Model–Based Interpretation of 3D Medical Images

A. Hill, A. Thornham and C. J. Taylor

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

Abstract

The automatic segmentation and labelling of anatomical structures in 3D medical images is a challenging task of practical importance. We describe a model–based approach which allows robust and accurate interpretation using explicit anatomical knowledge. Our method is based on the extension to 3D of Point Distribution Models (PDMs) and associated image search algorithms. A combination of global, Genetic Algorithm (GA), and local, Active Shape Model (ASM), search is used. We have built a 3D PDM of the human brain describing a number of major structures. Using this model we have obtained automatic interpretations for 30 3D Magnetic Resonance head images from different individuals. The results have been evaluated quantitatively and support our claim of robust and accurate interpretation.

1 Introduction

The use of 3D images in medicine is becoming increasingly routine. In many applications the first step in processing such images is segmentation. Structures must be identified and delineated for visualisation, volume measurement, treatment planning etc. Performing this segmentation by hand is extremely laborious and time–consuming. An automatic system for segmenting 3D images would be highly advantageous.

We have claimed previously that for robust automatic interpretation of medical images it is essential to have a model describing the shape of the object(s) which we wish to identify [1]. The range of shapes in organs of the human body is enormous and, furthermore, each organ can vary in shape quite markedly between different individuals. Any method of shape representation we employ must be capable of representing a large class of flexible shapes but must also have sufficient *specificity* to enable accurate segmentation of the given object(s) in real data. We have shown that compact, parameterised models of the shape of complex biological objects as observed in 2D medical images can be generated by representing objects as sets of labelled points. A statistical analysis of the way in which the position of the points vary over a set of training examples produces a *Point Distribution Model* (PDM) [1,3]. The PDM technique is capable of representing a wide class of shapes. By using both local [2,3,4] and global [6,7] image search we have shown how the specificity of a PDM enables robust and accurate segmentation of 2D medical images.

We are concerned here with extending the PDM technique to 3D images. We have used Magnetic Resonance (MR) volume images as a basis for building and testing a 3D PDM of structures in the brain. We show that the 3D PDM technique produces a compact parameterised model and we show also how the Active Shape Model (ASM)

BMVC 1993 doi:10.5244/C.7.34

approach to fitting PDMs locally to image data [2,3,4] can be extended to cope with the 3D PDM. Extending the Genetic Algorithm (GA) global optimisation technique [6,7] to 3D is considered and we present quantitative results for a combined GA + ASM segmentation system which show that the specificity of the PDM observed in 2D extends to the 3D case.

2 Point Distribution Models in 2D

2.1 Model Building

The key steps in constructing a PDM from a set of examples of an object to be modelled are as follows (see [1] for full details) :

- Generate a set of object descriptions upon which the model is to be trained. An object description is simply a labelled set of points $(x_0, y_0;\ x_1, y_1;\ \ldots)$. Each labelled point represents a particular position on the object (for example, the tip of the finger on a hand). Corresponding points on different objects represent equivalent locations on each object.
- Align the sets of points and perform a Principal Components Analysis on the ordinates of the points. This involves finding the mean position of each point and the co-variance matrix of the position variables $(x_0, y_0;\ x_1, y_1;\ \ldots)$. The principle eigenvectors of the matrix give the main modes of variation of the training set.

This procedure results in a model with a small set of parameters $\underline{b} = (b_1, b_2, .., b_m)$ which act as weights for the m major eigenvectors of the co-variance matrix. These weights can be manipulated to create new instances from the class of objects modelled.

2.2 Image Search

Cootes et al [2,4] have shown how a PDM can be used to locate an instance of the modelled object(s) in unseen images. The method is similar to the so called *snake* approach described by Kass et al [9] in that image evidence is used to deform the model in an iterative manner until good agreement is achieved between model and data. In the method presented by Cootes, however, the PDM is allowed to vary only in ways learnt during the training of the model i.e. no infeasible shapes are generated during the deformation process. Consequently, these deforming models have been dubbed *Active Shape Models* (ASMs) or *Smart Snakes*. Because ASMs use local image data to suggest deformations to the current instantiation of the PDM, a reasonable starting approximation or *cue* must be provided in the form of an initial set of shape and pose parameters if the procedure is to be successful.

We have shown previously [6] how Genetic Algorithms (GAs) [5,8] can be employed as a global method of optimisation to locate instances of a flexible template in unseen images. We have also shown [7] how GAs can be combined with the PDM technique to produce a generic system for constructing a flexible template of a given object and automatically locating instances of the model in image data. The ASM procedure can be embedded directly in the global GA search in order to improve the reliability of the search and the quality of the final interpretation; this is also discussed in [7].

We have used the PDM/GA/ASM system in 2D to model many different objects and locate instances of the models in unseen image data : structures in the heart [4,7], lat-

eral ventricles of the brain [4,7], organs in the abdomen [4], hands [2], components on a printed circuit board [2,7], brake assemblies [3], faces [3], vertebrae of the spine [11] and hand-written characters [10].

