A Method of Automated Landmark Generation for Automated 3D PDM Construction

A. D. Brett and C. J. Taylor
Department of Medical Biophysics
University of Manchester
Manchester M13 9PT, Uk
adb@sv1.smb.man.ac.uk

Abstract

A previous publication has described a method of pairwise 3D surface correspondence for the automated generation of landmarks on a *set* of examples from a class of shape [3]. In this paper we describe a set of improved algorithms which give more accurate and more robust results. We show how the pairwise corresponder can be used in a extension of an existing framework for establishing dense correspondences between a set of training examples [4] to build a 3D Point Distribution Model. Examples are given for both synthetic and real data.

1 Introduction

We are interested in building Point Distribution Models (PDMs) of 3D shapes [6]. This requires dense correspondences to be established between a set of training examples of the shape. Previous publications [3] including some describing methods of surface correspondence [7] [1] have described probable solutions to parts of this problem. Here we describe a completely automated approach which involves extending the previous work and improving the accuracy and robustness of some of the algorithms.

Currently, the construction of a PDM involves the manual identification of a set of L *landmarks* $\{\mathbf{x}_i; 1 \leq i \leq L\}$ for each of N training examples of a class of shapes. A landmark is a point which identifies a salient feature of the shape and which is present on every example of the class. Manual definition of landmarks on 2D shape has proved to be both time-consuming and subjective. The interactive identification of landmarks in 3D images or on pre-segmented surfaces is considerably more difficult and time-consuming than in the 2D case.

Hill *at al* have previously described a method of non-rigid correspondence in 2D between a pair of closed, pixellated boundaries [5] [4]. The method is based on generating sparse polygonal approximations for each shape; no curvature estimation for either boundary was required. Results were presented which demonstrate the ability of this algorithm to provide accurate, non-rigid correspondences. This pair-wise corresponder was used within a framework for automatic landmark generation which demonstrated that landmarks similar to those identified manually were produced by this approach.

A similar framework is the basis of the approach to 3D automatic landmark generation described here, and consists of the the construction of a binary tree of merged shapes.

BMVC 1998 doi:10.5244/C.12.91

Once such a tree has been produced, a set of L_t landmark points may be identified on the root (mean) shape of the tree and the positions of these landmarks propagated out to the N_t leaf (example) shapes. The algorithm used to construct this tree requires a pair-wise corresponder which both matches shapes (so that they can be merged) and measures the quality of the match (in order to decide which pairs to merge).

We demonstrate the production of a densely triangulated mean shape from a set of smooth synthetic examples by the correspondence of their sparse polyhedral approximations and the production of a binary tree of merged shapes. A 3D PDM constructed using these landmarks is demonstrated by displaying modes of shape variation. The ability of the method to build a model of a complex biological shape is also demonstrated using examples of the left ventricle of the brain.

2 Polyhedral-Based Correspondence

The pair-wise correspondence algorithm previously described [3] comprised two stages:

1. Generation of sparse polyhedral approximations $\mathbf{A}''$ and $\mathbf{B}''$ of the input shapes $\mathbf{A}$ and $\mathbf{B}$ by triangle decimation, for which $\{\mathbf{A}''_i\} \subset \{\mathbf{A}_i\}$ and $\{\mathbf{B}''_i\} \subset \{\mathbf{B}_i\}$. The connectivity descriptions of vertices in these polyhedra are updated during the decimation process.

2. Generation of a corresponding pair of sparse polyhedra $\mathbf{A}'$ and $\mathbf{B}'$. This is accomplished using a global Euclidean measure of similarity between both the sparse polyhedron $\mathbf{A}''$ and a subset of labelled vertices from $\mathbf{B}$ and between $\mathbf{B}''$ and a subset from $\mathbf{A}$. The connectivity and number of vertices from either $\mathbf{A}''$ or $\mathbf{B}''$ is chosen to define the polyhedra $\mathbf{A}'$ and $\mathbf{B}'$ from the labelled vertices $\{\mathbf{A}'_i\} \subset \{\mathbf{A}_i\}$ and $\{\mathbf{B}'_i\} \subset \{\mathbf{B}_i\}$.

