A Framework for Automatic Landmark Identification Using a New Method of Nonrigid Correspondence

Andrew Hill, Chris J. Taylor, and Alan D. Brett

Abstract—A framework for automatic landmark indentification is presented based on an algorithm for corresponding the boundaries of two shapes. The auto-landmarking framework employs a binary tree of corresponded pairs of shapes to generate landmarks automatically on each of a set of example shapes. The landmarks are used to train statistical shape models known as Point Distribution Models. The correspondence algorithm locates a matching pair of sparse polygonal approximations, one for each of a pair of boundaries by minimizing a cost function, using a greedy algorithm. The cost function expresses the dissimilarity in both the shape and representation error (with respect to the defining boundary) of the sparse polygons. Results are presented for three classes of shape which exhibit various types of nonrigid deformation.

Index Terms—Correspondence, critical points, polygonal approximation, automatic landmarks, flexible templates, point distribution models.

1 Introduction

In this paper, we describe a framework for generating *landmarks* automatically on each of a training set of example shapes for a class of objects. By a landmark, we mean a point which identifies a salient feature on an object and which is present on every example of the class. The example shapes of the class are therefore similar in that they can all be described by a set of landmarks which are homologous for each example in the class. Here, the landmarks are used to train a statistical flexible template known as a Point Distribution Model (PDM) [1]. This avoids the time-consuming and subjective process of identifying the landmark points manually. The framework generates landmarks via a binary tree of merged pairs of shapes. The algorithm for generating the tree relies upon the ability both to match pairs of shapes (in order for them to be merged) and to measure the quality of the match (in order to decide which pairs to merge). In order for this process to be successful, an accurate, robust method of pairwise correspondence is required.

Pairwise correspondence is a well-studied problem in computer vision. It requires the identification of the transformation which maps the boundary of one object onto that of another. In some cases, the transformation sought is known to be Euclidean, i.e., a *rigid* transformation. We are interested, however, in the case where the two boundaries represent different examples from the same class of objects (e.g., two hands) and a nonrigid transformation is required to map one boundary onto the other. To this end we present a novel method of pairwise correspondence which has proven to be both accurate and robust. The algorithm locates a pair of matching sparse polygonal approximations, one for each of a pair of boundaries, by minimizing a cost function using a greedy algorithm. The cost function expresses the dissimilarity in both the shape and representation error (with respect to the defining boundary) of the sparse polygons. The algorithm requires a single control parameter, associated with the cost function, the value of which is not critical.

We present results for three different classes of objects—hands, left ventricles of the heart, and resistors from printed circuit boards. Qualitative results of applying the correspondence algorithm to isolated pairs of shapes are presented. We also present quantitative results which demonstrate how the automatic landmark generation framework, when used in conjunction with the polygon-based correspondence algorithm, produces landmarks similar to those generated manually for training sets of example shapes.

2 Background

The motivation for the work presented here is to identify automatically, on each of a set of examples, a set of landmark points. Once corresponding points have been identified on each example, a statistical analysis of the shape of the object is possible. Bookstein [2] uses landmarks for morphometric analysis of biological data, while Goodall [3] discusses the registration of shapes and the use of Procrustes analysis for estimating the mean and covariances between landmarks and the differences between sets of shapes. In our case, a statistical model of the shape of the object is generated—a PDM.

Training a PDM from a set of N examples, each represented by a set of L landmarks $\{\mathbf{y}_{ij}; (1 \leq i \leq N), (1 \leq j \leq L)\}$ involves: Aligning the set of examples into a common frame

- A. Hill is with Kestra Ltd., Kestra House, Broughton Hall, Skipton, North Yorkshire BD23 3AE, UK. E-mail: andrew@kestra.com.
- C.J. Taylor and A.D. Brett are with Division of Imaging Science & Biomedical Engineering, University of Manchester, Oxford Road, Manchester M13 9PT, UK. E-mail: {c.taylor, a.brett}@man.ac.uk.

Manuscript received 19 Jul. 1996; accepted 19 Apr. 1999.
Recommended for acceptance by P. Flynn.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 107772.

of reference [4], $\mathbf{x}_i = \mathrm{aligned}(\mathbf{y}_i)$; calculating the mean of the aligned examples, $\overline{\mathbf{x}}$, and the deviation from the mean of each aligned example $\delta\mathbf{x}_i = \mathbf{x}_i - \overline{\mathbf{x}}$; and calculating the eigensystem of the the covariance matrix of the deviations, $\mathbf{C} = (1/N)\sum_{i=1}^{N}(\delta\mathbf{x}_i)(\delta\mathbf{x}_i)^T$. The t principal eigenvectors of the eigensystem are used to generate examples of the modeled object via the expression $\mathbf{x} = \overline{\mathbf{x}} + \mathbf{Pb}$, where $\mathbf{b}$ is a t-element vector of shape parameters and $\mathbf{P}$ is a $(2N \times t)$ in 2D or $(3N \times t)$ in 3D matrix of t eigenvectors. PDMs have been used to model many classes of variable objects including: faces, hands, structures in the brain and heart, vertebrae, livestock, components on printed circuit boards, and walking people [4], [5], [6], [7], [8]. The use of PDMs to automatically identify examples of the modeled object(s) in unseen images and the relationship of PDMs with other forms of flexible templates has been presented elsewhere [4].

While much attention has focused on how one might apply statistical techniques to landmark data once generated, few suggestions have been made as to how the landmarks might be acquired automatically or semiautomatically. Baumberg and Hogg [8] describe a system which generates landmarks automatically for outlines of walking people. The outlines are represented as pixellated boundaries extracted automatically from a sequence of images using motion analysis. Landmarks are generated on an individual basis for each boundary by computing the principal axis of the boundary, identifying a reference pixel on the boundary at which the principal axis intersects the boundary, and generating a number of equally spaced points from the reference point with respect to the path length of the boundary. While this process is satisfactory for silhouettes of pedestrians, it is unlikely that it will be generally successful (consider the hand example used here). Even for shapes which are amenable to this approach, the localization of the landmarks will be poor. In recognition of this, Baumberg and Hogg [9] describe how the position of the landmarks can be iteratively updated in order to generate improved shape models generated from the landmarks.

A method of repositioning the landmarks in order to generate better shape models has also been described by the authors [10]. What we are concerned with here, however, is generating an initial estimate of the landmarks in a manner which is robust and accurate. This reduces or removes the need to reparameterize the boundaries in order to generate better landmark positions.

Several methods of shape registration in 3D have been applied to the problem of building statistical models by producing point correspondences across a training set. Joshi et al. [11] deform a template onto hippocampal surface representations using the registration method of Christensen et al. [12]. This nonrigid registration uses a course linear elastic matching of volumes followed by refinement by a viscous fluid transform. However, this technique is computationally expensive, requiring a massively parallel computer to solve the partial differential equations of the fluid model. Fleute and Lavalée [13] use a framework of initially matching each training example to a single template, building a mean from these matched examples, and then iteratively matching each example to the current mean and repeating until convergence. Matching is performed using the multiresolution registration method of Szeliski and Lavalée [14]. This method deforms the volume of space embedding the surface rather than deforming the surface itself. Kelemen et al. [15] parameterize the surfaces of each of their shape examples using the method of Brechbühler et al. [16]. Correspondence may then be established between surfaces, but relies upon the choice of a parametric origin on each surface mapping and registration of the coordinate systems of these mappings by the computation of a rotation.

Recently, Kotcheff and Taylor [17] have described a method of constructing statistical template models automatically by direct optimization. The authors solve the optimization problem using a genetic algorithm. The models that this method produces are comparable to, and often better than, hand-built versions. However, this very large, nonlinear optimization problem is extremely computationally expensive.

