A Method of Non-rigid Correspondence for Automatic Landmark Identification

A. Hill and C. J. Taylor
Department of Medical Biophysics
University of Manchester
ah@sv1.smb.man.ac.uk

Abstract

A method for corresponding the boundaries of two shapes is presented. The algorithm employs a sparse polygonal approximation of one of the boundaries, generated using a critical point detection algorithm. A matching sparse polygon is sought on the second of the two boundaries which is both similar in shape to and has a similar representation error to the first. Optimisation of a cost function using a greedy algorithm locates the sparse polygon on the second boundary. Results are presented for three classes of shape which exhibit various types of non-rigid deformation. The algorithm is also applied to an automatic landmark identification task.

Keywords: Correspondence, Critical Points, Polygonal Approximation, Automatic Landmarks, Flexible Templates, Point Distribution Models.

1 Introduction

A frequently encountered problem in computer vision is that of finding 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 non-rigid transformation is required to map one boundary onto the other. The particular application which motivates our work is that of generating *landmarks* automatically on a set of examples. By a landmark we mean a point which identifies a salient feature on an object and which is present on each of a set of examples of the object. The landmarks can be used to train a statistical flexible template known as a Point Distribution Model (PDM) [2]. This avoids the time-consuming and subjective process of identifying the landmark points manually.

In a previous publication [4] we described a framework for automatically generating landmarks for a training set of shapes. The first step in our method was to generate an approximate set of landmarks for each example using a binary tree of merged pairs of shapes. The algorithm for generating the tree relied 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). The correspondence method we described was based on matching the curvature of the two boundaries using Dynamic Programming (DP). This DP approach to pair-wise

BMVC 1996 doi:10.5244/C.10.23

correspondence has not proven to be sufficiently robust or accurate to be generally useful and does not extend easily to 3D, which is one of our major objectives. We have also found that several other published methods of correspondence are not adequate for this task due to poor robustness and/or accuracy (see section 2).

Here, we present a novel method of pair-wise correspondence which has proved both accurate and robust. The algorithm employs a sparse polygonal approximation to one of the two boundaries which is transformed onto the other boundary via an optimisation scheme which minimises a cost function. The final value of the cost function is used to assess the quality of the match. 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. We also present results for the automatic landmark identification task outlined above.

2 Background

Duncan *et al* [3] Kambhamettu and Goldgof [5] and Cohen *et al* [1] all propose methods of correspondence based on the minimisation of a cost function which involves the difference in the curvature of two boundaries (or surfaces). As pointed out by Tagare *et al* [9], however, curvature is a rigid invariant of shape and its applicability to non-rigid correspondence is problematic. Our own experience of minimising the difference in curvature of two boundaries [4] confirms this. Tagare *et al* [9] recently proposed a method of correspondence based on the minimisation of a 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 optimisation scheme employed by Tagare to minimise the cost function requires five control parameters which may make the method difficult to use routinely as part of an automatic system.

The related methods of Scott and Longuet-Higgins [7] Shapiro and Brady [8] and Sclaroff and Pentland [6] 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 [6] is proposed for non-rigid 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 a FEM it is necessary to construct the Galerkin interpolation matrix, which is the inverse of the matrix $\mathbf{G} = [g_i(\mathbf{x}_j)]\,(1 \leq i,j \leq m)$ 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 [6]. 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.

3 Polygon-based Correspondence

In this section we describe a new correspondence algorithm which transforms a given discretised 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 normalised such that the centre-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 $\Phi = \{(\alpha_i, \beta_i); 1 \leq i \leq n_\Phi\}$. 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 sub-polygons of $\mathbf{A}$ and $\mathbf{B}$ respectively. The values α_i, β_i satisfy the following conditions:

$$\begin{array}{c} 1 \leq \alpha_i \leq n_A \\ 1 \leq \beta_i \leq n_B \end{array} \qquad (1)$$

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

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 condition(2) ensures that the indices $\{\alpha_i\}$, $\{\beta_i\}$ form legal sub-polygons of $\mathbf{A}$ and $\mathbf{B}$ respectively.