Our adaptation of the sparse polygon generation algorithm still makes use of a decimation method described by Schroeder *et al* [8]. However, we now use a *distance* metric which preserves sharp edges and thin structures, the volume metric used previously was less accurate in this respect. This distance metric still allows us to treat all of the vertices of a mesh uniformly and to dispense with the vertex characterisation step of Schroeder's algorithm.

The distance metric is computed using Schroeder's distance to mean plane measure. The mean plane associated with a vertex $\mathbf{v}_0$ is defined by a unit normal $\hat{\mathbf{u}}$ and weighted centroid $\overline{\mathbf{x}}$ which are defined using the $n_{\mathbf{v}_0}$ triangles connected to $\mathbf{v}_0$:

$$\mathbf{u}_0 = \frac{\sum_{i=1}^{n_{\mathbf{v}_0}} \hat{\mathbf{n}}_i A_i}{\sum_{i=1}^{n_{\mathbf{v}_0}} A_i}, \quad \hat{\mathbf{u}_0} = \frac{\mathbf{u}_0}{\|\mathbf{u}_0\|}, \quad \overline{\mathbf{x}}_0 = \frac{\sum_{i=1}^{n_{\mathbf{v}_0}} \mathbf{x}_i A_i}{\sum_{i=1}^{n_{\mathbf{v}_0}} A_i} \tag{1}$$

where $\mathbf{x}_i$, $\hat{\mathbf{n}}_i$ and A_i are respectively the centroids, unit surface normals and areas of triangle $i (1 \leq i \leq n_{\mathbf{v}_0})$ connected to $\mathbf{v}_0$. The distance metric, D, is computed as:

$$D(\mathbf{v}_0) = |d(\mathbf{v}_0) - d'(\mathbf{v}_0)| \tag{2}$$

where $d(\mathbf{v}_0)$ and $d'(\mathbf{v}_0)$ are the *signed* distances of the vertex $\mathbf{v}_0$ to the mean plane of the loop before and after decimation i.e. $d(\mathbf{v}_0) = \hat{\mathbf{u}} \cdot (\mathbf{v}_0 - \overline{\mathbf{x}})$. By decimation it is meant that the vertex associated with the loop has been removed and the resulting hole re-triangulated. Re-triangulation of the hole is by a recursive loop-splitting algorithm which seeks to fill the hole with triangles of low aspect ratio. The results of running this decimation algorithm on a densely triangulated surface representing a left lateral ventricle of the brain can be seen in Figure 1, notice the preservation of the thin edge structure at bottom left of the ventricle.

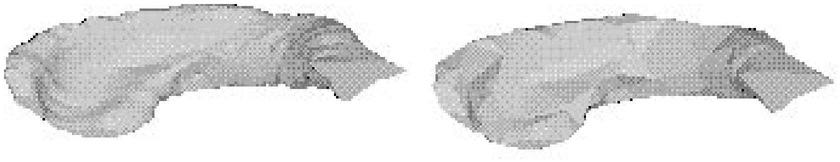

Figure 1: Result of applying the decimation algorithm to a triangulated surface of the left ventricle of the brain. On the left is a shaded representation of the original dense triangulation with approximately 2000 vertices. On the right the same surface represented by 200 vertices (decimated by 90%).

2.1 Correspondence via Euclidean Transformation

As in [3], correspondences are established using a symmetric version of the ICP algorithm [2] which determines a Euclidean transformation Q between the corresponded examples to satisfy:

$$\text{Min}\, E_{\mathrm{S}}^2 = \frac{1}{n_{\mathbf{A}''}} \sum_{i=1}^{n_{\mathbf{A}''}} \|Q(\mathbf{A}''_i) - Q^{-1}(\mathbf{B}'_i)\|^2 + \frac{1}{n_{\mathbf{B}''}} \sum_{j=1}^{n_{\mathbf{B}''}} \|Q(\mathbf{A}'_j) - Q^{-1}(\mathbf{B}''_j)\|^2 \quad (3)$$