3 Auto-Landmark Framework

Let us assume that the object we wish to model can be represented by a closed boundary within an image—we discuss the limitations imposed by this assumption later. The training set consists of a set of images, each containing one or more examples of the objects, which can appear with varying orientation, position, scale, and shape. In order to produce a set of landmarks on the boundary of each example, we must first be able to identify corresponding points on different examples of the object.

Let us assume that a method of establishing correspondence exists which can take a pair of object boundaries and generate a pixel-to-pixel mapping. Further assume that a metric Ω exists which describes how well the two shapes correspond. The details of such a metric will be presented in Section 5. We can use this pairwise corresponder to generate the mean shape for the set of examples using the following algorithm:

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

This algorithm can be represented as a binary tree, as shown in Fig. 1. The time complexity of building such a tree

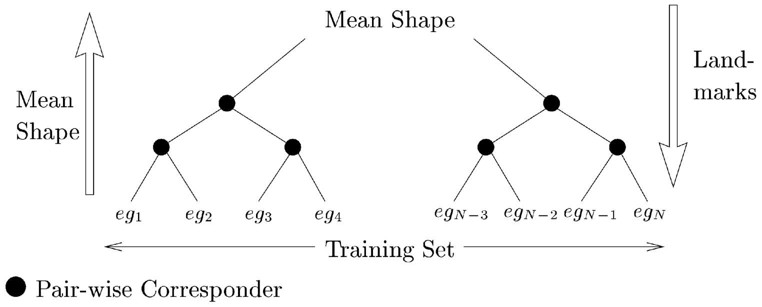


Fig. 1. Generating the mean shape and approximate landmarks.

is $O(N^2)$, where N is the number of training examples. The leaves of the tree are the individual members of the training set and the mean shape is the root. By choosing pairs of examples from the training set which have the lowest match cost, we use the pairwise corresponder on the most similar shapes in the training set. This improves the robustness of the landmarking framework. A set of landmarks can be generated automatically on the mean shape. Any sensible algorithm may be used for this purpose. We use a Critical Point Detection algorithm [18] to choose a set of points which best represent the mean shape as the vertices of a sparse polygon. This algorithm is described later in Section 5.1.

Once the landmarks have been placed on the mean shape, they are projected back along the branches of the tree towards the leaves. This is accomplished using the pixel correspondences used to generate the tree. Each parent node in the binary tree has two children. The pixel correspondences used to generate the (mean) parent shape for a given pair of children are used to identify the pixels on each of the children corresponding to the landmarks of the parent. This procedure is applied from the root to the leaves to generate the landmarks for each member of the training set. Clearly, this algorithm requires a reliable method of pairwise correspondence. We now briefly review previous methods of nonrigid correspondence before describing our new polygon-based correspondence algorithm.

4 Background to Nonrigid Correspondence

Duncan et al. [19], Kambhamettu and Goldgof [20], and Cohen et al. [21] all propose methods of correspondence based on the minimization of a cost function which involves the difference in the curvature of two boundaries (or surfaces). As pointed out by Tagare et al. [22], however, curvature is a rigid invariant of shape and its applicability to nonrigid correspondence is problematic. Tagare et al. [22] recently proposed a method of correspondence based on the minimization of a symmetric cost function which measures the difference between a geometric criterion, so-called *sphericity*, of the two boundaries. The cost function involves the computation of the curvature of the boundary but does not compare the curvature directly. The optimization scheme employed by Tagare to minimize the cost function requires five control parameters which may make the method difficult to routinely use as part of an automatic system.

The related methods of Scott and Longuet-Higgins [23], Shapiro and Brady [24], and Sclaroff and Pentland [25] describe methods of correspondence between two sets of points, the connectivity of which is not specified. The first two of these methods are better suited to the determination of the correspondences arising from a rigid transformation of one pointset onto the other. The method of Sclaroff and Pentland [25] is proposed for nonrigid correspondence of pixellated boundaries. The algorithm first constructs a finite element model (FEM) of each of the two pointsets. Modal analysis of the FEMs produces a set of *physical modes of variation* for each pointset. Correspondences are produced by matching the two sets of modes directly, following the approach of Shapiro and Brady. We have implemented this algorithm and found it unsuitable for our purposes. The reasons for this are twofold: First, in order to build an FEM, it is necessary to construct the Galerkin interpolation matrix, which is the inverse of the matrix $\mathbf{G} = \left[g_i(\mathbf{x}_j)\right](1 \leq i, j \leq m)$, where $g_i(\mathbf{x}_j) = e^{\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2}$ and $\mathbf{x}_i(1 \leq i \leq m)$ are the points in a given pointset. We have found that $\mathbf{G}$ is almost singular when points on the shape are close to one another or σ becomes large, a drawback which is not discussed in [25]. Second, as with the Scott and Shapiro methods, the algorithm is not guaranteed to generate a legal set of correspondences because the connectivity of the boundary is not enforced. By legal, it is meant that, as we travel around the boundaries of two coresponded shapes between successive point correspondences, the arc length on each boundary is nondecreasing. That is, the correspondence does not reverse direction or "fold" between the boundaries.

Rangarajan et al. [26], [27] describe a method of point matching which simultaneously determines a set of matches and the similarity transform parameters required to register two contours defined by dense point sets. The method is robust against point features on one contour that do not appear on the other. An optimization method similar to simulated annealing is used to solve the problem to produce a matrix of correspondences. In common with the methods described previously [23], [24], [25], the construction of the correspondence matrix cannot guarantee the production of a legal set of correspondences.

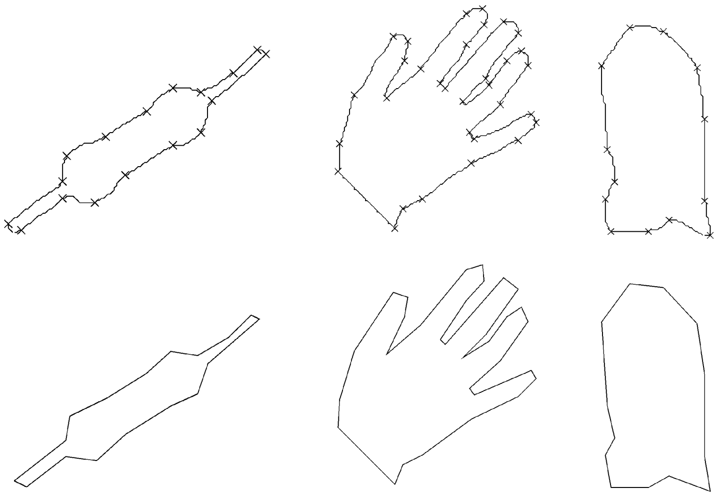


Fig. 2. Critical points and polygonal approximations for various shapes.

5 Polygon-Based Correspondence

In this section, we describe a new correspondence algorithm which transforms a given discretized boundary, $\mathbf{A} = \{\mathbf{A}_i; 1 \leq i \leq n_A\}$, onto some other boundary, $\mathbf{B} = \{\mathbf{B}_i; 1 \leq i \leq n_B\}$. We assume that each boundary has been normalized such that the center-of-gravity is at the origin and the mean distance of the points from the origin is 1. The output of the algorithm is a set of ordered pairs $\Phi = \{\phi_i = (\alpha_i, \beta_i); 1 \leq i \leq n_\Phi\}$. The integer values $\{\alpha_i\}$ index the pixels of $\mathbf{A}$ and the integer values $\{\beta_i\}$ index the pixels of $\mathbf{B}$. The shapes $\mathbf{A}' = \{\mathbf{A}'_i = \mathbf{A}_{\alpha_i}; 1 \leq i \leq n_\Phi\}$ and $\mathbf{B}' = \{\mathbf{B}'_i = \mathbf{B}_{\beta_i}; 1 \leq i \leq n_\Phi\}$ represent sparse subpolygons of $\mathbf{A}$ and $\mathbf{B}$, respectively. The values α_i, β_i satisfy the following conditions:

