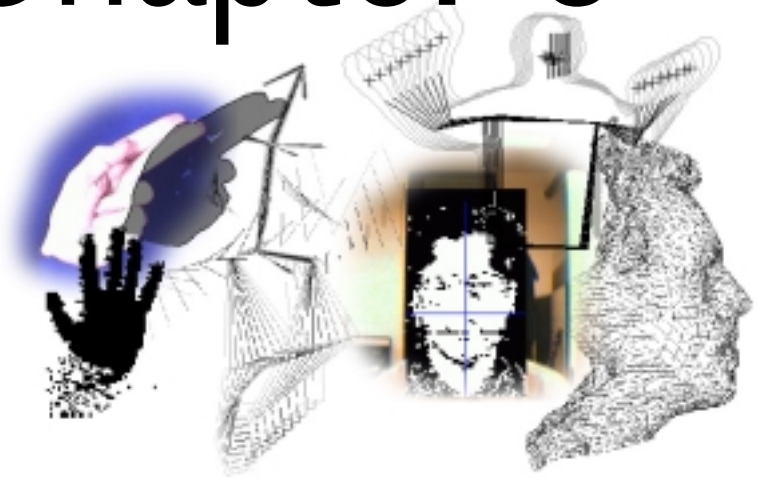
Chapter 8



8 3D Point Distribution Models

8.1 Introduction

It has thus far been demonstrated how a Point Distribution Model can be constructed for a 2D contour or shape (Chapter 3) and grey scale images (Section 5.5.2). Chapter 7 introduced a simple 3D PDM in the form of a stick human figure. This chapter will extend upon this to 3D eigensurface models which are constructed from polygonal surface representations and are the analogous extension into 3D of the 2D contour.

For a 2D contour, consisting of *n* points, a training example **x** is constructed by concatenating the constituent points of the contour into a single *2n* vector $\mathbf{x} \in \Re^{2n}$. As was shown in section 7.2, for 3D the procedure follows a similar procedure. Each point of the model differs only in its dimensionality. Therefore a 3D model consisting of *m* points (vertices) will form a vector $\mathbf{x} \in \Re^{3m}$. In chapter 7, where the 32 points consisted of key-points of a simple human skeletal model, this produced a 96 dimensional vector. However, more realistically the target data represents a surface, where each vertex of the surface represents a key point within the model. This results in extremely high dimensional spaces i.e. for a 3D mesh of 100 x 100 points, $\mathbf{x} \in \Re^{30000}$. Under these conditions it is often the

case that the number of training examples is less than the dimensionality of **x**, and hence technique 2 for PCA (detailed in section 3.2.5) is invaluable in the construction of 3D PDMs.

Although the construction of 3D PDMs is a simple extension to the 2D case, one of the major problems associated with their construction is the acquisition of training data and its alignment. Due to the complexity of constructing 3D surfaces by hand, automated procedures are essential. As has been discussed in chapter 8, many techniques such as isosurfacing produce complex discontinues surfaces which are unsuitable for statistical analysis. These 3D surfaces must be aligned and resampled in a similar manner to the 2D contour. However, the problem is compounded by high dimensionality and the resulting computational complexity of the procedure.

Section 8.2 demonstrates the construction of a 3D PDM using a synthetic drinking glass example. Sections 8.3 will show how this can be extended to real data and describe approaches to the resampling and alignment problem in 3D. This will be demonstrated by a 3D PDM of a human head. Finally conclusions will be drawn.

8.2 The Eigen Glass Model

8.2.1 Introduction

Point Distribution Models attempt to model the deformation of a class of objects or shapes with simple statistical analysis. The example shown here is that of a class of drinking vessels. This synthetic example data provides a data set with which to explore the construction of 3D PDMs and will be used in chapter 10 as an example for statistical inference.

8.2.2 Constructing the Training set

The eigen Glass training set consists of 7 types of glass shape (see Figure 8.2.1). Each example was created by sweeping a 2D contour around a central y-axis.

This forms a rotationally symmetric glass of varying shape and size. Since each example was constructed in a similar manner, with the same number of rotational steps and points along the contour, each example contains the same number of vertices.

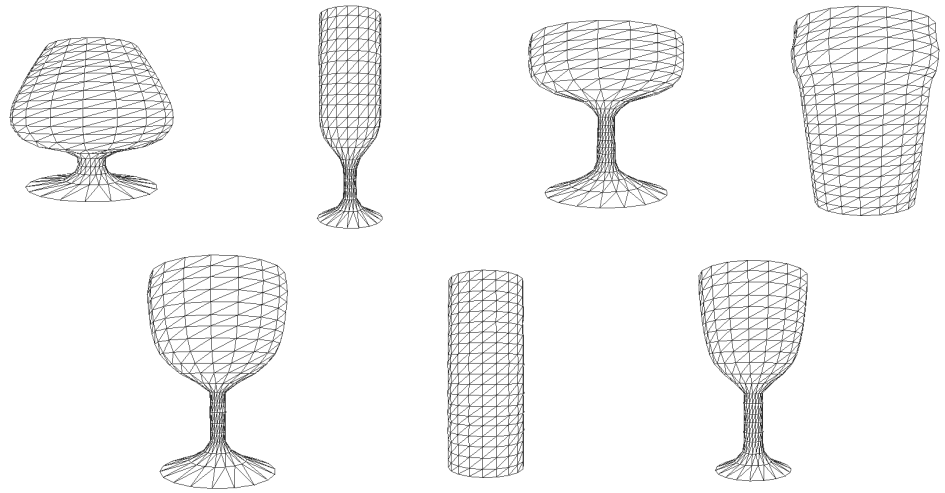


Figure 8.2.1 - Eigen Glass Training Set

The acquisition of the training set provides examples that have a direct correspondence of landmark points and therefore no further alignment or resampling is necessary.