where $Q(\mathbf{p}) = s\mathbf{R}\mathbf{p} + \mathbf{t}$, s is a scale factor, $\mathbf{R}$ is a rotation matrix and $\mathbf{t}$ is a translation. A single corresponding pair of sparse polyhedra must be now established from the two polyhedron/pointset pairs, $(\mathbf{A}'', \{\mathbf{B}'_i\})$ and $(\mathbf{B}'', \{\mathbf{A}'_i\})$. The choice of connectivity is not critical and so we choose the connectivity of $\mathbf{A}''$ (the binary tree framework causes the error of the match to be tested for both pair orderings). This choice defines one of the corresponding sparse polyhedra $\mathbf{A}' \equiv \mathbf{A}''$. However, the choice of a single connective description imposes a choice about the polyhedron/pointset pair used to establish correspondence, in this case, $(\mathbf{A}'', \{\mathbf{B}'_i\})$. This does not guarantee a good *representation* of shape $\mathbf{B}$ by the pointset $\{\mathbf{B}'_i\}$. At present we use the cost :

$$E = \lambda E_{\mathrm{S}} + E_{\mathrm{R}} \quad (4)$$

where

$$E_{\mathrm{R}}^2 = \frac{1}{n_{\mathbf{A}''}} \sum_{i=1}^{n_{\mathbf{A}''}} \underset{j}{\text{Min}} \|Q(\mathbf{A}_j) - Q(\mathbf{A}'_i)\|^2 + \frac{1}{n_{\mathbf{B}''}} \sum_{k=1}^{n_{\mathbf{B}''}} \underset{l}{\text{Min}} \|Q^{-1}(\mathbf{B}_l) - Q^{-1}(\mathbf{B}'_k)\|^2 \quad (5)$$

to determine which of the pair orderings to use for the match. As in the 2D algorithm [4], we set $\lambda = 0.2$, but have not found this value to be critical.

The connectivity description of $\mathbf{A}''$ is now combined with the pointset $\{\mathbf{B}'_i\}$ to produce a pair of matching polyhedra with a one-to-one mapping $(\mathbf{A}' \mapsto \mathbf{B}')$. The number of correspondences n_Φ has now been determined: $n_\Phi = n_{\mathbf{A}'} = n_{\mathbf{B}'}$.

3 Merging Shapes

Given a pair of corresponding sparse polyhedra $\mathbf{A}'$ and $\mathbf{B}'$, a local surface parameterisation is used to interpolate a dense set of vertices on each. The local surface parameterisation is of a single sparse triangle, and is produced by a parameterisation of the three surface paths corresponding to its three edges. If the triangles of the two sparse polyhedra correspond, then so will the interpolated vertices within them. The connectivity of these vertices is defined by the interpolation. The interpolated vertices of this dense corresponding pair may then be merged by combining the geometric information from both, with the connectivity description which is identical in each.

3.1 Parameterisation of Surface Paths

The accurate and robust parameterisation of surface paths between the vertices of a dense triangulated surface has proved to be the key to the use of this landmarking framework in 3D. It has also proved to be one of the most challenging problems.

As in the previous work, a ‘brushfire’ type distance transform algorithm is used to march the path across the surface between dense edges of the triangulation. At each stage, the minimisation of a cost $C_i(\mathbf{y}_0)$ locates the best next point for the surface path on edge $\mathbf{y}_i$ attached to $\mathbf{y}_0$. A point $\mathbf{x}$ on edge $\mathbf{y}_i$ can be expressed parametrically by:

$$\mathbf{x} = \mathbf{v}_1 + s(\mathbf{v}_2 - \mathbf{v}_1), \quad s \in [0,1] \tag{6}$$

where $\mathbf{v}_1, \mathbf{v}_2$ are the vertices on the mesh which define the edge $\mathbf{y}_i$. We consider not just the path $(\mathbf{a}, \mathbf{b})$ which is a sparse polyhedral edge, but also the dense polyhedral triangles $\mathbf{t}_1$ and $\mathbf{t}_2$ connected to the dense edge under consideration, see Figure 2.