$$\begin{array}{l} 1 \leq \alpha_i \leq n_A \\ 1 \leq \beta_i \leq n_B \end{array} \tag{1}$$

$$\left.\begin{array}{l} \sum_{i=1}^{n_\Phi} \delta(\alpha_i, \alpha_{i+1}) = 1 \\ \\ \sum_{i=1}^{n_\Phi} \delta(\beta_i, \beta_{i+1}) = 1 \end{array}\right\} \delta(j,k) = \left\{\begin{array}{ll} 1 & \text{if } j \geq k \\ & \\ 0 & \text{if } j < k \end{array}\right., \tag{2}$$

where we have assumed, as we will in the remainder of the paper, the appropriate modulo arithmetic for boundaries, i.e., $\alpha_0 = \alpha_{n_\Phi}$, $\alpha_1 = \alpha_{n_\Phi+1}$ etc. Condition (1) ensures that all indices are in the correct range, while (2) ensures that the indices $\{\alpha_i\}$, $\{\beta_i\}$ form legal subpolygons of $\mathbf{A}$ and $\mathbf{B}$, respectively. The algorithm is entirely symmetric with respect to $\mathbf{A}$ and $\mathbf{B}$, i.e., the same result is obtained when $\mathbf{A}$ is matched to $\mathbf{B}$ as that obtained when $\mathbf{B}$ is matched to $\mathbf{A}$.

The correspondence algorithm comprises three parts:

1. Generation of sparse polygonal approximations to both $\mathbf{A}$ and $\mathbf{B}$, $\mathbf{A}''$ and $\mathbf{B}''$, respectively. $\mathbf{A}''$ and $\mathbf{B}''$ are simply sparse representations of $\mathbf{A}$ and $\mathbf{B}$—no correspondences are established at this stage and the polygons will usually be different sizes, i.e., $n_{\mathbf{A}''} \neq n_{\mathbf{B}''}$.
2. Generation of an initial estimate of the corresponding sparse polygons $\mathbf{A}'$ and $\mathbf{B}'$. This is accomplished using a correspondence algorithm based on the polygonal arc path-lengths of $\mathbf{A}$ and $\mathbf{B}$ utilizing $\mathbf{A}''$ and $\mathbf{B}''$ generated in (1). The number of ordered pairs, n_Φ, is fixed at this stage, $n_\Phi = (n_{\mathbf{A}''} + n_{\mathbf{B}''})/2$.
3. Refinement of the initial set of correspondences using a greedy optimization scheme which modifies the ordered pairs $\{\phi_i\}$ in order to generate sparse polygons $\mathbf{A}'$ and $\mathbf{B}'$ which are similar in shape to one another and have similar *representation errors* with respect to their defining boundaries, $\mathbf{A}$ and $\mathbf{B}$. The optimization scheme modifies either the α_i values or the β_i values, but not both, i.e., either the polygon $\mathbf{A}'$ is fixed and $\mathbf{B}'$ modified or vice versa.

Each of these steps will now be described in greater detail.

5.1 Sparse Polygon Generation

To generate a sparse polygon, $\mathbf{A}''$, representing $\mathbf{A}$ we have used the critical point detection (CPD) algorithm described by Zhu and Chirlian [18]. The CPD algorithm assigns a *critical value* to each point on the boundary which is simply the area of the triangle constructed from the given point and its two immediate neighbors. An iterative decimation process is used which removes the point with the smallest critical value, recomputes the critical value of the immediate neighbors of the point which has just been deleted, and reidentifies the point with smallest critical value. The process terminates when the remaining smallest critical value is above some threshold set by the user.

In order to have as few controlling parameters as possible, we have automated the selection of the threshold for a given boundary as follows: The monotonically decreasing curve of (the number of critical points) versus (threshold value) for threshold values [0..0.75] is generated. A straight line is drawn connecting the first and last points of the curve. The point in the curve which has a maximum distance from this line defines the threshold value. The result of applying this process to various shapes is shown in Fig. 2.

5.2 Path-Based Correspondence

In this section, we describe the method of generating an initial estimate of the corresponding polygons $\mathbf{A}'$ and $\mathbf{B}'$. We use the assumption that $\mathbf{A}$ and $\mathbf{B}$ are similar shapes to predict that the spacing of the points $\mathbf{A}_{\alpha_i}$ with respect to the polygonal arc path-length of $\mathbf{A}$ will be similar to the spacing of the corresponding points $\mathbf{B}_{\beta_i}$ with respect to the polygonal arc path-length of $\mathbf{B}$, i.e., if two points on $\mathbf{A}'$, $\mathbf{A}_{\alpha_i}$ and $\mathbf{A}_{\alpha_{i+1}}$, are separated by 5 percent of the total path-length of $\mathbf{A}$, then we expect the corresponding points on $\mathbf{B}'$, $\mathbf{B}_{\beta_i}$ and $\mathbf{B}_{\beta_{i+1}}$, to be separated by 5 percent of the total path-length of $\mathbf{B}$. The polygonal arc path-length between two points $\mathbf{A}_i$, $\mathbf{A}_j$ is defined by $\sum_{k=i}^{j-1} \|\mathbf{A}_{k+1} - \mathbf{A}_k\|$. If we require this arc path length to be always nondecreasing on both $\mathbf{A}$ and $\mathbf{B}$, the correspondences generated by this method will be legal.

The path matching correspondence algorithm exhaustively tests every pixel $\mathbf{B}_i (1 \leq i \leq n_B)$ in the following manner: A reference point of correspondence, $\phi_0 = (\alpha_0 = 1, \beta_0 = i)$ is generated. A set $\Phi = \{\phi_i; 1 \leq i \leq (n_{\mathbf{A}''} + n_{\mathbf{B}''})\}$ of correspondences is determined by projecting the path-length spacing of the points defining $\mathbf{A}''$ (with respect to $\mathbf{A_1}$) onto $\mathbf{B}$ (with respect to $\mathbf{B}_i$) and also projecting the path-length spacing of the points defining $\mathbf{B}''$ (with respect to $\mathbf{B}_i$) onto $\mathbf{A}$ (with respect to $\mathbf{A}_1$). For example, if $\mathbf{A}_j''$ lies 5 percent of the total path-length of $\mathbf{A}$ from $\mathbf{A}_1$, its corresponding point lies 5 percent of the total path-length of $\mathbf{B}$ from $\mathbf{B}_i$. The projection is in both directions, i.e., $\mathbf{A}''$ onto $\mathbf{B}$ and $\mathbf{B}''$ onto $\mathbf{A}$, to obtain a symmetric result. The set of correspondences is arranged such that conditions (1) and (2) are satisfied and the pose, Q, which satisfies:

$$\text{Min } E_i^2 = \sum_{j=1}^{n_{\mathbf{A}''}+n_{\mathbf{B}''}} \|\mathbf{A}_{\alpha_j} - Q(\mathbf{B}_{\beta_j})\|^2 \tag{3}$$

is determined, where Q represents the Euclidean transformation $Q(\mathbf{p}) = s\mathbf{R}\mathbf{p} + \mathbf{t}$, s is a scale factor, $\mathbf{R}$ is a rotation matrix, and $\mathbf{t}$ is a translation. See for example, Umeyama [28] for a solution to (3) and Horn [29] for the modification required to make the determination of Q symmetric.