8.2.3 Building the Eigen Model

Each glass example consists of 440 vertices which, when converted to a vector, produces a training example $\mathbf{x} \in \Re^{3n} \Rightarrow \Re^{1320}$. As there are only seven examples in the training set, technique 2 (section 3.2.5) results in a large computation saving during shape analysis. The use of this technique allows decomposition to be performed upon a 7x7 matrix. This produces a significant computational saving over performing a full decomposition upon the 1320 x 1320 covariance matrix.

Figure 8.2.2 demonstrates the primary 3 modes of variation of the resulting 3D PDM rendered in wire frame with hidden line removal. The primary mode is also shown in Gouraud shaded form. The maximum number of modes of deformation for the model is 6 (ie. 100% of the deformation present within the training set is

contained within the first 6 eigenvectors). This is because the number of eigenvectors can never exceed *N-1*, where *N* is the number of training examples. In fact, 99% of the deformation is contained within the primary 4 modes of variation.

This high reduction of the shape space is similar to that shown in earlier cases. However, it is important to note that, due to the rotational symmetry of each of the objects, the training examples contain no additional information after the contours had been swept into a 3D surface. The model could equally have been constructed by performing PCA upon the original contours and sweeping the reconstructed contour, generated from the PDM, around the central axis. This is demonstrated in Figure 8.2.3 where PCA has been performed upon the contours and the resulting 2D PDM extracted.

If Figure 8.2.3 is compared to Figure 8.2.2, it should be apparent that the deformation contained in the modes of variation of the 2D PDM are exactly the same as those of the 3D object. Since both models contain the same information the resulting PDMs have the same characteristics with a total of 7 modes where the first 4 encompass 99% of the deformation. The redundant dimensionality introduced when the contour is swept into a 3D surface does not introduce any additional information and this additional dimensionality is disregarded by PCA demonstrating that both models lie upon the same dimensional sub space.

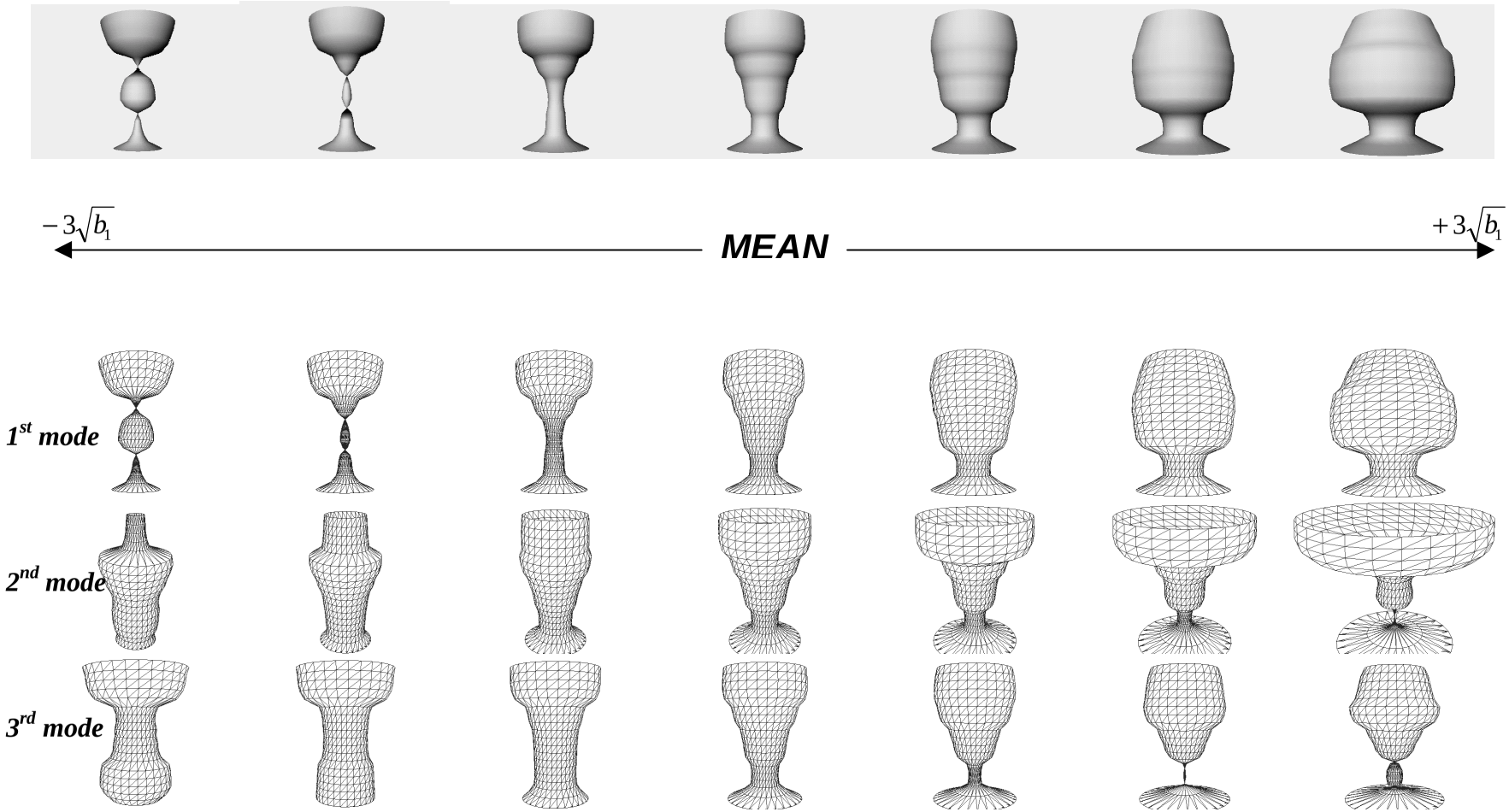
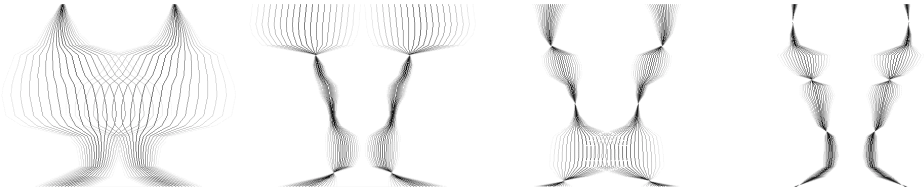