The correspondence algorithm comprises of three parts:

1. Generation of a sparse polygonal approximation, $\mathbf{A}'$, of $\mathbf{A}$. This sparse polygon determines the number of ordered pairs, n_Φ, and is fixed for the remainder of the correspondence algorithm.
2. Generation of an initial estimate of the corresponding polygon, $\mathbf{B}'$, on $\mathbf{B}$. This is accomplished using a correspondence algorithm based on the polygonal arc path-lengths of $\mathbf{A}$ and $\mathbf{B}$.
3. Refinement of the initial set of correspondences using an optimisation scheme which modifies the values $\{\beta_i\}$ in order to generate a sparse polygon $\mathbf{B}'$ which is similar in shape to $\mathbf{A}'$ and has a similar *representation error* with respect to its original boundary $\mathbf{B}$ as $\mathbf{A}'$ has with respect to $\mathbf{A}$.

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

3.1 Sparse Polygon Generation

To generate a sparse polygon, $\mathbf{A}'$, on $\mathbf{A}$ we have used the critical point detection (CPD) algorithm described by Zhu and Chirlian [11]. 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 neighbours. An iterative decimation process is used which removes the point with the smallest critical value, re-computes the critical value of the immediate neighbours of the point which has just been deleted and re-identifies the point with smallest critical value. The process terminates when the remaining smallest critical value is above some threshold set by the user.

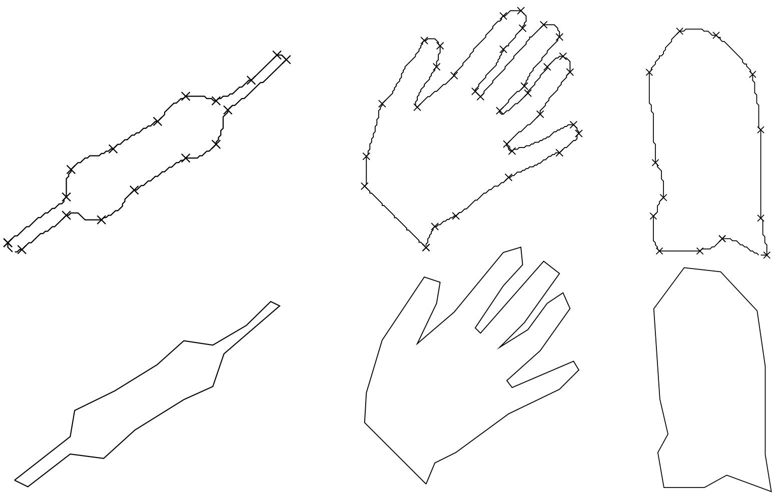


Figure 1: Critical Points for Various Shapes

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) *vs* (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 figure 1.

3.2 Path-based Correspondence

In this section we describe a method of generating the polygon $\mathbf{B}'$ which corresponds approximately to the polygon $\mathbf{A}'$. 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% 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% 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 $\sum_{k=i}^{j-1} ||\mathbf{A}_{k+1} - \mathbf{A}_k||$.

The path matching correspondence algorithm exhaustively tests every pixel $\mathbf{B}_i (1 \leq i \leq n_B)$ in the following manner: Set $\beta_1 = i$ to give the first corresponding pair of points $(\mathbf{A}_{\alpha_1}, \mathbf{B}_{\beta_1})$. Determine the remaining $\beta_j (2 \leq j \leq n_\Phi)$ values by projecting the relative path-length spacing of the polygon segments $< \mathbf{A}_{\alpha_{j-1}}, \mathbf{A}_{\alpha_j} >$ onto $\mathbf{B}$. Determine the pose, Q, which satisfies:

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

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 [10] for a solution to equation(3).

