This is most likely to happen when the clutter has a similar structure to that of the objects being modeled and is thus most likely to be mistaken for some part of the object. Because the models we use are specific, we argue that we are applying the strongest possible shape-based constraints. In extreme cases, where the method fails, the fault lies in our approach to incorporating the image evidence—this is discussed below.

5.2.2. *Updating the Model Parameters.* How suggested adjustments are found for each point is important. Calculating the suggested movement by looking for strong nearby edges is simple and has proved effective in many cases. However, to search for more complex objects, where the model points do not necessarily lie on strong edges, more sophisticated algorithms are required. Potential maps can be derived, describing the likelihood for each point in an image that it is a particular model point. During search each model point attempts to move to more likely locations, climbing hills in the potential map. Alternatively, a model of the expected gray levels around each model point can be generated from the training examples; during image search each point is then moved toward the nearby area which best matches its local gray level model. Experiments using the latter technique have shown that it is more flexible and less sensitive to noise and clutter than simply searching for strong edges [32, 34].

By allowing the model to deform, but only in ways seen in the training set, we have a powerful technique for refinement. The constraints on the shape of the model are applied by the limits on the shape parameters. The $2n - t$ unrepresented modes of variation effectively have limits of zero on their parameters. Rather than fixed limits being used to enforce shape constraints, restoring forces in the parameter space could be applied, pulling the parameters back toward zero against the external "forces" from the image;

$$\mathbf{b} \rightarrow \mathbf{b} + \mathbf{W}_b d\mathbf{b} - k_b \mathbf{W}_b \mathbf{b} \quad (0 < k_b < 1). \tag{28}$$

This would give more weight to solutions closer to the mean shape, and require strong evidence for shapes which were considerably deformed. However, since this would be likely to lead to compromise solutions between image data and model we favor the fixed limit approach.

5.2.3. *Comparison with Other Work.* The work we present here can be thought of as a two dimensional application of Lowe's refinement technique [37]. Because of the linear nature of the Point Distribution Model, the mathematics is considerably simpler and leads to rapid execution.

Our Active Shape Models are superficially similar to Active Contour Models (Snakes) and Finite Element Models (FEMs). Each have their advantages. Snakes and FEMs can be created relatively easily and have the ability to locate partially occluded objects in noisy, cluttered scenes. The models are not, however, very specific, so under difficult conditions they can generate implausible interpretations. ASMs are harder to create because they require the user to annotate each of a set of training images with the correct interpretation. However, they model the allowed variability more specifically and are thus more robust to noise, clutter, and occlusion. A detailed experimental comparison between ASM and FEM methods is beyond the scope of this paper, but we hope to present results in the near future. We are also working actively

on methods which incorporate the advantages of both approaches.

5.2.4. *A Framework for Object Modeling and Recognition.* We have conducted experiments which suggest that our local optimization method can be fruitfully used in conjunction with a Genetic Algorithm (GA) search [26–28]. The GA can be run as a cue generator to produce a number of object hypotheses, which can be refined using the Active Shape Model. Alternatively, the ASM can be combined with the GA search, applying one iteration at each generation of the Genetic Algorithm. Both techniques have been used successfully to locate complex structures in a variety of images.

6. CONCLUSIONS

We have described Point Distribution Models (PDMs)—statistical models of shape which can be constructed from training sets of correctly labeled images. A PDM represents an object as a set of labeled points, giving their mean positions and a small set of modes of variation which describe how the object's shape can change. Applying limits to the parameters of the model enforces global shape constraints ensuring that any new examples generated are similar to those in the training set. Given a set of shape parameters, an instance of the model can be calculated rapidly. The models are compact and are well suited to generate-and-test image search strategies.

Active Shape Models (ASMs) exploit the linear formulation of PDMs in an iterative search procedure capable of rapidly locating the modeled structures in noisy, cluttered images—even if they are partially occluded. Object identification and location are robust because the models are specific in the sense that instances are constrained to be similar to those in the training set.

We have demonstrated the ability to create compact models of resistors, hearts (in echocardiograms), and hands. We have also shown that these models can be used successfully in image search. Using a conventional workstation a good interpretation can typically be obtained in seconds. We have described elsewhere various other applications in which the same methods have been exploited successfully, including examples where very complex structures (e.g., faces and automobile brake assemblies) are modeled. The important point to stress is that precisely the same software can be applied to a broad range of image interpretation problems—both medical and industrial—specialized only by training with suitable examples.

We believe that this approach holds considerable promise as a practical but generic technique for automated image interpretation.