Figure 8.2.2 - The Primary Modes of the eigenGlass Model



1st mode *2nd mode* *3rd mode* *4th mode*

Figure 8.2.3 - The Primary Modes of the 2D eigenGlass Model

8.3 Resampling Meshes

8.3.1 Mesh Alignment

In the previous synthetic eigenGlass demonstration, the simplicity of construction was due to the direct correspondence of landmark points throughout the training set and the artificial way in which it was created. However, this is seldom the case and to ensure the construction of a PDM is successful, careful alignment and resampling must be performed to provide a good correspondence of landmark points between examples.

As with the 2D contour, to ensure a good correspondence between training examples each must be aligned. Techniques like those presented by Cootes *et al* [Cootes 95] for 2D alignment become infeasible due to the high dimensionality of the models. A similar, but less time consuming, alignment process can be performed by treating it as an optimisation problem, solved using an approach to optimisation such as Simulated Annealing or Genetic Algorithms. Such approaches rely upon a fitness function being formulated which assesses what is a good (optimum) match.

For two meshes $\mathbf{x}$ and $\mathbf{y}$, where
$\mathbf{x} = \left(\mathbf{v}1_1^{xyz}, \mathbf{v}1_2^{xyz}, \ldots, \mathbf{v}1_n^{xyz}\right)$, $\mathbf{y} = \left(\mathbf{v}2_1^{xyz}, \mathbf{v}2_2^{xyz}, \ldots, \mathbf{v}2_m^{xyz}\right)$ and $\mathbf{v}_n^{xyz} \in \Re^3$ is the n^{th} vertex of the mesh, a suitable fitness function to be minimised would be the mean distance between the vertices of each mesh,

$$f = \frac{1}{n}\sum_{i=1}^{n}\min_{j=1}^{m}\left\|\mathbf{v}1_i^{xyz} - \left(M\left(s_x, s_y, s_z, \theta_x, \theta_y, \theta_z\right)\left[\mathbf{v}2_j^{xyz}\right] + t_{xyz}\right)\right\|$$

where s_x is a scaling in x, θ_x is a rotation around x, and $t_{xyz} \in \Re^3$ is a translation vector in Euclidean space.

However, this function must be assessed for each pose $\left(s_{xyz}, \theta_{xyz}, t_{xyz}\right)$ of the model in order to find the optimum mapping of one mesh to another and quickly becomes an unfeasible solution as the size of the mesh increases. In addition to this complexity, the procedure must be repeated for all meshes in the training set.

If known features exist upon the surface and the position of these features can be accurately located (such as large planar segments or areas of high curvature), these features can be used in the fitness function rather than every vertex of the mesh.

The simplest method of alignment is similar to that suggested in Section 3.2.4 where the mesh is treated as a cloud of points in $\Re^3$. The centre of gravity of the cloud, $\mathbf{C}^{xyz}$, can then be calculated and subtracted from each vertex to translate the mesh to the origin, where

Equation 8.3-1
$$\mathbf{C}^{xyz} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{v}_i^{xyz}$$

To normalise the mesh, and hence avoid numerical instability during PCA, each vertex is then scaled by the mean distance of all the vertices from the origin, where

Equation 8.3-2
$$\left|\mathbf{v}_i^{xyz}\right|' = \frac{\mathbf{v}_i^{xyz} - \mathbf{C}^{xyz}}{l(\mathbf{v})} \text{ , and } l(\mathbf{v}) = \frac{1}{n}\sum_{i=1}^{n}\left|\mathbf{v}_i^{xyz} - \mathbf{C}^{xyz}\right|$$

By then performing PCA upon the cloud (as done in 3.2.4) principal moments of the shape and therefore the primary axes can be extracted. Once done, the shape can be projected onto these axes to align the principal moments of the shape with

the axes of Euclidean space. Providing the shape does not vary too extensively this approach provides a fast and simple method for object alignment and scaling.