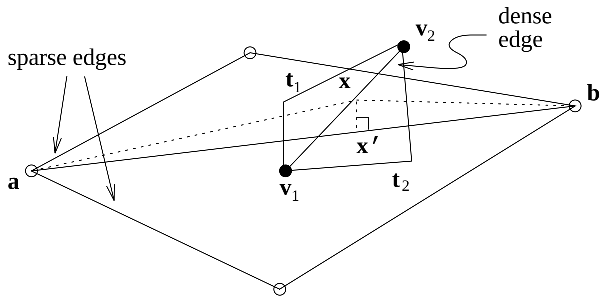

Figure 2: The cost function used to produce surface paths is defined in terms of a pair of dense polyhedral triangles connected to the dense edge under consideration.

We construct a plane normal to the surface defined by the reference point $\mathbf{c} = (\mathbf{a} + \mathbf{b})/2$ and by the unit normal $\hat{\mathbf{n}}_\mathrm{c}$, where $\hat{\mathbf{n}}_\mathrm{c} \cdot (A_1\hat{\mathbf{n}}_1 + A_2\hat{\mathbf{n}}_2) = 0$, in which A_1 and A_2 are the areas of the triangles $\mathbf{t}_1$ and $\mathbf{t}_2$ respectively, and $\hat{\mathbf{n}}_1$ and $\hat{\mathbf{n}}_2$ are the unit normals to these triangles. The point of intersection of this plane and the line defined in Equation 6 is found. The cost function of igniting an edge $\mathbf{y}_i$ from edge $\mathbf{y}_0$ is defined only for the intersection of the line segment given by $s \in [0, 1]$:

$$C_i(\mathbf{y}_0) = (\|\mathbf{a} - \mathbf{x}'\|^2 + \|\mathbf{b} - \mathbf{x}'\|^2)/\|\mathbf{a} - \mathbf{b}\|^2 \tag{7}$$

where $\mathbf{x}'$ is the projection of $\mathbf{x}$ on the sparse edge $(\mathbf{a} - \mathbf{b})$. This cost constrains the surface path to lie within the line defined by $\mathbf{a}$ and $\mathbf{b}$, thus preventing it from looping back around the entire surface of the shape.

3.2 Producing a Mean Shape

The connectivity of $\mathbf{A}'$ and $\mathbf{B}'$ are identical. Therefore, we can correspond the individual sparse triangles of the polyhedra. A dense set of corresponded sampling points may then be generated within each pair of triangles. First, the three surface paths corresponding to the edges of a sparse triangle are extracted using the method of the previous section. Next, three new vertices are defined at the midpoints of these paths. The triangle is now split into four new sub-triangles. This process is applied to all the triangles of $\mathbf{A}'$ and $\mathbf{B}'$ recursively to some *depth* to produce the dense triangulation $\mathbf{A}'_\mathrm{d}$ and $\mathbf{B}'_\mathrm{d}$. Now a densely triangulated mean shape may be generated by averaging the geometric information of these dense triangulations to produce a pointset $\{\mathbf{C}_i\}$ and this is combined with the connectivity from $\mathbf{A}'_\mathrm{d}$ to produce a densely triangulated polyhedron $\mathbf{C}$.

4 Automated Landmarking

The pairwise corresponder described above is used to build a binary tree of merged shapes with a single mean shape at the root and the examples from the training set at the leaves. We produce a set of landmarks $\{\mathbf{C}_{\mathrm{l},i}\} \subset \{\mathbf{C}_\mathrm{i}\}$ on the mean shape by decimation to the required number of landmark points. The connectivity of these points is defined by the sparse polyhedron $\mathbf{C}_\mathrm{l}$. These landmark points are then propagated down the branches of the tree to the original example shapes at the leaves.

At each branch of the tree, each of the landmark points can be projected onto a triangle of the sparse version of the mean shape $\mathbf{C}'$ which is the mean of $\mathbf{A}'$ and $\mathbf{B}'$. The projection is along the normal to the sparse triangle associated with the landmark point, which is chosen from the dense result of splitting that triangle. The sparse triangle is then parameterised along a baseline and a vector between the baseline and opposite vertex, see Figure 3. The projection $\mathbf{e}$ on the triangle $(\mathbf{a}, \mathbf{c}, \mathbf{b})$ is now uniquely defined by the parametric pair (t, u).