APPENDIX A: ALIGNING A PAIR OF SHAPES

Given two similar shapes, $\mathbf{x}_1$ and $\mathbf{x}_2$, we would like to choose a rotation, θ, a scale, s, and a translation, (t_x, t_y), mapping $\mathbf{x}_2$ onto $M(\mathbf{x}_2) + \mathbf{t}$ so as to minimize the weighted sum

$$E = (\mathbf{x}_1 - M(s, \theta)[\mathbf{x}_2] - \mathbf{t})^{\mathrm{T}}\mathbf{W}(\mathbf{x}_1 - M(s, \theta)[\mathbf{x}_2] - \mathbf{t}), \quad (3)$$

where

$$M(s, \theta)\begin{bmatrix} x_{jk} \\ y_{jk} \end{bmatrix} = \begin{pmatrix} (s\cos\theta)x_{jk} - (s\sin\theta)y_{jk} \\ (s\sin\theta)x_{jk} + (s\cos\theta)y_{jk} \end{pmatrix}, \quad (4)$$

$$\mathbf{t} = (t_x, t_y, \ldots, t_x, t_y)^{\mathrm{T}}, \quad (29)$$

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

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

a 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,

$$\begin{pmatrix} X_2 & -Y_2 & W & 0 \\ Y_2 & X_2 & 0 & W \\ Z & 0 & X_2 & Y_2 \\ 0 & Z & -Y_2 & X_2 \end{pmatrix}\begin{pmatrix} a_x \\ a_y \\ t_x \\ t_y \end{pmatrix} = \begin{pmatrix} X_1 \\ Y_1 \\ C_1 \\ C_2 \end{pmatrix}, \quad (30)$$

where

$$X_i = \sum_{k=0}^{n-1} w_k x_{ik} \quad Y_i = \sum_{k=0}^{n-1} w_k y_{ik} \quad (31)$$

$$Z = \sum_{k=0}^{n-1} w_k(x_{2k}^2 + y_{2k}^2) \quad W = \sum_{k=0}^{n-1} w_k \quad (32)$$

$$C_1 = \sum_{k=0}^{n-1} w_k(x_{1k}x_{2k} + y_{1k}y_{2k}) \quad (33)$$

$$C_2 = \sum_{k=0}^{n-1} w_k(y_{1k}x_{2k} - x_{1k}y_{2k}). \quad (34)$$

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

APPENDIX B: CALCULATING THE EIGENVECTORS OF THE COVARIANCE MATRIX WHEN THERE ARE FEWER SAMPLES THAN CO-ORDINATES

When there are fewer training examples, N, than point co-ordinates, $2n$, the eigenvectors of the $2n \times 2n$ covariance matrix $\mathbf{S}$ can be calculated from the eigenvectors of

a smaller $N \times N$ matrix derived from the same data. Because the eigenvector calculation time goes as the cube of the size of the matrix, this can give substantial time savings.

We have N examples, $\mathbf{x}_i$ $(i = 1, \ldots, N_s)$. Let $\mathbf{D}$ be a $2n \times N$ matrix with these as its columns;

$$\mathbf{D} = (\mathbf{x}_1\ \mathbf{x}_2\ \ldots\ \mathbf{x}_N). \tag{35}$$

We can write the covariance matrix, $\mathbf{S}$, as

$$\mathbf{S} = \frac{1}{N}\mathbf{D}\mathbf{D}^T. \tag{36}$$

Let $\mathbf{T}$ be the $N \times N$ matrix

$$\mathbf{T} = \frac{1}{N}\mathbf{D}^T\mathbf{D}, \tag{37}$$

and let $\mathbf{e}_i$ $(i = 1, \ldots, N)$ be the unit, orthogonal eigenvectors of $\mathbf{T}$ with corresponding eigenvalues γ_i:

$$\mathbf{T}\mathbf{e}_i = \gamma_i\mathbf{e}_i \quad (i = 1, \ldots, N). \tag{38}$$

Then

$$\frac{1}{N}\mathbf{D}^T\mathbf{D}\mathbf{e}_i = \gamma_i\mathbf{e}_i. \tag{39}$$

Premultiplying by $\mathbf{D}$,

$$\frac{1}{N}\mathbf{D}\mathbf{D}^T\mathbf{D}\mathbf{e}_i = \gamma_i\mathbf{D}\mathbf{e}_i \tag{40}$$

$$\mathbf{S}(\mathbf{D}\mathbf{e}_i) = \gamma_i(\mathbf{D}\mathbf{e}_i). \tag{41}$$