8.3.2 Nearest Neighbour Resampling

Once all training examples have been aligned, they must be resampled to provide a direct correspondence for each vertex, and the associated connectivity across all training examples. It is also important that each example has the same number of vertices so that all training examples have the same dimensionality.

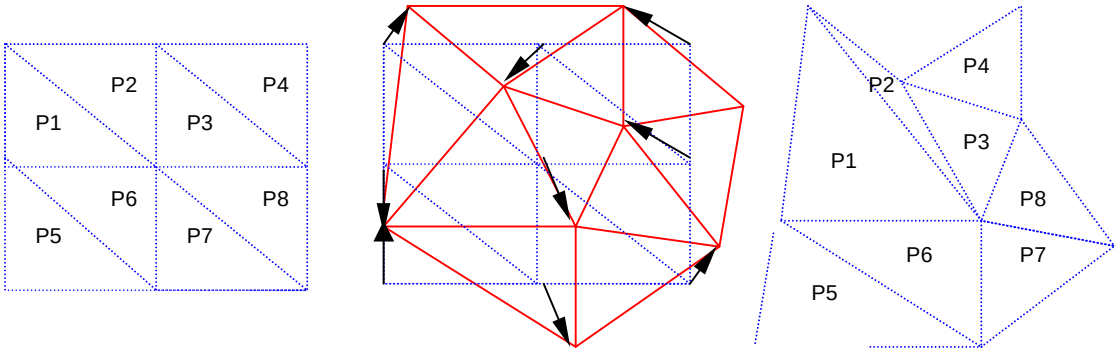


Figure 8.3.1 - Nearest Neighbour Resampling

This can be accomplished by taking a known mesh and deforming it to fit to each example in turn. Figure 8.3.1 demonstrates this procedure using a nearest neighbour approach, a regular mesh (blue) is constructed which has a known number of vertices and connectivity. The regular mesh is then deformed by moving each vertex to the closest vertex of a training example (red) in $\Re^3$. The resulting mesh has the same basic overall shape of the training example but has the connectivity and number of vertices of the regular mesh. This procedure can be repeated for each aligned training example to provide a consistent training set on which statistical analysis can be performed. However, this procedure results in the loss of information as the regular mesh may not contain the local density of vertices required to successfully model high curvature. If the number of polygons is increased further to accommodate this, then unnecessary dimensionality is introduced for areas of low curvature. This approach also introduces problems when mesh elements on the regular mesh are smaller than

those on the training example mesh. Under these circumstances multiple vertices of the resampled mesh may be attracted to a single vertex resulting in polygons of zero area (this will be shown shortly).

Another major disadvantage is that the procedure relies upon the correct alignment of the training examples. If sufficient difference is present between examples then it is possible that vertices will be assigned to completely unrelated features across the training set. This effect can be minimised by utilising the assumption that training examples do not vary extensively between individual examples, although the overall variation may be considerable. Using this assumption a mesh can be deformed to fit a training example and the same mesh applied to the next example until the whole training set has been processed. However, this approach requires user intervention to ensure that an optimum ordering is used for the resampling sequence.

8.3.3 K-nearest Neighbour Resampling

An alternative approach is to use a variation of a clustering algorithm. This results in a consistent mesh with known connectivity, but provides the advantage that vertices on the resampled mesh attempt to best mimic the local features of the surface by averaging the position of the vertices locally.

A mesh $\mathbf{y} = \left(\mathbf{v}1_1^{xyz}, \mathbf{v}1_2^{xyz}, \ldots, \mathbf{v}1_k^{xyz}\right)$ of known connectivity and size *k* is to be fitted to second mesh $\mathbf{x} = \left(\mathbf{v}2_1^{xyz}, \mathbf{v}2_2^{xyz}, \ldots, \mathbf{v}2_m^{xyz}\right)$ of variable size *m*. The vertices of **x** are treated as a cloud of points in $\Re^3$ and the vertices of **y** as exemplars in a k-means algorithm (see Appendix 1). Each vertex of **x** is assigned to an exemplar of **y** in a nearest neighbour sense using the crisp membership function

Equation 8.3-3 $$u_j\left(\mathbf{v}2_i^{xyz}\right) = \begin{cases} 1 & if \left|\mathbf{v}2_i^{xyz} - \mathbf{v}1_j^{xyz}\right| = \min\left|\mathbf{v}2_i^{xyz} - \mathbf{v}1_j^{xyz}\right| \\ 0 & otherwise \end{cases}$$

Each vertex of y is then moved to minimise the distance from its assigned members where

Equation 8.3-4

$$\mathbf{v}1_i^{xyz} = \frac{\sum_{i=1}^{m} u_j\left(\mathbf{v}2_i^{xyz}\right)\mathbf{v}2_i^{xyz}}{\sum_{i=1}^{m} u_j\left(\mathbf{v}2_i^{xyz}\right)} \quad \textbf{and} \quad j = 1,2,\ldots,k$$

This procedure is repeated until the total displacement of the vertices of **x** has dropped below a threshold (i.e. equals zero); at this point the algorithm has converged upon a solution.