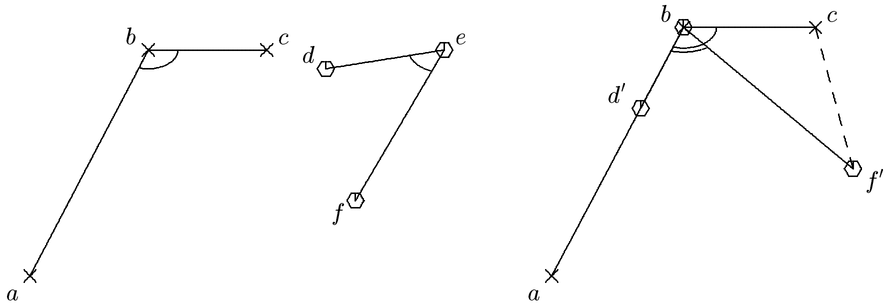


Figure 2: Calculation of $S(<a,b,c>,<d,e,f>) = ||c-f'||/||c-b||$

Now the index i for which $E_i^2 \leq E_j^2 \,\forall\, j$ identifies the pixel on $\mathbf{B}$ which matches the pixel $\mathbf{A}_{\alpha_1}$ and results in the best correspondence (in a Euclidean sense) when the points which define $\mathbf{B}'$ are spaced along $\mathbf{B}$ to reproduce the spacing of the points $\mathbf{A}'$ along $\mathbf{A}$ and thus determines the initial set of β values. We note here that this path-matching algorithm can recover exactly any Euclidean transformation of $\mathbf{A}$ i.e. $\mathbf{B} = Q(\mathbf{A})$, the normalised polygonal arc path-length of a boundary being invariant under a Euclidean transformation.

3.3 Optimisation Scheme

Given an initial set of correspondences generated by the path-based correspondence algorithm described above, an iterative local optimisation scheme can be used to modify the β values in order to minimise the following cost function:

$$E = \lambda E_S + E_R \tag{4}$$

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 $\mathbf{B}'$.

The term E_S is defined as:

$$\left(\frac{L(\mathbf{A}')}{n_\Phi}\right)\left(\frac{1}{n_\Phi}\sum_{i=1}^{n_\Phi} S(<\mathbf{A}'_{i-1},\mathbf{A}'_i,\mathbf{A}'_{i+1}>,<\mathbf{B}'_{i-1},\mathbf{B}'_i,\mathbf{B}'_{i+1}>)\right) \tag{5}$$

where $L(\mathbf{A}')$ is the total polygonal arc path-length of the polygon $\mathbf{A}'$. The first bracketed term in equation(5) simply defines the mean length of the segments which make up the polygon $\mathbf{A}'$. 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 figure 2. The value returned by S is the difference in shape expressed as a relative proportion of the length of the segment $<\mathbf{A}'_i,\mathbf{A}'_{i+1}>$. The product of the two terms in equation(5), then, expresses the difference in shape of $\mathbf{A}'$ and $\mathbf{B}'$ as a mean distance error.

E_S is not in itself sufficient to ensure a good correspondence between $\mathbf{A}$ and $\mathbf{B}$. It might well be possible to construct a $\mathbf{B}'$ on $\mathbf{B}$ which is similar to $\mathbf{A}'$ but

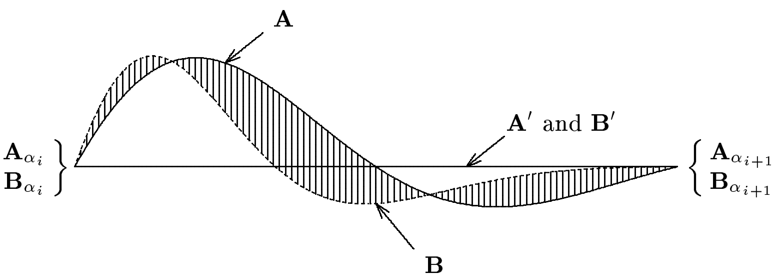


Figure 3: Calculation of $R(\mathbf{A}, \alpha_i, \alpha_{i+1}, \mathbf{B}, \beta_i, \beta_{i+1})$ – the shaded area defines R