Thus if $\mathbf{e}_i$ is an eigenvector of $\mathbf{T}$, then $\mathbf{D}\mathbf{e}_i$ is an eigenvector of $\mathbf{S}$ and has the same eigenvalue. The N unit orthogonal eigenvectors of $\mathbf{S}$ are then $\mathbf{p}_i$ $(i = 1, \ldots, N)$, where

$$\mathbf{p}_i = \frac{1}{\sqrt{\gamma_i N}}\mathbf{D}\mathbf{e}_i \tag{42}$$

with corresponding eigenvalues $\lambda_i = \gamma_i$. The scaling factor in Eq. 42 is required to give the eigenvectors unit length. Mutual orthogonality is easily shown:

$$\begin{aligned}\mathbf{p}_i^T\mathbf{p}_j &= \frac{1}{\gamma_i N}\mathbf{e}_i^T\mathbf{D}^T\mathbf{D}\mathbf{e}_j = \frac{1}{\gamma_i}\mathbf{e}_i^T\mathbf{T}\mathbf{e}_j \\ &= \mathbf{e}_i^T\mathbf{e}_j \quad \begin{matrix}= 1 & (i = j) \\ = 0 & (i \neq j)\end{matrix}.\end{aligned} \tag{43}$$

ACKNOWLEDGMENTS

This work was funded by the Science and Engineering Research Council under the Information Engineering Advanced Technology Programme (Project Number 3/2114). Tim Cootes is currently funded by an SERC Post-Doctoral Fellowship. The authors thank the other members of the Wolfson Image Analysis Unit particularly D. Bailes and A. Hill, for their help and advice, and the anonymous reviewers for their suggestions.

REFERENCES

1. R. T. Chin and C. R. Dyer, Model-based recognition in robot vision, *Comput. Surv.* **18,** 1986, 67–108.
2. W. E. L. Grimson, *Object Recognition by Computer: The Role of Geometric Constraints*, MIT Press, Cambridge, MA, 1990.
3. M. Kass, A. Witkin, and D. Terzopoulos, Snakes: Active contour models, in *Proceedings, First International Conference on Computer Vision*, pp. 259–268, IEEE Comput. Soc. Press, 1987.
4. G. E. Hinton, C. K. I. Williams, and M. D. Revow, Adaptive elastic models for hand-printed character recognition, in *Advances in Neural Information Processing Systems 4* (J. E. Moody, S. J. Hanson, and R. P. Lippmann, Eds.), Morgan Kauffmann, San Mateo, CA, 1992.
5. A. L. Yuille, D. S. Cohen, and P. Hallinan, Feature extraction from faces using deformable templates, *Int. J. Comput. Vision* **8,** 1992, 99–112.
6. P. Lipson, A. L. Yuille, D. O'Keeffe, J. Cavanaugh, J. Taaffe, and D. Rosenthal, Deformable templates for feature extraction from medical images, in *Proceedings of the First European Conference on Computer Vision* (O. Faugeras, Ed.), Lecture Notes in Computer Science, pp. 413–417, Springer–Verlag, Berlin/New York, 1990.
7. A. Hill and C. J. Taylor, Model based image interpretation using genetic algorithms, *Image Vision Comput.* **10,** 1992, 295–300.
8. A. Beinglass and H. J. Wolfson, Articulated object recognition, or: How to generalize the generalized Hough transform in *Proceedings, IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1991*, pp. 461–466.
9. G. L. Scott, The alternative Snake—And other animals, in *Proceedings, 3rd Alvey Vision Conference, Cambridge, 1987*, pp. 341–347.
10. L. H. Staib and J. S. Duncan, Parametrically deformable contour models, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, 1989*, pp. 427–430.
11. H. I. Bozma and J. S. Duncan, Model-based recognition of multiple deformable objects using a game-theoretic framework, in *Information Processing in Medical Imaging—Proceedings of the 12th International Conference*, pp. 358–372, Springer-Verlag, Berlin/New York, 1991.
12. U. Grenander, Y. Chow, and D. M. Keenan, *Hands. A Pattern Theoretic Study of Biological Shapes*, Springer-Verlag, New York, 1991.
13. U. Grenander and M. I. Miller, Representations of knowledge in complex systems. *J. R. Stat. Soc. B*, in press.
14. C. Goodall, Procrustes methods in the statistical analysis of shape (with discussions), *J. R. Stat. Soc. B.* **53,** 1991, 285–339.
15. K. V. Mardia, J. T. Kent, and A. N. Walder, Statistical shape models in image analysis, in *Proceedings of the 23rd Symposium on the Interface, Seattle 1991*, pp. 550–557.
16. F. L. Bookstein, *Morphometric Tools for Landmark Data*, Cambridge Univ. Press, London/New York, 1991.
17. F. L. Bookstein, Principal warps: Thin-plate splines and the decomposition of deformations, IEEE Trans. Pattern Anal. Mach. Intell. **11,** 1989, 567–585.
18. A. Pentland, Automatic extraction of deformable part models, *Int. J. Comput. Vision* **13,** No. 2, 1990, 107–126.