8.3.4 K-cluster Elastic Mesh

Both nearest neighbour and k-nearest neighbour approaches are subject to the same problem i.e. the incorrect convergence on local minima. This is largely a problem of model initialisation. Features upon the meshes must be close if a good correspondence is to be achieved as each vertex is only attracted to the closest corresponding point in both techniques. Again, this approach places a large emphasis on the accurate alignment of examples.

This can be overcome to an extent by extending the k-nearest neighbour approach to an elasticised k-cluster approach, which provides the same mechanism for local resampling, but allows global constraints to be placed upon the shape of the mesh.

In addition to the local attraction of the regular mesh to vertices upon the training mesh, elastic properties are added to the connectivity as described in Section 8.5. As the mesh is deformed to fit the training data the elasticity of the mesh attempts to retain as small and as planar a mesh as possible, thus smoothing the mesh and ensuring that the connectivity is preserved.

If the elastic force from section 8.5.2 (equation 8.5-4) is taken and placed in the context of the mesh **y,** the displacement of a node $\mathbf{v}1_i$ from the elastic force is

Equation 8.3-5

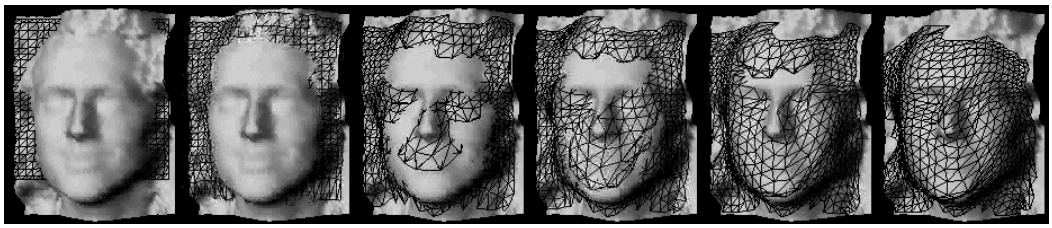
$$\Delta s_i = \frac{\alpha}{n} \sum_{j=0}^{n} r_{ij}$$

where α is the stiffness, $r_{ij} = \mathbf{v}2_j^{xyz} - \mathbf{v}2_i^{xyz}$ the vector separation of two connecting nodes and p is the number of nodes connecting to node $\mathbf{v}1_i$. Combining this force with that of the k-means displacement (Equation 8.3-4) the total movement of a the node $\mathbf{v}1_i$ at each iteration is

Equation 8.3-6

$$\mathbf{v}1_i^{xyz} = \frac{\sum_{i=1}^{m} u_j \left(\mathbf{v}2_i^{xyz}\right) \mathbf{v}2_i^{xyz}}{\sum_{i=1}^{m} u_j \left(\mathbf{v}2_i^{xyz}\right)} - \Delta s_i$$

In order to balance the attraction force and the surface tension of the mesh a weighting parameter which balances the two influences is required. However, the stiffness parameter α can be used for this purpose as it controls the strength of surface tension. This weighting parameter determines the influence of the two forces on the movement of the mesh. When $\alpha = 0$ the mesh operates as the k-nearest neighbour resampling procedure described earlier. When $\alpha \to \infty$ the mesh will not converge on any solution, remaining rigid. Upon initialisation the force is set to allow surface tension to dominate i.e $\alpha = 2$. This parameter and hence the effect of surface tension is decreased at each iteration of the procedure allowing the surface to deform to the data while retaining the constraints of connectivity.



i=0 *i=5* *i=10* *i=15* *i=20* *i=25*

Figure 8.3.2 - Elastic k-cluster mesh

Figure *8.3.2* demonstrates the use of the elastic *k*-cluster mesh technique to resampling a surface of a human head. The shaded surface consists of an irregular mesh consisting of 3896 vertices, which represent the shape of a face. The wire frame mesh is a flat regular tri-mesh of known connectivity and 625 vertices. The flat wire frame mesh is located close to the face mesh and is rendered slightly in front so the shape can be seen as the algorithm iterates. At each iteration α is decreased by 10% and after 25 iterations ($i=25$) the wire frame mesh has deformed to best fit the original face mesh while retaining its connectivity and smoothness. Without this elastic surface tension which smoothes the resulting surface, the mesh would instantly crease and deform as the initial attraction of the k-means algorithm is initially very large. As k-means will only find a local optimum, this initial creasing of the surface remains throughout the fitting. The elasticity ensures that the mesh retains its original shape and connectivity while trying to best deform to resample the mesh.

However, this approach has two major drawbacks

1. The speed of the algorithm is prohibitive, as the computation complexity at each iteration is considerable for even the simplest of surfaces.
2. The rate at which the weighting parameter is decreased is an unknown. Since the rate at which the parameter decreases is responsible for the number of iterations required (and hence the overall speed), an optimum rate must be determined which provides the best time to convergence while allowing the correct convergence on the shape. This is similar to the annealing schedule used in simulated annealing but is beyond the scope of this work.