which does not *represent* $\mathbf{B}$ in the same way that $\mathbf{A}'$ represents $\mathbf{A}$. We need also to 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:

$$E_R = \frac{1}{L(A')} \sum_{i=1}^{n_\Phi} R(\mathbf{A}, \alpha_i, \alpha_{i+1}, \mathbf{B}, \beta_i, \beta_{i+1}) \tag{6}$$

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 figure 3. The representation error is simply the area between the sparse polygon and its pixellated boundary.

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 empirically and found the range 0.1–0.4 to be useful for real data. For all the results presented in section 4 the value of λ was fixed at 0.2. We note here that both E_S and E_R are **local** operators. This means that E requires only partial re-computation when a small subset of the β indices are modified which leads to a computationally efficient algorithm.

Given the expression for E in equation(4) the β values are modified so as to minimise E thus bringing $\mathbf{A}'$ and $\mathbf{B}'$ into correspondence. We have used the following greedy descent algorithm to accomplish this:

```
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
```

First the number of indices to be adjusted simultaneously, j, is set. The algorithm moves groups of j contiguous points on $\mathbf{B}'$ distances determined by f and tests to see whether this reduces the value of E. 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 can be found for the particular values of j and f. The maximum amount by which points are allowed to move is then halved 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.

4 Results

To investigate the performance of the correspondence algorithm described in section 3 we have used three classes of object – hands, left ventricles of the heart and resistors on a printed circuit board. The shapes within each class vary non-rigidly and the type of non-rigid deformation is different for each of the classes. Figure 4 shows the result of applying the polygon-based correspondence algorithm to difficult pairs of shapes from each of these three classes.

The boundaries of the hand were generated for 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 observed as the digits change position. Apical 4-chamber 2D echocardiograms of a beating heart exhibit considerable non-rigid 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 resistors we have used vary principally in two ways: i) the position of the body of the resistor on the wire on which it is mounted and ii) the shape of the body of the resistor. Surprisingly, this has proved 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. We have used this technique routinely for generating landmark data. To investigate the accuracy of the polygon-based correspondence algorithm presented in section 3 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 [4], which incorporated the polygon-based corresponder. The non-leaf nodes of such a tree represent the mean of two other shapes which are themselves nodes in