Now, the index i for which $E_i^2 \leq E_j^2 \, \forall j$ identifies the pixel $\mathbf{B}_i$ which matches the pixel $\mathbf{A}_1$ and results in the best correspondence (in a Euclidean sense) when the spacing of the points which define $\mathbf{B}''$ is replicated on $\mathbf{A}$ and the spacing of the points which define $\mathbf{A}''$ is replicated on $\mathbf{B}$. Note that this path-matching algorithm can recover exactly any Euclidean transformation of $\mathbf{A}$, i.e., $\mathbf{B} = Q(\mathbf{A})$, the normalized polygonal arc path-length of a boundary being invariant under a Euclidean transformation. This procedure is not entirely symmetric because there may be a different number of unequally spaced pixels on $\mathbf{A}$ and $\mathbf{B}$. For a perfectly symmetric result, we consider a number, $(n_\mathbf{A} + n_\mathbf{B})/2$, of equally spaced "virtual" pixels on $\mathbf{A}$ and $\mathbf{B}$, starting with $\mathbf{A}_1$ and $\mathbf{B}_1$, respectively. These virtual pixels are placed on the original polygons of $\mathbf{A}$ and $\mathbf{B}$ that were defined by the pixels $\mathbf{A}_i$ and $\mathbf{B}_i$. However, these virtual pixels are spaced at intervals of $2L(\mathbf{A})/(n_\mathbf{A} + n_\mathbf{B})$ and $2L(\mathbf{B})/(n_\mathbf{A} + n_\mathbf{B})$, respectively, where $L(\mathbf{A})$ is the total polygonal arc path length of $\mathbf{A}$:

$$L(\mathbf{A}) = \sum_{k=1}^{n_\mathbf{A}} \|\mathbf{A}_{(k+1) \bmod n_\mathbf{A}} - \mathbf{A}_k\|. \tag{4}$$

Likewise, $L(\mathbf{B})$ is the total polygonal arc path length of $\mathbf{B}$. Exhaustive testing with respect to these equally spaced pixels gives the required symmetric result.

This path-matching algorithm produces a set of correspondences $\Phi = \{\phi_i; 1 \leq i \leq (n_{\mathbf{A}''} + n_{\mathbf{B}''})\}$ derived from the critical points located on $\mathbf{A}$ and $\mathbf{B}$. It is quite possible that a subset of the critical points $\mathbf{A}''$ correspond directly with a subset of the critical points $\mathbf{B}''$. This represents redundant information which we eliminate using a decimation process similar to that used to generate the critical points. For each pair of corresponding points $(\mathbf{A}_{\alpha_i}, \mathbf{B}_{\beta_i})$, we compute:

$$T_i = \text{Max } (\text{Area}(\mathbf{A}_{\alpha_{i-1}}\mathbf{A}_{\alpha_i}\mathbf{A}_{\alpha_{i+1}}), \text{Area}(\mathbf{B}_{\beta_{i-1}}\mathbf{B}_{\beta_i}\mathbf{B}_{\beta_{i+1}})), \tag{5}$$

where $\text{Area}()$ computes the area of a given triangle. The correspondence, ϕ_i, for which T_i is a minimum is deleted repeatedly until the required number of correspondences is achieved, $n_\Phi = (n_{\mathbf{A}''} + n_{\mathbf{B}''})/2$.

5.3 Optimization Scheme

Given an initial set of correspondences generated by the path-based correspondence algorithm described in Section 5.2, an iterative local optimization scheme is used to modify the correspondences in order to minimize the following cost function:

$$E = \lambda E_S + (1 - \lambda) E_R. \tag{6}$$

E_S measures the difference in shape between the fixed polygon $\mathbf{A}'$ and its corresponding polygon $\mathbf{B}'$, while E_R measures the difference between the *representation errors* of $\mathbf{A}'$ and $\mathbf{B}'$. This provides a metric which describes how well the two shapes correspond.

The term E_S is defined as:

$$\left(\frac{L(\mathbf{A}') + L(\mathbf{B}')}{2n_\Phi}\right)\left(\frac{1}{n_\Phi}\sum_{i=1}^{n_\Phi} \begin{matrix} S(<{\mathbf{A}'}_{i-1}, {\mathbf{A}'}_i, {\mathbf{A}'}_{i+1}>, \\ <{\mathbf{B}'}_{i-1}, {\mathbf{B}'}_i, {\mathbf{B}'}_{i+1}>) \end{matrix}\right), \tag{7}$$

where $L(\mathbf{A}'), \mathbf{L}(\mathbf{B}')$ are the total polygonal arc path-lengths of the polygons $\mathbf{A}'$, $\mathbf{B}'$, respectively. The first bracketed term in (7) simply defines the mean length of the segments which make up the polygons $\mathbf{A}'$, $\mathbf{B}'$. The second term measures the difference in shape of the two polygons as the mean value of a local shape difference operator S. This operator takes two corresponding triplets

$$<{\mathbf{A}'}_{i-1}, {\mathbf{A}'}_i, {\mathbf{A}'}_{i+1}>, <{\mathbf{B}'}_{i-1}, {\mathbf{B}'}_i, {\mathbf{B}'}_{i+1}>$$

and measures both the difference in the angles ${\mathbf{A}'}_{i-1}{\mathbf{A}'}_i{\mathbf{A}'}_{i+1}$ and ${\mathbf{B}'}_{i-1}{\mathbf{B}'}_i{\mathbf{B}'}_{i+1}$ and the difference in the lengths of the segments $<{\mathbf{A}'}_i, {\mathbf{A}'}_{i+1}>$ and $<{\mathbf{B}'}_i, {\mathbf{B}'}_{i+1}>$—see Fig. 3. In reference to that figure, S is given by:

$$2\|\mathbf{c} - \mathbf{f}'\|/(\|\mathbf{c} - \mathbf{b}\| + \|\mathbf{f} - \mathbf{e}\|). \tag{8}$$

The value returned by S is the difference in shape expressed as a relative proportion of the mean length of the segments $<\mathbf{A}_i', \mathbf{A}_{i+1}'>, <\mathbf{B}_i', \mathbf{B}_{i+1}'>$. The product of the two terms in (7), then, expresses the difference in shape of $\mathbf{A}'$ and $\mathbf{B}'$ as a mean distance error. Note that (7) is symmetric with respect to $\mathbf{A}$ and $\mathbf{B}$.

E_S is not, in itself, sufficient to ensure a good correspondence between $\mathbf{A}$ and $\mathbf{B}$. It might be possible to

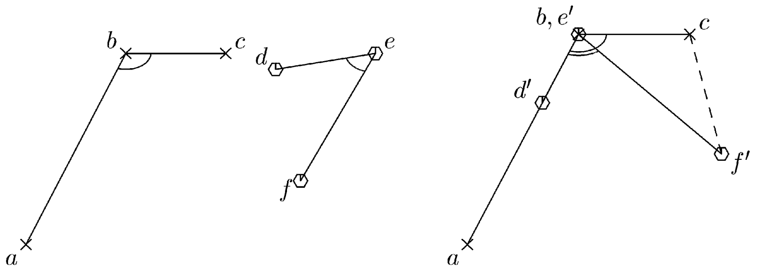


Fig. 3. Calculation of the local shape operator, $S(<\mathbf{a},\mathbf{b},\mathbf{c}>,<\mathbf{d},\mathbf{e},\mathbf{f}>)$.The triplet $<\mathbf{d},\mathbf{e},\mathbf{f}>$ is translated and rotated to $<\mathbf{d}',\mathbf{e}',\mathbf{f}'>$ so that $\mathbf{e}'$ and $\mathbf{b}$ are coincident and the directions $(\mathbf{b}-\mathbf{a})/\|\mathbf{b}-\mathbf{a}\|$ and $(\mathbf{e}'-\mathbf{d}')/\|\mathbf{e}'-\mathbf{d}'\|$ are equal. S is then given by $2\|\mathbf{c}-\mathbf{f}'\|/(\|\mathbf{c}-\mathbf{b}\|+\|\mathbf{f}-\mathbf{e}\|)$.