There is a correspondence between the vertices of this sparse triangle $(\mathbf{a}, \mathbf{c}, \mathbf{b})$ on $\mathbf{C}'$ and the vertices of a pair of sparse triangles on $\mathbf{A}'$ and $\mathbf{B}'$. For simplicity, we will deal only with the mapping to $\mathbf{A}'$. Call the sparse corresponding vertices on $\mathbf{A}'$, $(\mathbf{a}', \mathbf{c}', \mathbf{b}')$. The projection point $\mathbf{e}$ can therefore be mapped onto the sparse triangle $(\mathbf{a}', \mathbf{c}', \mathbf{b}')$, by parameterising it in t at u to give $\mathbf{e}'$.

We must now reconstruct a point on the *surface* of $\mathbf{A}$ which corresponds to the projection point $\mathbf{e}'$ mapped onto a sparse triangle. We do this by first constructing surface

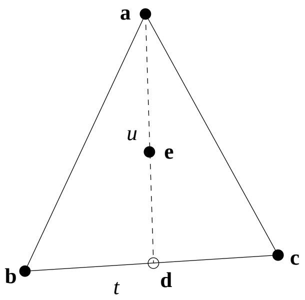

Figure 3: Sparse triangles are parameterised along the baseline (t) and along a vector between the baseline and opposite vertex (u) of a sparse triangle to define the position of point $\mathbf{e}$.

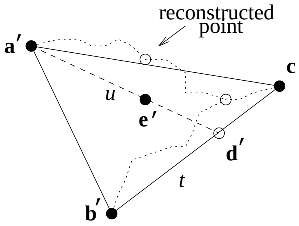

Figure 4: Projected points are reconstructed on dense surfaces by the parameterisation of surface paths constructed across the baseline and from the opposite vertex of a sparse triangle.

paths using the method of section 3.1. First, we construct a surface path between $\mathbf{b}'$ and $\mathbf{c}'$. By parameterising this surface path, we can find an approximation to the position of $\mathbf{d}'$ reconstructed on the surface of the dense triangulation $\mathbf{A}$. We now construct another surface path from $\mathbf{a}'$ to the reconstruction of $\mathbf{d}'$. Parameterisation of this second surface path gives an approximate position for the reconstruction of $\mathbf{e}'$ on the dense surface, see Figure 4. Finally we choose the landmark on the dense surface as the vertex with smallest Euclidean distance to the reconstruction of $\mathbf{e}'$. This process is repeated for both shapes $\mathbf{A}$ and $\mathbf{B}$ that contributed to the mean $\mathbf{C}$.

The choice of a vertex from the dense surfaces of the two contributing shapes provides us with a set of accurate landmarks which are a subset of the densely triangulated shapes. Starting at the root, we can repeat the two mapping procedures at each level of the tree of merged shapes until the leaf level of example shapes is reached. The connectivity of each set of landmark points is that of the mean shape and is propagated through the tree.

5 Results

5.1 Tree of Merged Shapes

In order to test the pair-wise correspondence and merging algorithms, we have constructed a binary tree of merged shapes, and used this to produce a set of landmarks for a 3D PDM. The shapes are six smooth synthetic examples of ellipsoids with varying aspect ratios. Each shape has ≈ 1000 vertices. Such shapes are difficult to match and merge because there are no obvious landmarks (salient features). In addition, any errors introduced during the construction of dense merged mean shapes will be indicated by prominent features on these surfaces which are expected to be smooth.

The tree of merged shapes is shown in Figure 5 as a series of shaded and rendered triangulated surfaces. The first level shows six input example shapes. The second and third levels show the means of the shapes to the left and right in the levels below. Level 4 is the merged mean of the two shapes in level 3 and represents the mean shape of the six input examples. In each case, the original triangulated surface was decimated to 10%

of the original number of vertices in the example shapes to produce the sparse polyhedral representation during matching.