19. A. Pentland and S. Sclaroff, Closed-form solutions for physically based modeling and recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* **13,** 1991, 715–729.
20. D. Terzopoulos and D. Metaxas, Dynamic 3D models with local and global deformations: Deformable superquadrics, *IEEE Trans. Pattern Anal. Mach. Intell.* **13,** 1991, 703–714.
21. C. Nastar and N. Ayache, Fast segmentation, tracking and analysis of deformable objects, in *Proceedings, International Conference on Computer Vision, 1993,* pp. 275–279, IEEE Comput. Soc. Press, 1993.
22. P. Karaolani, G. D. Sullivan, K. D. Baker, and M. J. Baines, A finite element method for deformable models, in *Proceedings of the Fifth Alvey Vision Conference, Reading, 1989,* pp. 73–78.
23. P. Karaolani, G. D. Sullivan, and K. D. Baker, Active contours using finite elements to control local scale, in *Proceedings, British Machine Vision Conference 1992,* pp. 481–487, Springer-Verlag, Berlin/New York, 1992.
24. J. C. Gower, Generalized Procrustes analysis, *Psychometrika* **40,** 1975, 33–51.
25. R. A. Johnson and D. W. Wichern, *Multivariate Statistics, A Practical Approach,* Chapman & Hall, London/New York, 1988.
26. A. Hill, T. F. Cootes, and C. J. Taylor, A genetic system for image interpretation using flexible templates, in *British Machine Vision Conference,* Springer-Verlag, 1992.
27. A. Hill, C. J. Taylor, and T. Cootes, Object recognition by flexible template matching using genetic algorithms, in *Proceedings, European Conference on Computer Vision* (G. Sandini, Ed.), pp. 852–856, Springer-Verlag, Berlin/New York, 1992.
28. A. Hill, A. Thornham, and C. J. Taylor, Model-based interpretation of 3D medical images, in *Proceedings, British Machine Vision Conference 1993* (J. Illingworth, Ed.), Vol. 2, pp. 339–348, BMVA Press, 1993.
29. D. J. Burr, A dynamic model for image registration, *Comput. Graphics Image Process.* **15,** 1981, pp. 102–112.
30. S. Sclaroff and A. Pentland, A model framework for correspondence and description, in *Proceedings, International Conference on Computer Vision, 1993,* pp. 715–729, IEEE Comput. Soc. Press, 1993.
31. T. F. Cootes, D. H. Cooper, C. J. Taylor, and J. Graham, A trainable method of parametric shape description, *Image Vision Comput.* **10,** 1992, 289–294.
32. T. F. Cootes and C. J. Taylor, Active shape model search using local grey-level models: A quantitative evaluation, in *Proceedings, British Machine Vision Conference 1993* (J. Illingworth, Ed.), Vol. 2, pp. 639–648, BMVA Press, 1993.
33. T. F. Cootes, A. Hill, C. J. Taylor, and J. Haslam, The use of active shape models for locating structures in medical images, in *Proceedings, Information Processing for Medical Imaging* (H. H. Barrett and A. F. Gmitro, Ed.), pp. 33–47, Springer-Verlag, Berlin/New York, 1993.
34. T. F. Cootes, C. J. Taylor, A. Lanitis, D. H. Cooper, and J. Graham, Building and using flexible models incorporating grey-level information, in *Proceedings, International Conference on Computer Vision,* pp. 242–246, IEEE Comput. Soc. Press, 1993.
35. A. Hill, A. Thornham, and C. J. Taylor, Model-based interpretation of 3D medical images, in *Proceedings, British Machine Vision Conference, 1993* (J. Illingworth, Ed.), Vol. 1, pp. 339–348, BMVA Press, 1993.
36. A. Lanitis, C. J. Taylor, and T. F. Cootes, A generic system for classifying variable objects using flexible template matching, in *Proceedings, British Machine Vision Conference, 1993* (J. Illingworth, Ed.), Vol. 1, pp. 329–338, BMVA Press, 1993.
37. D. G. Lowe, Fitting parameterized three-dimensional models to images, *IEEE Trans. Pattern Anal. Mach. Intell.* **13,** 1991, 441–450.
38. J. A. Grogan, *Automated Analysis of Pedobarograph Images,* M. Sc. Thesis, Victoria University of Manchester, Oct. 1993.