3 Point Distribution Models in 3D

3.1 Model Building

The requirements in 3D are essentially the same as in 2D i.e. that each object is described by a labelled set of points; the points are in 3D rather than 2D $(x_0, y_0, z_0;\ x_1, y_1, z_1;\ \ldots)$. This means that we must be able to generate points on the surface of a given 3D object in a consistent manner such that the same point marks the same location on the surface of all examples of the object. Consider the finger of a hand. Rather than simply marking the tip of the finger (as was the case in 2D) we are now required to mark a point on the surface of the tip of the finger - the tip of the nail, say. In this paper we are particularly interested in modelling structures in the brain as imaged by MR. In order to be able to mark points on the surface of objects in such images the objects must first be segmented from the 3D datasets. The segmentations are usually generated by hand in a slice by slice fashion. We have used this *contour* data as a basis for generating 3D PDMs.

Let us assume that we wish to model a set of cylinders of various heights/widths. Furthermore, assume that the cylinders are described by a set of contours, the number of contours being dependent upon the height of the cylinder (see figure 1). To construct a PDM cylinder model we first generate a number of points along the length of each contour in a similar fashion to the 2D case. Curves in 3D space which connect these points together on successive contours can now be generated. These curves are sub-divided into any number of sub-sections, independent of the number of contours from which they were constructed, but always the same number of sub-sections for each example of the object (see figure 1). These points can now be used to train a PDM as each point is situated at an equivalent location on the surface of each example.

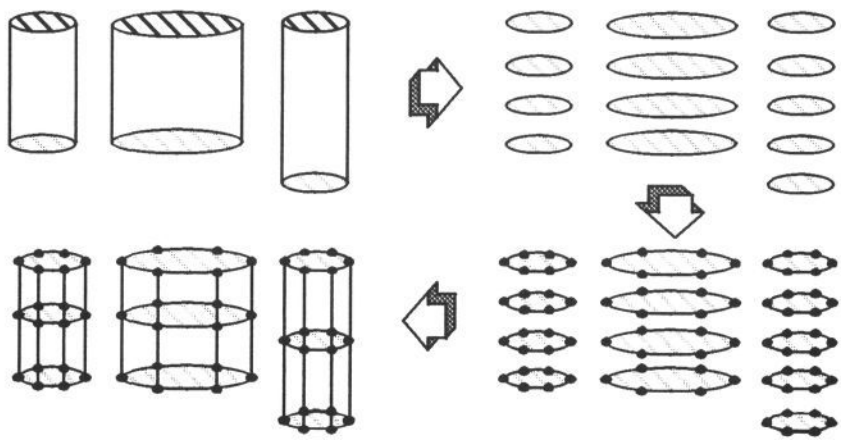

Figure 1: Generating Points for a 3D PDM from Contour Data.

Anatomical objects form complex 3D shapes. The technique described here can only be applied to simple structures in which the topology does not vary from slice to slice.

Complex shapes must be sub–divided into simpler structures to which this technique can then be applied. The most complicated objects we have modelled in the brain are the lateral ventricles which are, essentially, two inverted U shaped tubes which can be modelled as 6 separate parts (three for each tube) - see figure 2.

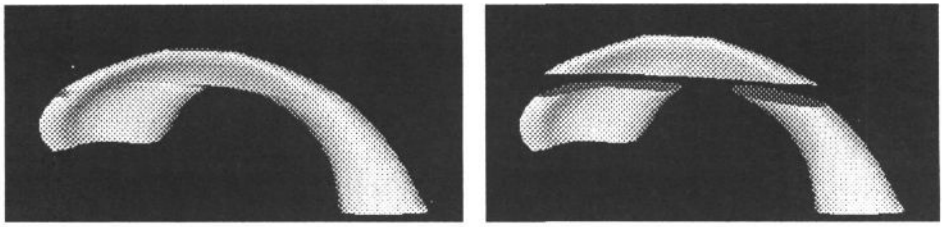

Figure 2: Modelling the Left Lateral Ventricle as 3 Sub–parts.

3.2 Image Search

The ASM technique described by Cootes et al [2,4] has three phases:

i) Profiles normal to the current boundary are generated at each model point. The image data is inspected along the length of the profile to identify an improved location for the given point. The displacement of the model point from its current position is regarded as a *force* acting upon the point.

ii) Changes in the position (t_x, t_y), scale (s) and orientation (θ) are calculated to minimise the distance (in a least squares sense) between the current model points and their new suggested locations. The changes $(t_x, t_y), s, \theta$ are then applied to the model points and the residual displacements which remain are computed. These residuals are considered to be associated with changes in shape.

iii) The residual displacements are used to compute changes to the model shape parameters, b_i, in order to minimise the distance (again in a least squares sense) between the translated, scaled and rotated model points and the positions of the model points suggested by the image data.