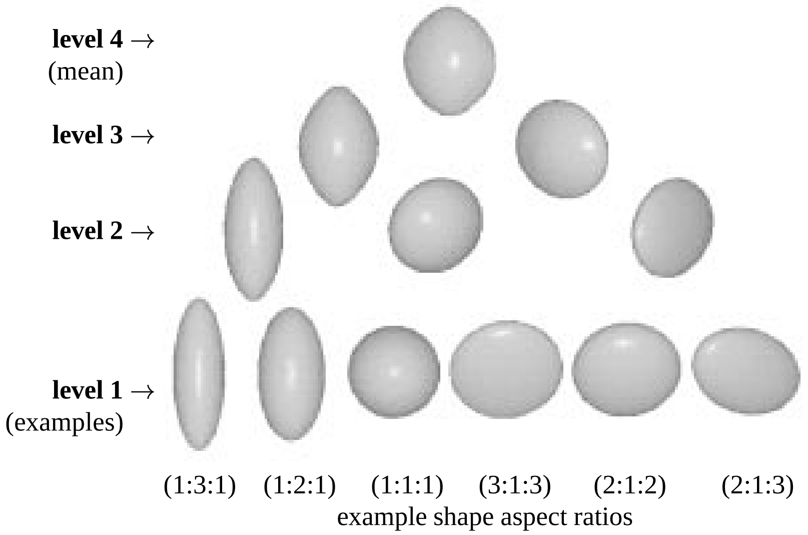

Figure 5: A merge tree of six synthetic examples of varying aspect ratios. Level 1 shows the six original example shapes which are merged to produce the three densely triangulated mean shapes of level 2. These shapes have been merged to produce the mean shapes of level 3, and finally the single mean shape of level 4. At each level of the tree, the shape is a mean of the two shapes to the left and right in the level below.

The pair-wise correspondence and merging algorithms have proved to be computationally tractable - the matching of two shape surfaces (≈ 1000 vertices) using a decimation of 90% for the sparse polyhedral representations takes around 15 CPU seconds on a Sun UltraSPARC 2. The merging algorithm takes a further 30 CPU seconds to produce a densely triangulated mean from the resulting matched sparse polyhedrons.

5.2 A 3D Point Distribution Model

We have constructed a 3D Point Distribution Model of the six smooth synthetic shapes described in the previous section. Each example shape consisted of 300 landmark points. An instance of the model $\mathbf{x}_i = \{x_{i,1}, y_{i,1}, z_{i,1}, \ldots, x_{i,n}, y_{i,n}, z_{i,n}\}$ can be generated by:

$$\mathbf{x}_i = \overline{\mathbf{x}} + \mathbf{P}\mathbf{b} \tag{8}$$

where $\overline{\mathbf{x}}$ is the mean shape, the columns of $\mathbf{P}$ are a set of mutually orthogonal 'modes' of shape variation, and $\mathbf{b}$ is a vector of shape parameters. The proportion of shape variation explained by each shape parameter is shown in Table 1. These numbers are calculated as the proportion of total shape variance that is projected onto each of the vectors in $\mathbf{P}$. The shape variance is measured in the frame of the normalised mean shape aligned to each of the sets of landmarked shapes by minimum Euclidean distance.

The first two shape parameters of this model explain over 90% of the total shape variation indicating a compact model. The shape parameters $b_3 \ldots b_5$ appear to be due to

mode of variation	variation explained / %
b_1	66.6
b_2	25.5
b_3	3.0
b_4	2.8
b_5	2.1

Table 1: The proportion of total shape variation explained by each of the 5 modes of shape variation of the 3D PDM constructed from six smooth synthetic shapes.

noise in the placement of landmarks. A set of instances of the model with the first two shape parameters set to $-1.5\,\sigma_i \leq b_i \leq +1.5\,\sigma_i$ from the mean shape, where σ_i is the standard deviation of b_i over the training set, is shown in Figure 6. The geometric shapes are shown in both plan and front face view.