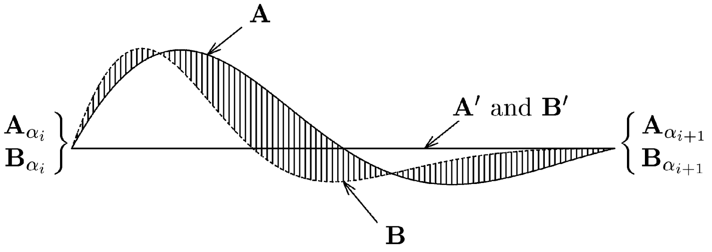


Fig. 4. Calculation of the local area difference operator, $R(\mathbf{A},\alpha_i,\alpha_{i+1},\mathbf{B},\beta_i,\beta_{i+1})$—the shaded area defines R.

construct a $\mathbf{B}'$ on $\mathbf{B}$ which is similar to $\mathbf{A}'$, but which does not *represent* $\mathbf{B}$ in the same way that $\mathbf{A}'$ represents $\mathbf{A}$. We need to also ensure that the manner in which $\mathbf{A}'$ differs from $\mathbf{A}$ is as similar as possible to the manner in which $\mathbf{B}'$ differs from $\mathbf{B}$. To accomplish this, we define:

$$E_R = \frac{2}{L(\mathbf{A}')+L(\mathbf{B}')}\sum_{i=1}^{n_\Phi} R(\mathbf{A},\alpha_i,\alpha_{i+1},\mathbf{B},\beta_i,\beta_{i+1}), \qquad (9)$$

where R is the local area difference operator for the segments $<\mathbf{A}'_i,\mathbf{A}'_{i+1}>$ and $<\mathbf{B}'_i,\mathbf{B}'_{i+1}>$. R returns the absolute difference in the representation errors of $\mathbf{A}$ and $\mathbf{B}$ for the given segment $<i,i+1>$—see Fig. 4. Again, $L(\mathbf{A}'),L(\mathbf{B}')$ are the total polygonal arc path-lengths of the polygons $\mathbf{A}'$, $\mathbf{B}'$, respectively. The representation error is simply the area between the sparse polygon and its pixellated boundary. Note again that (8) is symmetric with respect to $\mathbf{A}$ and $\mathbf{B}$.

E_R, then, measures the difference in the representation errors of $\mathbf{A}'$ and $\mathbf{B}'$ expressed as a mean distance error and is thus directly comparable with E_S. The parameter λ, expresses the relative contribution of the two terms in the cost function. We have determined suitable values of λ experimentally by comparing the position of landmarks generated manually with those generated automatically for λ in the range [0..1]—see Section 6. We note here that both S and R are local operators. This means that the values of S for each triplet in $\mathbf{A}$ and $\mathbf{B}$ may be stored and reused. Only the values of S for which corresponding triplets in $\mathbf{A}$ and $\mathbf{B}$ have been changed need to be recalculated to determine E_S each time Φ is altered. Similarly, only values of R for which corresponding segments in $\mathbf{A}$ and $\mathbf{B}$ have to be recalculated in order to determine E_R. This leads to a computationally efficient algorithm. The local nature of these operators also ensures that the legality (nonfolding) of the correspondences we have generated is maintained.

Given the expression for E in (6), we require a method modifying the correspondences, ϕ_i, in order to minimize E and, thus, bring $\mathbf{A}'$ and $\mathbf{B}'$ into better correspondence.

```
for $j$ = $N, N-1, \ldots, 1$
    set $f$ = 1/2
    do
        do
            for all indices $\beta_i$
                set $\delta = backwards(\beta_{i-j}, \beta_{i-j+1}, f)$
                set $\beta'_k = \beta_k - \delta$ for $k = i-j+1 \ldots i$
                evaluate $E$ using $\beta'_k$ in place of $\beta_k$
                set $\delta = forwards(\beta_i, \beta_{i+1}, f)$
                set $\beta'_k = \beta_k + \delta$ for $k = i-j+1 \ldots i$
                evaluate $E$ using $\beta'_k$ in place of $\beta_k$
                accept the best (if any) improvement in $E$ for $i$
        while improvement in $E$ continues
        set $f = f/2$
    while maximum possible movement of any $\beta_i$ is $\geq 1$ pixel
```

Fig. 5. Greedy algorithm used to minimize E in (6) for fixed $\mathbf{A}'$.

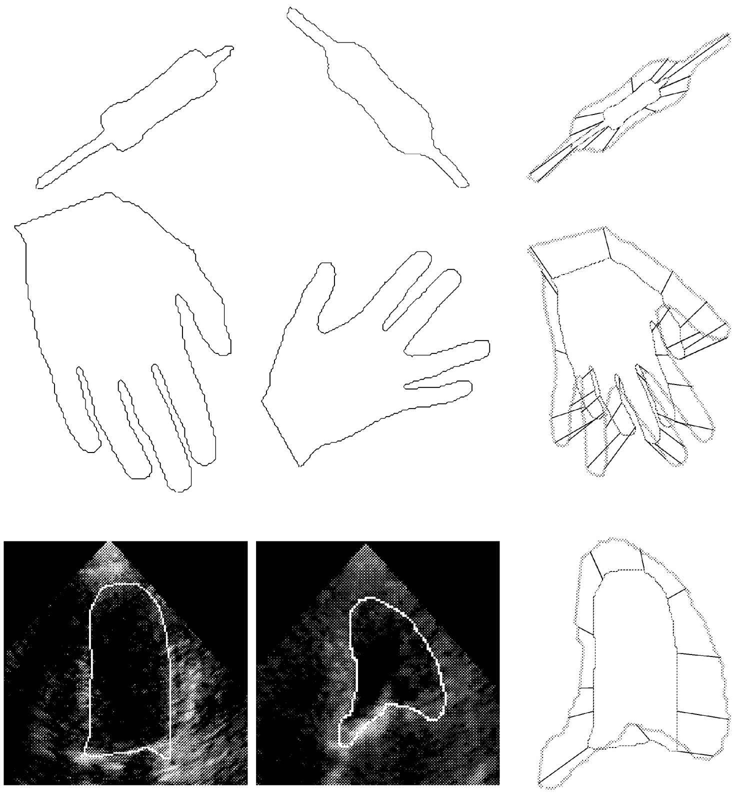


Fig. 6. Polygon-based correspondence applied to various shapes. The first column represents shape $\mathbf{A}$, the second $\mathbf{B}$, and the third shows the set of connections between the corresponding points of the sparse polygons.

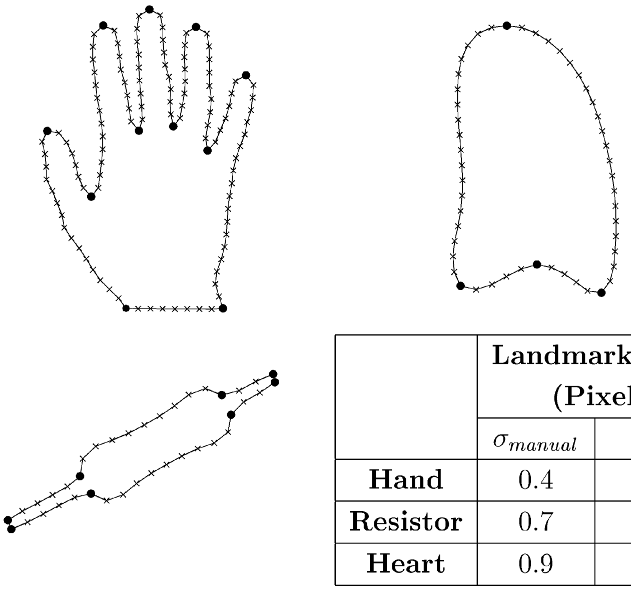
The first decision to make is whether to adjust the α values, the β values, or both. Now, the polygons $\mathbf{A}'$ and $\mathbf{B}'$ are intended to be sparse polygonal approximations of $\mathbf{A}$ and $\mathbf{B}$, respectively. If both the α and β values are allowed to vary there is no longer any guarantee that the resulting sparse polygons will be good approximations to $\mathbf{A}$ and $\mathbf{B}$. Consequently, we require either $\mathbf{A}'$ or $\mathbf{B}'$ remain fixed. Assume that $\mathbf{A}'$ is fixed. The greedy descent algorithm shown in Fig. 5 is used to minimize E in (6).