8.4 3D Head PDM

8.4.1 Constructing the Training set

To illustrate the alignment and construction of a 3D PDM, a model of the human head was built. The head data set consists of 25 surface meshes of varying size and structure acquired using a C3D[10] scanning device. Each mesh has between 4000 and 5000 vertices and differing local mesh densities modelling local

curvature. The examples were first aligned using the alignment procedure outlined in section 8.3 such that each lies within a left handed co-ordinate system with the z-axis is aligned with the direction of the gaze of the face. Once done, each mesh was translated to ensure that the apex of the nose was at the origin. The nose can easily be estimated as the point on the mesh which has the greatest z-value. Each mesh was then normalised to lie within a unit cube as shown in Figure 8.4.1.

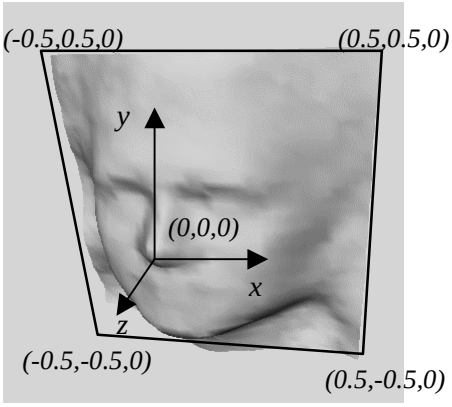


Figure 8.4.1 - Aligning the Face Training Set

Once all the example meshes have been transformed in this way, the next step is to resample each to a uniform mesh structure. A regular triangular faceted mesh was generated as shown in Figure 8.4.2. The regular mesh consists of 1849 vertices and is a unit square with its centre at the origin and aligned with the *x* and *y*-axis.

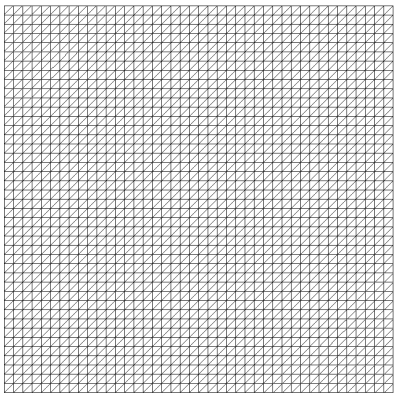


Figure 8.4.2 - Regular tri-mesh

[10] C3D Scanner model courtesy of the Turing Institute, all head models are freely available via the web at http://www.turing.gla.ac.uk