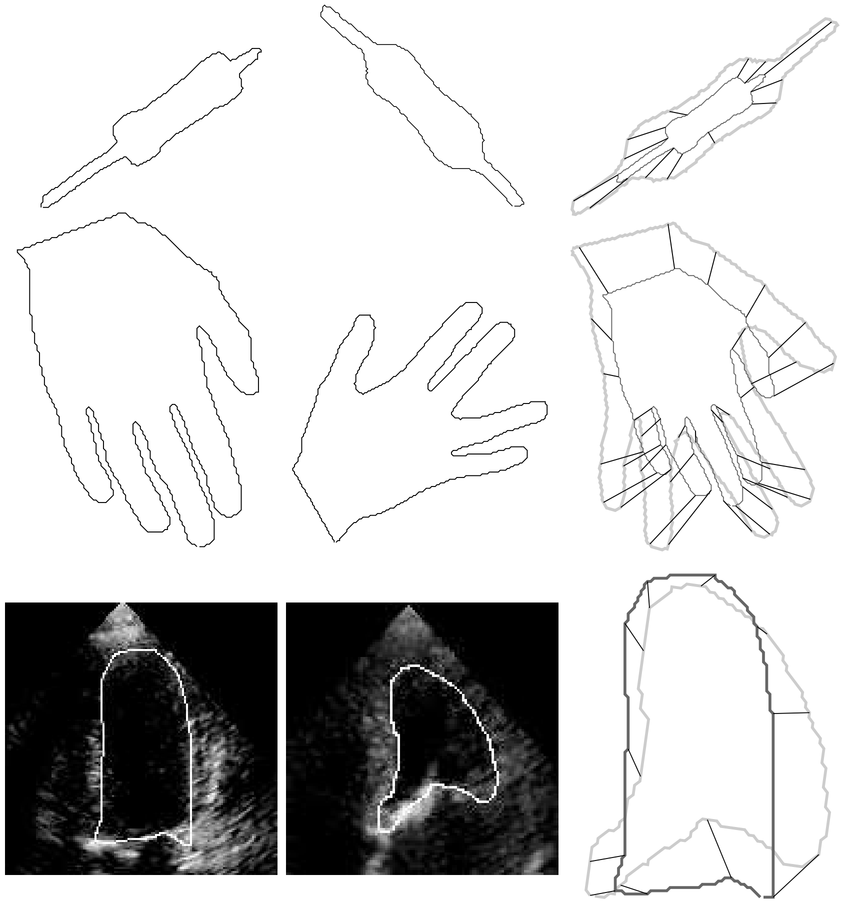
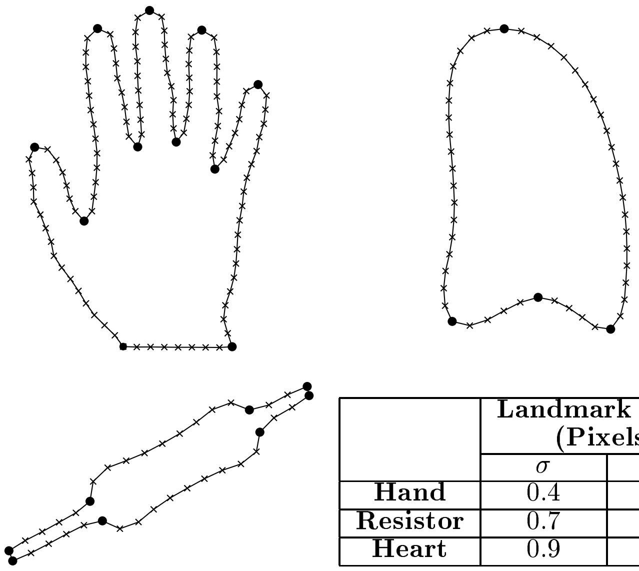


Figure 4: Polygon-based Correspondence Applied to Various Shapes. The first column represents shape **A**, the second **B** and the third shows the set of connections between the corresponding points of the sparse polygons.

the tree. The leaf nodes represent the original shapes in the training set and the root node represents the mean shape. Landmarks placed on the mean shape are projected from the root to the leaves. The comparison was made as follows:

1. A set of landmarks was generated manually on each of a set of examples M times (6 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: $\sigma = \sqrt{\frac{1}{MNL}\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=1}^{L}||\mathbf{x}_{ijk} - \bar{\mathbf{x}}_{jk}||^2}$ where $\mathbf{x}_{ijk}$ is the k^{th} landmark on the j^{th} shape for the i^{th} set of landmarks, $\bar{\mathbf{x}}_{jk}$ is the mean position of the k^{th} landmark on the j^{th} example, N is the number of examples and L is the number of landmarks. σ 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



	Landmark Error (Pixels)	
	σ	μ
Hand	0.4	0.9
Resistor	0.7	1.3
Heart	0.9	1.8

Figure 5: Manually Selected Major Landmarks, •, Equally-Spaced Minor Landmarks, ×, and Pixel Location Errors for Manual Landmark Identification, σ, and Automatically Generated Landmarks, μ. No. of examples: hand=18, resistor=33, heart=66. No. of pixels per boundary: hand≈650, resistor/heart≈300.

truth landmarks were identified as follows: For each landmark, k $(1 \leq k \leq L)$, the pixel on the mean shape corresponding to $\min\limits_{i} \sum_{j=1}^{N} ||\mathbf{x}_{jp(i,j)} - \bar{\mathbf{x}}_{jk}||$ was identified and labelled i_k, where $\mathbf{x}_{jp(i,j)}$ is the $p(i,j)^{th}$ pixel on the j^{th} example and $p(i,j)$ is the pixel index projection operator for the i^{th} pixel of the mean shape and the j^{th} example.