The greedy algorithm operates as follows: First, the number of indices to be adjusted simultaneously, j, is set. Groups of j contiguous points on $\mathbf{B}'$ are moved distances determined by f and tested to see whether the value of E is reduced. Any such improvement is accepted. The size of the adjustment of the β values is computed by the $backwards$ and $forwards$ operators. These operators compute the number of pixels to move the specified fraction, f, of the distance towards β_{i-j} from β_{i-j+1} ($backwards$) or towards β_{i+1} from β_i ($forwards$). The points on $\mathbf{B}'$ are repeatedly visited until no improvement in E is found for the particular values of j and f. The maximum amount by which points are allowed to move is then reduced (halved in our experiments) and the process repeated until the amount by which any point can be moved drops below a single pixel. The number of indices to be simultaneously adjusted, j, is repeatedly reduced until $j = 1$. In all of our experiments, we have found a value of $N = 2$ to be adequate i.e., the algorithm first moves pairs of points on $\mathbf{B}'$ until no further improvement can be made and then moves single points. Using $N > 2$ simply takes longer and does not appear to produce a result which is significantly different.

The overall approach we employ is to apply the optimization algorithm twice, each time starting with the same data, i.e., the pair of polygons $\mathbf{A}'$ and $\mathbf{B}'$ generated by the path-based correspondence algorithm. In the first application, the polygon $\mathbf{A}'$ is fixed and $\mathbf{B}'$ located to minimize E in (6), E_1 say. In the second application, $\mathbf{B}'$ is fixed and $\mathbf{A}'$ located to minimize E, E_2 say. The pair of sparse polygons which correspond to $\mathrm{Min}(\mathrm{E}_1, \mathrm{E}_2)$ is output by the algorithm.

6 Results

To investigate the performance of the correspondence algorithm described in Section 5, we have used three classes of object—hands, left ventricles of the heart, and resistors on a printed circuit board. The outlines of resistors and hands were both produced by photographing example objects on a white background, digitizing the images to 512×512, and thresholding. Outlines of ventricles were produced by manual tracing of ultrasound images frame-grabbed using 512×512 pixels. The shapes within each class vary nonrigidly and the type of nonrigid deformation is different for each of the classes. Fig. 6 shows the result of applying the polygon-based correspondence algorithm to difficult pairs of shapes from each of these three classes. The value of λ employed to generate these and all the other results we present was 0.2. We will see later how this value of λ was selected.



	Landmark Error (Pixels)	
	σ_{manual}	σ_{auto}
Hand	0.4	1.1
Resistor	0.7	1.4
Heart	0.9	2.7

Fig. 7. Manually selected major landmarks, •, equally-spaced minor landmarks, ×, and pixel location errors for manual landmark identification, σ_{manual}, and automatically generated landmarks, σ_{auto}. Number of examples: hand = 18, resistor = 33, heart = 66. Number of pixels per boundary: hand ≈ 650, resistor/heart ≈ 300.

The 18 examples of boundaries of the hand were generated from the same person, but with different positions of the thumb and fingers. When an outline of the hand is observed from above, the digits appear to shorten as the knuckles are raised off the plane and elastic deformation of the skin is observed as the digits change position. We used 66 examples of apical 4-chamber 2D echocardiograms of a beating heart. These exhibit considerable nonrigid deformation of the boundary of the left ventricle due to muscular activity and change in 2D view as the heart naturally rotates as it beats. Further shape changes occur when one compares hearts from different individuals. The 33 examples of resistors we have used vary principally in two ways: 1) the position of the body of the resistor on the wire on which it is mounted and 2) the shape of the body of the resistor. Surprisingly, this has proven to be the most difficult of the classes of shape we have considered. This is because the landmark representing the junction of the wire and the body of the resistor must be accurately located in order to generate a useful statistical model. Poor correspondence in this area results in a model which generates many implausible examples (poor *specificity*).

As already mentioned, our goal is to identify a set of landmarks automatically on each of a set of examples in order to generate statistical shapes models. Previously, we have generated these landmarks manually by placing a small number of *major* landmarks on each example and generating *minor* landmarks equally spaced between major landmarks. In general, this consists of a learning phase of around one hour during which the operator decides upon a suitable set of homologous landmarks for the class. It then takes a few minutes to landmark each example. We have used this technique routinely for generating landmark data. To investigate the accuracy of the polygon-based correspondence algorithm presented in Section 5 with respect to this automatic landmarking task, we have compared the position of landmarks placed manually on a set of examples with those generated using a binary tree of merged pairs of shapes, as described in Section 3, which incorporated the polygon-based corresponder. The comparison was made as follows:

1. A set of landmarks was generated manually on each of a set of examples M times (six in our experiments). The mean set of landmarks for each example was computed and regarded as *ground truth*. The standard deviation of the distance of all landmarks on all sets of examples from their mean position was computed:

$$\sigma_{manual} = \sqrt{\frac{1}{MNL}\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=1}^{L}\|\hat{\mathbf{x}}_{ijk} - \bar{\mathbf{x}}_{jk}\|^2},$$

   where $\hat{\mathbf{x}}_{ijk}$ is the kth manual landmark on the jth shape for the ith set of landmarks, $\bar{\mathbf{x}}_{jk}$ is the mean position of the kth landmark on the jth example, N is the number of examples, and L is the number of landmarks. σ_{manual} represents the uncertainty in landmark position associated with selecting the landmarks manually.
2. A merge-tree was generated for the given set of examples. The pixels on the mean shape generated by the tree which best corresponded to the ground truth landmarks were identified as follows: For each landmark, k $(1 \leq k \leq L)$, the pixel on the mean shape corresponding to:

$$\min_{i}\sum_{j=1}^{N}\|\mathbf{x}_{jp(i,j)} - \bar{\mathbf{x}}_{jk}\|$$

   was identified and labeled i_k, where $\mathbf{x}_{jp(i,j)}$ is the $p(i,j)$th pixel on the jth example and $p(i,j)$ is the pixel index projection operator for the ith pixel of the mean shape and the jth example.

TABLE 1
Mean Landmark Representation Error Due to Model Truncation

	Mean Representation Error of Landmarks (Pixels)					
	Hand		**Resistor**		**Heart**	
Modes	**manual**	**auto**	**manual**	**auto**	**manual**	**auto**
0	13.7	13.8	5.8	5.7	13.7	16.6
1	8.3	8.1	1.7	1.8	7.9	10.3
2	4.3	4.1	1.2	1.0	4.7	6.9
3	2.1	2.0			3.0	4.9
4	1.3	1.1			2.1	3.5
5					1.5	2.4
6					1.0	1.7
7						1.3
8						1.0

When 0 modes are specified, only the mean shape is used for shape respresentation.

3. The root mean squared distance between the automatically generated landmarks and ground truth was computed:

$$\sigma_{auto} = \sqrt{\frac{1}{NL}\sum_{j=1}^{N}\sum_{k=1}^{L}\|\mathbf{x}_{jp(i_k,j)} - \bar{\mathbf{x}}_{jk}\|^2}.$$

σ_{auto} represents the pixel error in landmark position for landmarks generated automatically.

In Fig. 7, we show the landmarks used for the three examples to conduct the experiment. The values of σ_{manual} and σ_{auto} for the three cases are also shown in Fig. 7. These results show that the errors (with respect to the ground truth landmarks) associated with generating the landmarks automatically are of the same order as the misplacement errors associated with identifying the landmarks manually.