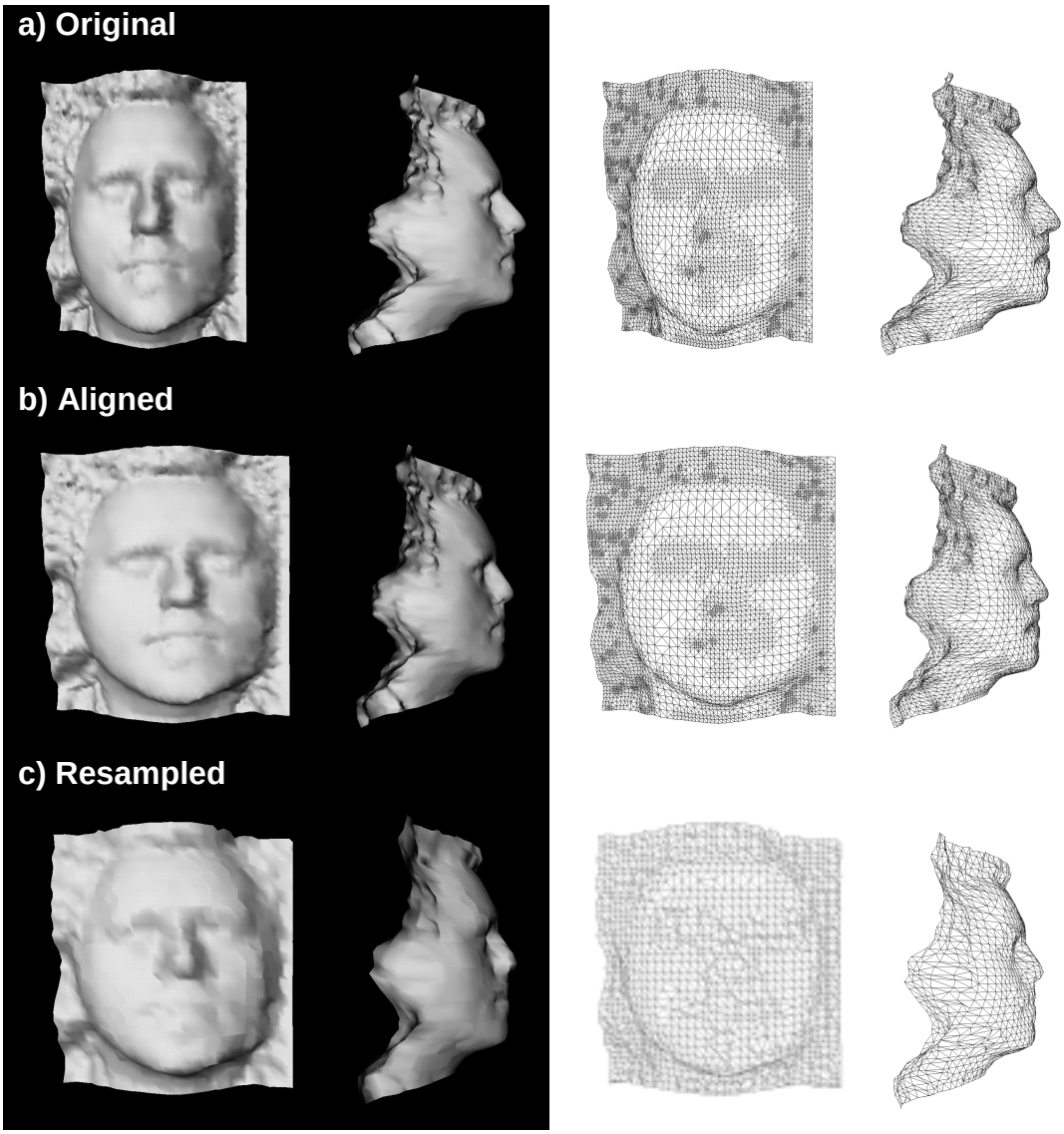


Figure 8.4.3 - Resampling a 3D Mesh

(a) The original mesh (b) The aligned mesh (c) The resampled mesh

For each mesh in turn, the regular mesh is deformed to fit using the nearest neighbour approach described previously. Figure 8.4.2 shows the regular mesh, Figure 8.4.3 (a) the original training example, (b) shows the aligned mesh, and (c) shows the resampled mesh after each vertex has been deformed to fit the example. It should be noted that the final resampled wire frame mesh does not look dissimilar to the original. However, the shaded version shows a step effect to the mesh. This is due to two reasons

1. The local surface density of patches is not optimum to model the curvature hence areas of high curvature have less polygons and consequently a less smooth appearance i.e. the number of vertices has been reduced from around 5000 to 1849.
2. Many polygons have zero area. Where this occurs normal calculations are ill-defined and hence Gouraud shading fails and reverts to a flat shading algorithm.

The problem of zero area polygons, where multiple vertices of the regular mesh have been assigned to a single vertex on the example mesh, is one of the disadvantages that were mentioned in section 8.3. It is not possible to simply remove these polygons as all training examples must have the same dimensionality. A polygon could therefore only be removed if it had zero area in all training examples. However, it will be shown later that the smoothing properties of PCA will remove some of these inaccuracies (see section 8.4.2).

8.4.2 The Face Eigen Model

Upon completion of the resampling procedure a training set is now available on which statistical analysis can be performed. The results of which can be seen in Figure 8.4.4. However, it is difficult to see the overall effect of these modes of deformation except at the extremities of the eigenvectors where the greatest deformation is apparent. Figure 8.4.5 shows the primary 21 eigenvectors corresponding to the 21st largest eigenvalues which encompass 99.998% of the deformation. Each mode is colour coded to represent the deformation. Red, Green and Blue coloured areas represent deformation in x,y and z respectively. The intensity of the image is proportional to the size of the local deformation.