3. The mean distance between the automatically generated landmarks and ground truth was computed: $\mu = \frac{1}{NL} \sum_{j=1}^{N} \sum_{k=1}^{L} ||\mathbf{x}_{ji_k} - \bar{\mathbf{x}}_{jk}||$. μ represents the pixel error in landmark position for landmarks generated automatically.

In figure 5 we show the landmarks used for the three examples to conduct the experiment. The values of σ and μ for the three cases are also shown in figure 5. These results show that the landmarks selected automatically are, on average, within 0.9–1.8 pixels of the ground truth landmarks. This compares favourably with the error associated with placing the landmarks manually, the standard deviation of this error being between 0.4–0.9 pixels.

The polygon-based approach to correspondence has proved to be computationally efficient – a pair of hands (≈ 650 pixels per boundary) requires 0.9 CPU seconds and a pair of resistors/left ventricles (≈ 300 pixels per boundary) requires 0.4 CPU seconds on a Sun SPARCstation 20.

5 Conclusions

We have presented a novel method for the non-rigid correspondence of two closed, pixellated boundaries. The method is based on generating a sparse polygonal approximation to one shape and searching for a similar polygon on the other shape. No curvature estimation of either boundary is required and the algorithm requires only a single control parameter, λ. Results have been presented which demonstrate the ability of the algorithm to provide accurate, non-rigid correspondences for three classes of shape – hands, chambers of the heart and resistors on printed circuit boards. The appropriateness of the algorithm for the automatic landmark generation task has also been demonstrated. Our current research is concerned with extending polygon-based correspondence to objects represented by multiple open/closed boundaries and 3D voxellated surfaces.

References

[1] I. Cohen, N. Ayache, and P. Sulger. Tracking points on deformable objects using curvature information. In G. Sandini, editor, 2^{nd} *European Conference on Computer Vision*, pages 458–466, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.

[2] T. F. Cootes, D. H. Cooper, C. J. Taylor, and J. Graham. A trainable method of parametric shape description. *Image and Vision Computing*, 10(5):289–294, June 1992.

[3] J. Duncan, R. L. Owen, L. H. Staib, and P. Anandan. Measurement of non-rigid motion using contour shape descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 318–324, 1991.

[4] A. Hill and C. J. Taylor. Automatic landmark generation for point distribution models. In E. Hancock, editor, 5^{th} *British Machine Vison Conference*, pages 429–438, York, England, Sept. 1994. BMVA Press.

[5] C. Kambhamettu and D. B. Goldgof. Point correspondence recovery in non-rigid motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 222–227, 1992.

[6] S. Sclaroff and A. P. Pentland. Modal matching for correspondence and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6):545–561, 1995.

[7] G. L. Scott and H. C. Longuet-Higgins. An algorithm for associating the features of two images. *Proceedings of the Royal Statistical Society of London*, 244:21–26, 1991.

[8] L. S. Shapiro and J. M. Brady. A modal approach to feature-based correspondence. In P. Mowforth, editor, 2^{nd} *British Machine Vison Conference*, pages 78–85, Glasgow, Scotland, Sept. 1991. Springer-Verlag.

[9] H. D. Tagare, D. O'Shea, and A. Rangarajan. A geometric criterion for shape-based non-rigid correspondence. In 5^{th} *International Conference on Computer Vision*, pages 434–439, MIT, Cambridge, Massachusetts, June 1995.

[10] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, Apr. 1991.

[11] P. Zhu and P. M. Chirlian. On critical point detection of digital shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):737–748, 1995.