While this comparison indicates that the framework for automatic landmark identification generates similar landmarks to those generated manually, it gives no indication as to whether the errors involved are in any way systematic. Recall that the *minor* manual landmarks are generated equally spaced between *major* manual landmarks, which are the only landmarks actually identified manually. We have used this major/minor landmark approach to minimize the amount of human interaction. It may be the case, however, that there are further landmarks that can be identified manually (the knuckles of the thumb and fingers or the shoulder on the body of the resistor). These "unidentified" landmarks are poorly localized using the equal spacing of minor landmarks described above. The automatic framework may identify these landmarks more accurately than the manual approach, thus leading to systematic differences between the two approaches.

Our ultimate goal is to build statistical shape models—PDMs—from the landmark data. Recall from Section 2 that a PDM consists of a mean set of landmarks

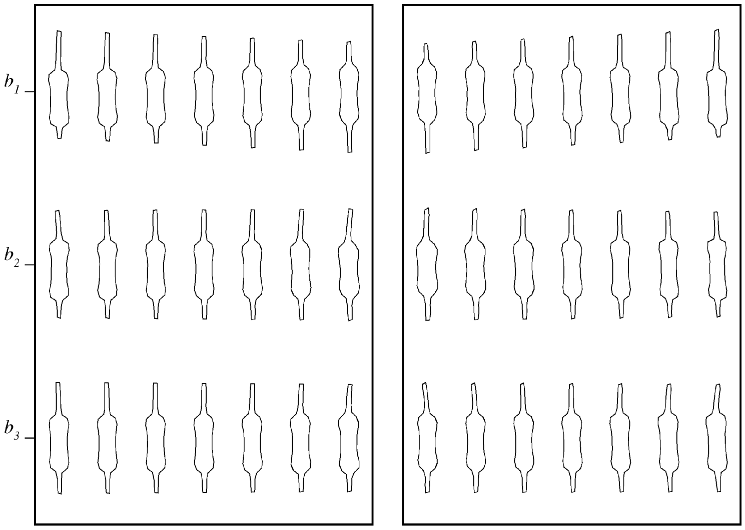


Fig. 8. Modes of variation for the resistor data. The modes are displayed by varying the shape parameters, b_i. The modes for the PDM built from the manual landmarks are shown on the left and from the automatic landmarks on the right. The (arbitrary) signs of the eigenvectors in the two models are different, producing mirror images of the left with the right.

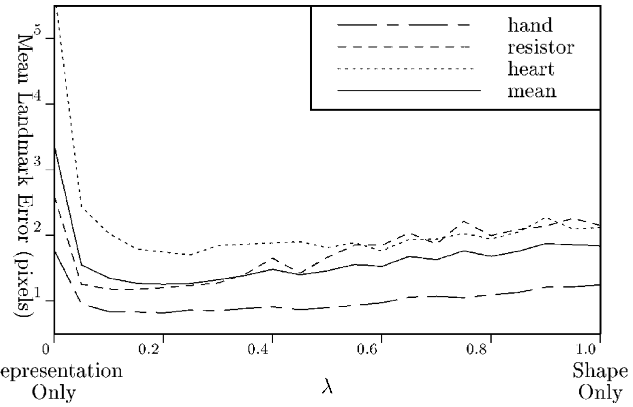


Fig. 9. Effects of the parameter λ on the error in the position of automatically generated landmarks. λ controls the balance between the two terms in the cost function—(6)—relating to the errors in shape and representation.

and a small number of basis eigenvectors which define how the training examples tend to vary from the mean; the so-called *modes of variation*. When a truncated set of eigenvectors is used in a PDM, it is no longer possible to represent exactly the landmarks for each of the training examples. We use the representation error as a basis for selecting the number of eigenvectors to retain in the model. It is also a useful basis for comparing the performance of manual and automatic landmarking.

Table 1 shows the mean pixel representation error for all landmarks for PDMs of the hand, resistor, and heart as the number of eigenvectors is varied. PDMs were generated from the ground truth manual landmarks, $\bar{\mathbf{x}}_{jk}$, and landmarks generated by the automatic system, $\mathbf{x}_{jp(i_k,j)}$. For the hand and resistor examples, it is clear that the models are very similar. Qualitatively, the modes of variation for the PDMs built from the manual and automatic landmarks are almost identical—see Fig. 8. This suggests that the errors between the automatic landmarks and the manual landmarks for the hand and resistor data presented in Fig. 7 are systematic. For the heart, Table 1 shows that the landmarks generated manually are better localized than those generated automatically. In this case, the errors are probably random. The PDMs are still similar, but, for a given approximation error, a few more modes are required for the PDM built from the automatic landmarks, i.e., the model is less *compact*. Qualitatively, the first three modes are indistinguishable.

Finally, we describe the experiment we conducted to identify suitable values of λ in (6) and the sensitivity of the system with respect to this parameter. The experiment was similar to that used to compare the position of the landmarks generated manually with those generated automatically. Rather than calculate the spread of the error, σ_{auto}, we calculated the mean error in landmark placement:

$$\mu_\lambda = \frac{1}{NL}\sum_{j=1}^{N}\sum_{k=1}^{L}\|\mathbf{x}_{jp(i_k,j)} - \bar{\mathbf{x}}_{jk}\|,$$

for the automatic landmarks $\mathbf{x}_{jp(i_k,j)}$ generated using a particular value of λ. Now, the two terms E_S and E_R in (6) have the same units (both are distance errors) and one would expect a "natural" value of λ to lie in the region of 0.5. Fig. 9 shows the value of μ_λ for λ in the range $[0..1]$ for the hand, resistor, and heart. Inspection of Fig. 9 indicates that, ignoring very small values of λ, the error associated with landmark position is not critical with respect to λ. Values of λ in the range $[0.1..0.4]$, tend to produce the best results and we selected $\lambda = 0.2$ as a default value.

7 Conclusions

We have presented a novel method for the nonrigid correspondence of two closed, pixellated boundaries. The method is based on generating corresponding sparse polygonal approximations for each shape. No curvature estimation of either boundary is required and the algorithm requires only a single control parameter, λ, the value of which is not critical. Results have been presented which demonstrate the ability of the algorithm to provide accurate, nonrigid correspondences for three classes of shape—hands, chambers of the heart, and resistors on printed circuit boards. The algorithm is computationally efficient —a pair of hands ($\approx$ 650 pixels per boundary) requires 1.3 CPU seconds and a pair of resistors/left ventricles ($\approx$ 300 pixels per boundary) requires 0.5 CPU seconds on a Sun SPARCstation 20.

The corresponder has been used within a framework for generating landmarks automatically for a set of example shapes. The framework uses a binary tree of matched pairs of shapes. Results have been presented which demonstrate that landmarks similar to those identified manually are produced by the framework when the polygon-based correspondence algorithm is employed as the method of pairwise correspondence. At present, the only way to detect a failure of the algorithm to correctly correspond two example shapes is to visually inspect the PDM that is produced from the resulting landmarks. A further development of the framework might be the detection of bad matched and merged pairs by comparison with the class of examples as a whole.

The framework for automatic landmark identification which we have presented and the PDM approach to modeling the shape of a class of objects represented by landmark data are entirely general. The polygon-based correspondence algorithm employed within the framework, however, considered only objects represented by closed boundaries. The extension of the algorithm to open curves is straightforward if it is assumed that the end-points of the curves correspond, thus simplifying the correspondence problem. Although the correspondence algorithm should be able to handle self-occluding boundaries, what is less straightforward is the extension of the algorithm to objects represented by multiple open/closed boundaries, e.g., faces. Addressing this problem is part of our current research, together with extending the correspondence algorithm to voxellated surfaces in 3D.

Acknowledgments

This research was funded by the United Kingdom Engineering and Physical Sciences Research Council.