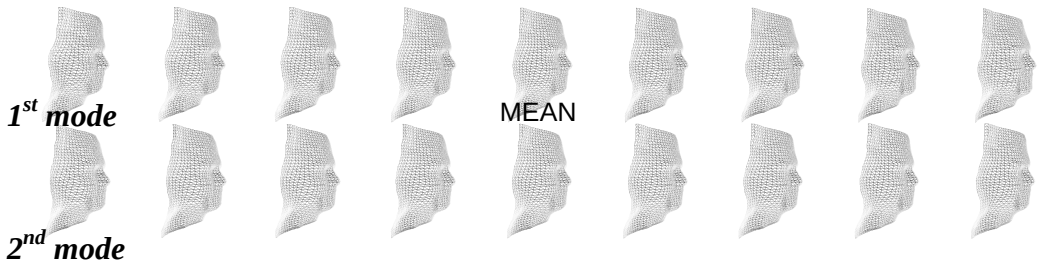


Figure 8.4.4 - Primary two modes of the 3D eigenFace model

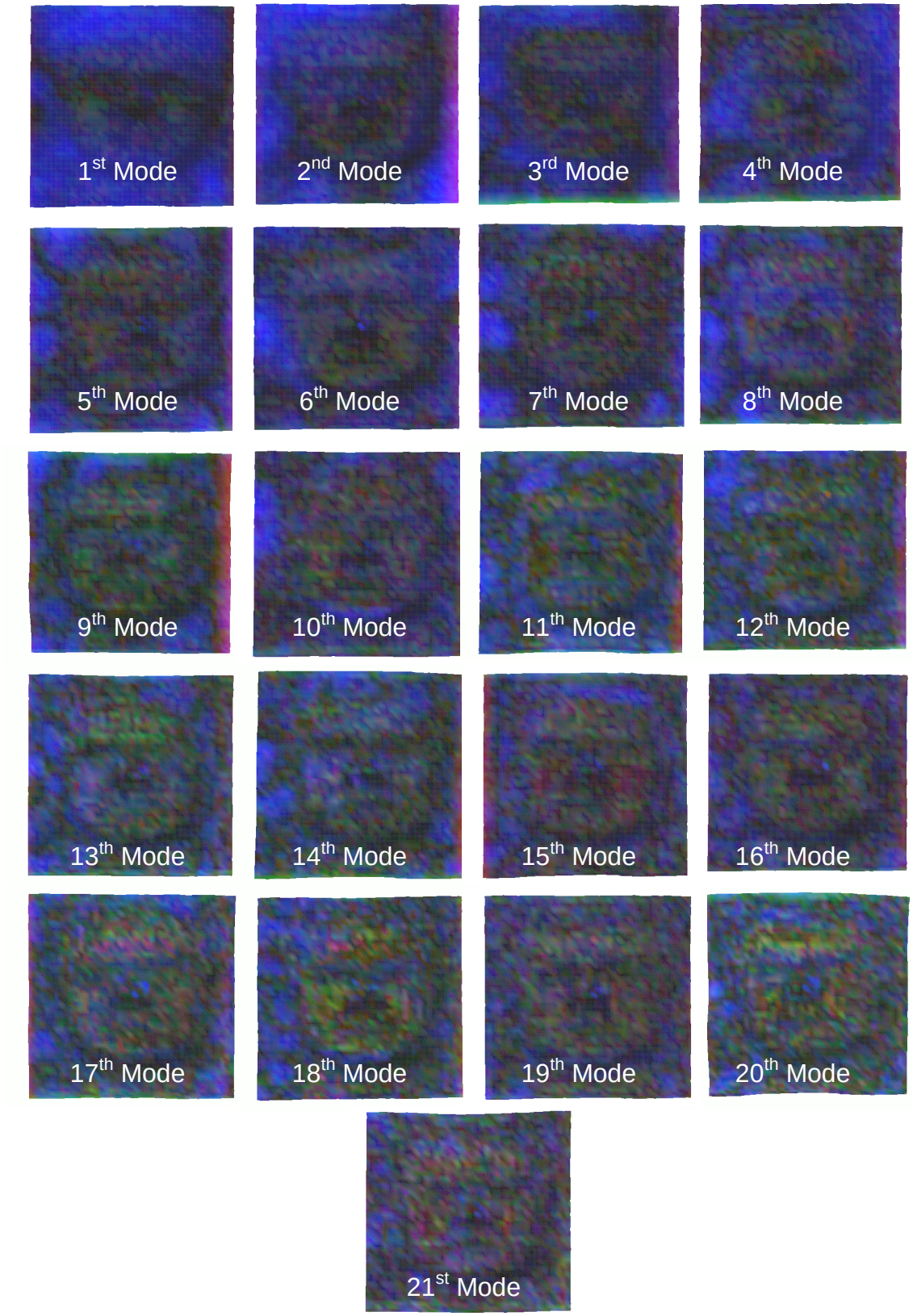


Figure 8.4.5 - Colour map showing deformation of primary modes for eigenFace model

By examining these colour maps it is far easier to infer specific functions for various modes. From the shading on the 6th mode it can be deduced that this mode is responsible for the movement of the eyebrows and cheek areas. The 8th mode however is clearly responsible for the movement of the eyes and mouth. It can be seen that the primary mode contains mainly deformation in 'z' along the top and bottom of the mesh surface. This is due to the large variation in background depth, hair and neck between individual examples. Indeed, the primary modes display large areas of blue showing that they mainly contribute to the depth information of the mesh. As the number of the modes increases a more speckled effect is observed. These effects are the high frequency oscillations, which are typically picked-out by the lower modes of variation. However, much of these high frequency oscillations are due to the nearest neighbour resampling which resulted in zero area polygons.

The original training example mesh size were of the order of 5000 vertices. With 3 dimensions for each vertex this generates examples in a 15000 dimensional space. Resampling each example to a mesh with 1849 vertices provides a consistent dimensionality of 5547 throughout the entire training set. However 90% of the deformation is contained within the primary 10 modes of variation. So, although the training set was originally in 15000 dimensional space, the data actually lies upon a subspace of only 10 dimensions. The most important aspect of the PDM is the predominant z-deformation (blue) in these primary 10 modes. This demonstrates that the alignment and resampling procedure has been successful. During resampling the simplicity of the resampling scheme lead to zero area polygons. After PCA these do not occur as vertices are statistically smoothed by the model. The perturbations of vertices in the x-y plane, which were generated by zero area polygons, are expressed within the lower modes of variation and effectively removed from the model.

8.5 Conclusions

This chapter has demonstrated how the techniques for the assembly of 2D PDMs can easily be extended to 3D. Approaches to the alignment and resampling

procedure have been proposed and a 3D PDM of a human face constructed. Due to the high dimensionality and corresponding complexity of these techniques, variations on the resampling method have been proposed which can be used depending upon the extent and complexity of the training data. It has also been demonstrated that errors introduced during resampling are statistically smoothed and manifest themselves as high frequency oscillations of the model contained within the lower modes of deformation. Since these lower modes are typically discarded it can be deduced that the smoothing effect of the PDM can help reduce errors introduced during assembly.

Future work is to apply these techniques to volumetric segmentation techniques detailed in Appendix 2 to construct 3D PDMs from medical imaging data.