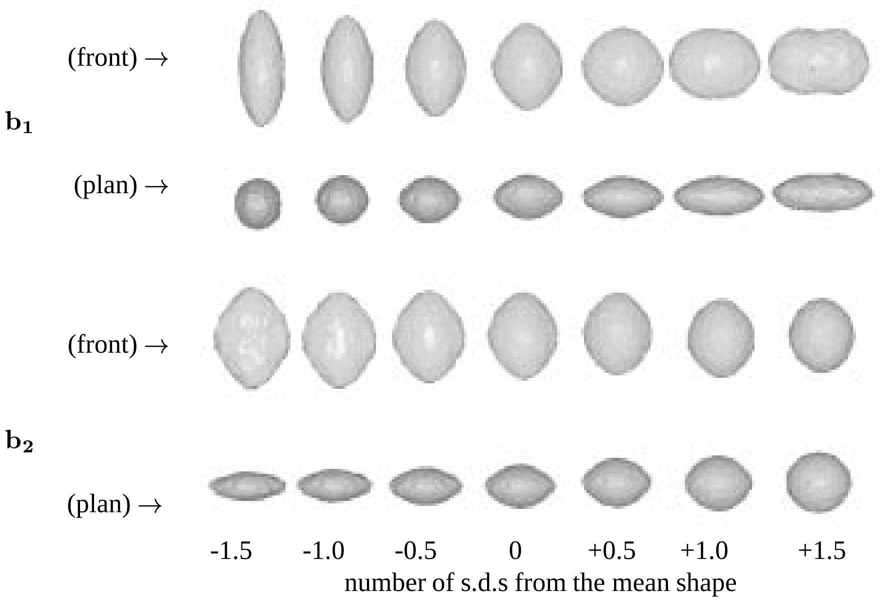

Figure 6: Shape instances generated using the 3D PDM of six smooth synthetic shapes. The instances are generated by varying a single shape parameter, holding all others constant at 0 s.d.s from the mean shape. Each instance of the shape model consists of 300 points.

We have also generated a 3D PDM from four complex biological shapes - left ventricles of the brain, see Figure 7. These have been defined by hand as contours on a series of 2D slices from 3D Magnetic Resonance images. The ventricles of the brain have a complex structure and vary significantly between individuals in both their size and shape. The example shapes consisted of $\approx$ 2000 vertices, upon which were placed 200 landmark points. The first two modes of variation of this model are illustrated in Figure 8, b_1

explains 54.0 % of the total variation, and b_2 explains 25.2 %.

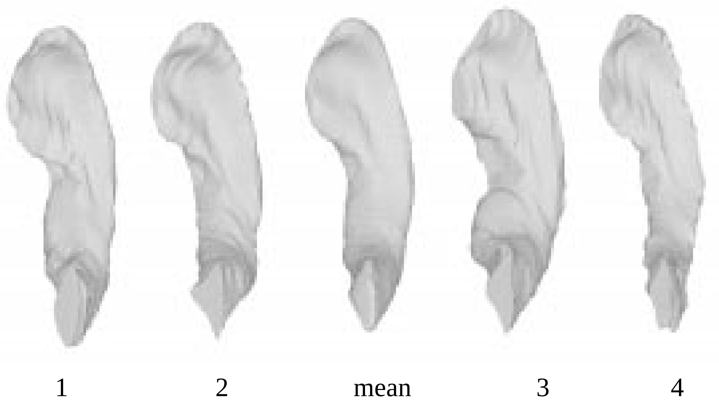

Figure 7: The four left brain ventricle examples used to generate a 3D PDM and their densely triangulated mean.

Upon inspection, we can see that the shape instances generated are legal and accurately reflect the shape variation present in the training set of four examples - indicating an accurate automated placing of landmark points. There are some errors in the reconstruction of shape instances in the upper lefthand edge of the shape, along a very thin edge present in each of the examples. We believe these are due to illegal (crossing) correspondences generated by the ICP algorithm in an area of high triangle density.