REFERENCES

[1] T.F. Cootes, D.H. Cooper, C.J. Taylor, and J. Graham, "A Trainable Method of Parametric Shape Description," *Image and Vision Computing,* vol. 10, pp. 289-294, June 1992.

[2] F.L. Bookstein, *Morphometric Tools for Landmark Data.* Cambridge Univ. Press, 1991.

[3] C. Goodall, "Procrustes Methods in the Statistical Analysis of Shape," *J. Royal Statistical Soc. B,* vol. 53, no. 2, pp. 285-339, 1991.

[4] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham, "Active Shape Models—Their Training and Application," *Computer Vision and Image Understanding,* vol. 61, pp. 38-59, Jan. 1995.

[5] A. Hill, T.F. Cootes, C.J. Taylor, and K. Lindley, "Medical Image Interpretation: A Generic Approach Using Deformable Templates," *J. Medical Informatics,* vol. 19, no. 1, pp. 47-59, 1994.

[6] A. Lanitis, C.J. Taylor, and T.F. Cootes, "A Unified Approach to Coding and Interpreting Face Images," *Proc. Fifth Int'l Conf. Computer Vision,* pp. 368-373, June 1995.

[7] J.A. Marchant and C.M. Onyango, "Fitting Grey Level Point Distribution Models to Animals in Scenes," *Image and Vision Computing,* vol. 13, pp. 3-12, Feb. 1995.

[8] A. Baumberg and D. Hogg, "Learning Flexible Models from Image Sequences," *Proc. Third European Conf. Computer Vision,* pp. 299-308, 1994.

[9] A. Baumberg and D. Hogg, "An Adaptive Eigenshape Model," *Proc. Sixth British Machine Vison Conf.,* D. Pycock, ed., pp. 87-96, Sept. 1995.

[10] A. Hill and C.J. Taylor, "Automatic Landmark Generation for Point Distribution Models," *Proc. Fifth British Machine Vision Conf.,* E. Hancock, ed., pp. 429-438, Sept. 1994

[11] S.C. Joshi, A. Banerjee, G.E. Christensen, J.G. Csernansky, J.W. Haller, M.I. Miller, and L. Wang, "Gaussian Random Fields on Sub-Manifolds for Characterizing Brain Surfaces," *Proc. 15th Conf. Information Processing in Medical Imaging,* J. Duncan and G. Gindi, eds., pp. 381-386, 1997.

[12] G.E. Christensen, S.C. Joshi, and M. Miller, "Volumetric Transformation of Brain Anatomy," *IEEE Trans. Medical Image,* vol. 16, pp. 864-877, 1997.

[13] M. Fleute and S. Lavallée, "Building a Complete Surface Model from Sparse Data Using Statistical Shape Models: Application to Computer Assisted Knee Surgery," *Medical Image Computing and Computer-Assisted Intervention,* pp. 878-887, 1998.

[14] G. Szeliski and S. Lavalée, "Matching 3D Anatomical Surface with Non-Rigid Deformations Using Octree-Splines," *Int'l J. Computer Vision,* vol. 18, no. 2, pp. 171-186, 1996.

[15] A. Kelemen, G. Székely, and G.G. Gerig, "Three-Dimensional Model-Based Segmentation," Technical Report 178, Image Science Lab, ETH Zürich, 1997.

[16] C. Brechbühler, G. Gerig, and O. Kübler, "Parameterisation of Closed Surfaces for 3D Shape Description," *Computer Vision, Graphics, and Image Processing,* vol. 61, pp. 154-170, 1995.

[17] A.C.W. Kotcheff and C.J. Taylor, "Automatic Construction of Eigenshape Models by Direct Optimisation," *Medical Image Analysis,* vol. 2, no. 4, pp. 303-314, 1998.

[18] P. Zhu and P.M. Chirlian, "On Critical Point Detection of Digital Shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 8, pp. 737-748, Aug. 1995.

[19] J. Duncan, R.L. Owen, L.H. Staib, and P. Anandan, "Measurement of Non-Rigid Motion Using Contour Shape Descriptors," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 318-324, 1991.

[20] C. Kambhamettu and D.B. Goldgof, "Point Correspondence Recovery in Non-Rigid Motion," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 222-227, 1992.

[21] I. Cohen, N. Ayache, and P. Sulger, "Tracking Points on Deformable Objects Using Curvature Information," *Proc. Second European Conf. Computer Vision,* G. Sandini, ed., pp. 458-466, May 1992.

[22] H.D. Tagare, D. O'Shea, and A. Rangarajan, "A Geometric Criterion for Shape-Based Non-Rigid Correspondence," *Proc. Fifth Int'l Conf. Computer Vision,* pp. 434-439, June 1995.

[23] G.L. Scott and H.C. Longuet-Higgins, "An Algorithm for Associating the Features of Two Images," *Proc. Royal Statistical Soc. of London,* vol. 244, pp. 21-26, 1991.

[24] L.S. Shapiro and J.M. Brady, "A Modal Approach to Feature-Based Correspondence," *Proc. Second British Machine Vison Conf.,* P. Mowforth, ed., pp. 78-85, Sept. 1991.

[25] S. Sclaroff and A.P. Pentland, "Modal Matching for Correspondence and Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 6, pp. 545-561, June 1995.

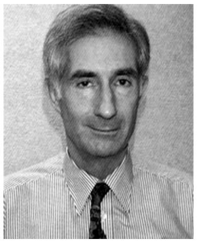
[26] A. Rangarajan, E. Mjolsness, S. Pappu, L. Davachi, P.S. Goldman-Rakic, and J.S. Duncan, "A Robust Point Matching Algorithm for Autoradiograph Alignment," *Visualisation in Biomedical Computing,* pp. 277-286, 1996.

[27] A. Rangagajan, H. Chui, and F.L. Bookstein, "The Softassign Procrustes Matching Algorithm," *Proc. 15th Conf Information Processing in Medical Imaging,* pp. 29-42, 1997.

[28] S. Umeyama, "Least-Squares Estimation of Transformation Parameters Between two Point Patterns," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, no. 4, pp. 376-380, Apr. 1991.

[29] B.K.P. Horn, "Closed-Form Solution of Absolute Orientation Using Orthonormal Matrices," *J. Optical Soc. Am.,* vol. 5, pp. 1,127-1,135, July 1988.

Andrew Hill received the BSc degree in computer science/mathematics and the PhD degree in computer science in 1985 and 1988, respectively, from the University of York, England. After completing his PhD, he spent the next eight years as a research associate at the Wolfson Image Analysis Unit, University of Manchester, England. He has spent the last two years with Kestra, a new company developing automated optical inspection systems for the electronics industry. His research interests include model-based approaches for object recognition and stochastic methods of optimization.



Chris J. Taylor received the BSc degree in physics and the PhD degree in computer image analysis from the University of Manchester, England, in 1967 and 1972, respectively. He is currently a professor of medical biophysics and computer science and director of the Wolfson Image Analysis Unit at the University of Manchester. Dr. Taylor's long-term interest is in practical applications of computer vision in both medicine and industry. From 1974 to 1986, he was involved in developing high performance hardware architectures and model-based algorithms on which commercial systems for medical analysis, environmental monitoring, and industrial inspection were based. His current research interests are in the areas of model-based vision (with particular emphasis on modeling variable objects), medical image interpretation, user-configurable inspection systems, and vision-based human-computer interaction, including face and gesture recognition.

Alan D. Brett received the BSc degree in physics in 1988 and the MSc degree in applied radiation physics in 1991, both from the University of Birmingham, England. In 1997, he received the PhD degree in medical imaging from University College, London, England. Since 1995, he has been employed as a research associate in the Wolfson Image Analysis Unit at the University of Manchester, England. His research interests include three-dimensional modeling of shape variation, automated shape model building, and medical image interpretation.