6 Conclusions

We have presented a novel method for the correspondence of two faceted (triangulated) surfaces. The method is based on the production of a sparse polyhedral representation of one shape and matching this to a sparse pointset representation of the other. Qualitative results have been presented which show that the algorithm can produce accurate non-rigid correspondences between pairs of smooth synthetic shapes in the production of a binary tree of merged shapes. Such a tree provides a framework for the automated landmarking of the input example shapes necessary for the production of a 3D PDM. A method for the accurate propagation of landmarks from the root to the leaf shapes of such a tree has been described. We have presented sets of shape instances generated from a 3D PDM built from such a set of propagated landmarks. The model has been shown to be compact and to generate legal instances of the set of smooth shapes. We have also produced a model from complex biological shapes and shown that this model reconstructs legal shape instances.

At present, we do not combine the two sets of correspondence pairs produced by the symmetric ICP approach. This would require the combination of two connectivity descriptions, one from each sparse shape. This leads to our use of a representation error and control parameter λ. One focus of our current research is the combination of these two connective descriptions to eliminate the additional representation error and the trapping of illegal correspondences generated by the ICP algorithm.

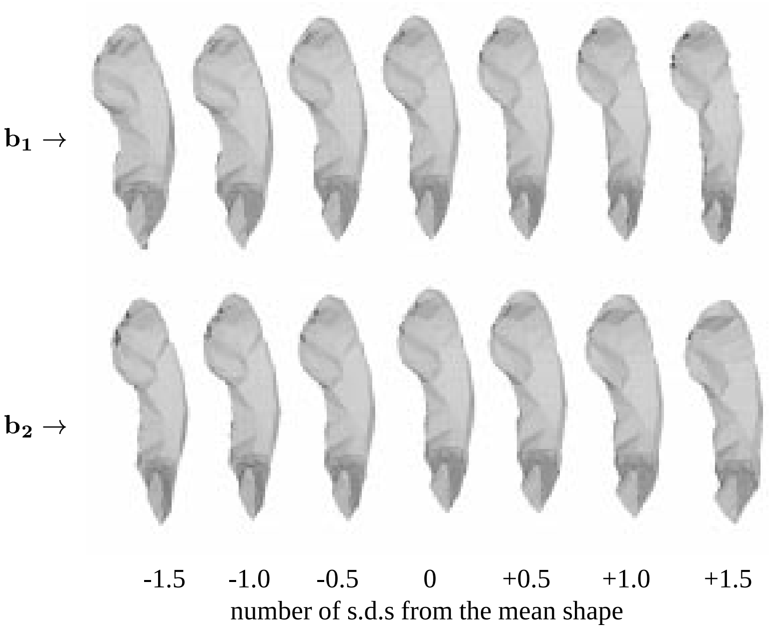

Figure 8: Shape instances generated using a 3D PDM of four left brain ventricles. The model consists of 200 points.

References

[1] A. Benayoun, N. Ayache, and I. Cohen. Adaptive meshes and nonrigid motion computation. In 12th *International Conference on Pattern Recognition*, pages 730–732, Jerusalem, Israel, 1994.

[2] P. J. Besl and N. D. McKay. A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[3] A. D. Brett, A. Hill, and C. J. Taylor. A method of 3D surface correspondence for automated landmark generation. In 8th *British Machine Vison Conference*, pages 709–718, Essex, England, Sept. 1997.

[4] A. Hill, A. D. Brett, and C. J. Taylor. Automatic landmark identification using a new method of non-rigid correspondence. In J. Duncan and G. Gindi, editors, 15th *Conference on Information Processing in Medical Imaging*, pages 483–488, Poulteney, VT, 1997. Springer-Verlag.

[5] A. Hill and C. J. Taylor. A method of non-rigid correspondence for automatic landmark identification. In 7th *British Machine Vison Conference*, pages 323–332, Edinburgh, Scotland, Sept. 1996. BMVA Press.

[6] A. Hill, A. Thornham, and C. J. Taylor. Model-based interpretation of 3D medical images. In J. Illingworth, editor, 4th *British Machine Vison Conference*, pages 339–348, Guildford, England, Sept. 1993. BMVA Press.

[7] 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.

[8] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen. Decimation of triangle meshes. *Computer Graphics*, 26(2):65–70, 1992.