This approach can be extended to 3D. Profiles normal to the surface of the 3D object(s) are generated. The image data is inspected along the length of each profile to generate improved model point locations. In the experiments described below we simply look for the strongest edge in the image data along the profile. We intend to improve this scheme by using the statistics of the grey level values along the profiles as observed in the training data as described in [3,4] for the 2D case. Changes in 3D position (t_x, t_y, t_z), scale (s) and orientation (θ) are computed to minimise the distance between the current model points and the positions suggested by the data (see Appendix A). The changes in $(t_x, t_y, t_z), s, \theta$ are then applied and the resulting residual displacements used to compute changes in the shape parameters (see Appendix B). The ASM in 3D proves to be a very effective technique for generating accurate interpretations, even when the initial cue is fairly poor. In figure 3 we show the initial cue and final interpretation for several slices from a 3D MR volume of the brain.

For the GA global search there is little difference between 2D and 3D save for the number of parameters which must be manipulated. In 2D there are 4 global transform-

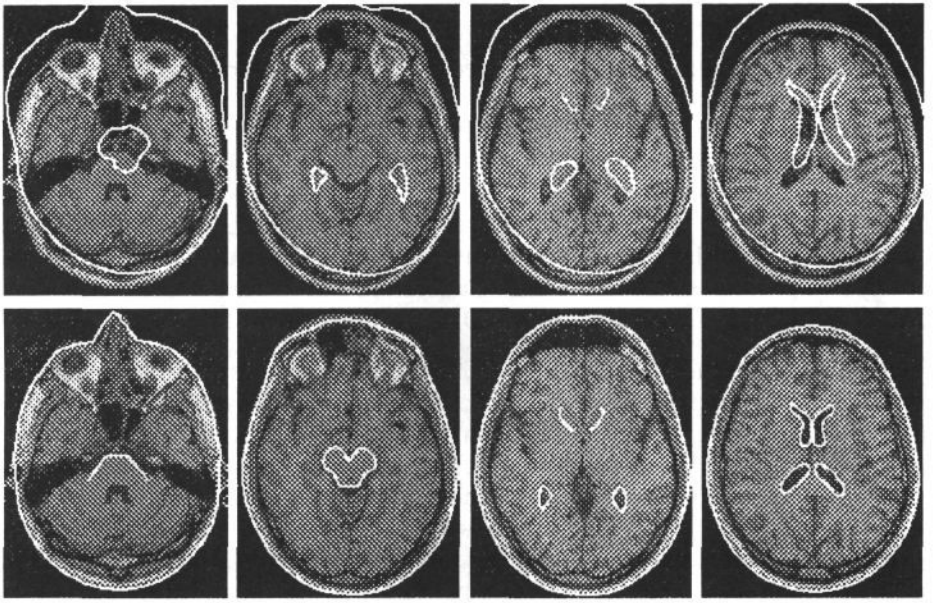

Figure 3: Several Slices from a 3D MR Volume of the Brain. The top row shows the initial cue for the ASM; the bottom row shows the final solution. The ASM fits the 3D PDM to all slices simultaneously.

ation parameters $(t_x, t_y), s, \theta$ while in 3D there is one extra – t_z (for the brain example below the two extra rotations are captured by the model – see section 4). For most of the models we have employed in 2D the number of shape parameters required to capture the variation observed in the training set is typically 3–12. For the 3D model of the brain we have generated, the number of parameters is 20–30. Clearly, this represents a much greater search problem but one which the GA is still capable of handling. The ASM technique can be embedded within the GA search in the same way as described in [7] for the 2D case. The GA must also be supplied with an objective function to optimise. The objective function expresses the amount of evidence there is in the image data for a given instantiation of the model; we have used the following function in the experiments described below :

$$f = \frac{1}{P}\sum_{i=1}^{P}\hat{f}_i, \quad P = \text{number of separate 3D sub-parts in the model and}$$

$$\hat{f}_i = \frac{1}{\bar{g}}\frac{1}{N_i}\sum_{j=1}^{N_i}\left\{\left|\frac{p_j}{\max(|p_{\min}|,|p_{\max}|)}\right| + \left|1-\frac{g_j}{\bar{g}}\right|\right\}, \quad \bar{g} = \frac{1}{N_i}\sum_{j=1}^{N_i} g_j$$

where N_i is the number of points for the i^{th} 3D sup-part, p_j is the position along the j^{th} profile lying on $[p_{\min}, p_{\max}]$ at which the best edge is located and g_j is the strength of the edge at p_j. This function is minimised by strong edges ($\bar{g}$ large) of equal magnitude $(1 - g_j/\bar{g} = 0 \ \forall \ j)$ located at the points predicted by the model ($p_j = 0$).

4 The PDM Brain Atlas

The model of the brain we have constructed consists of the lateral ventricles, the caudate nucleus, a section of the brain stem and the sub–cutaneous fat which we will loosely refer to as *skin*. The skin and lateral ventricles are clearly visible in T1 weighted MR volume images of the brain. The caudate nucleus and, to some extent, the brain stem are poorly defined structures. The caudate nucleus has a strong spatial relationship with the lateral ventricles, in fact, they share a common surface. We have included it in the model in order to show that such a poorly defined structure can be located reliably as a consequence of its spatial relationships with other better defined structures. The PDM uses approximately 2,300 points to represent these structures.

The training set for the PDM consisted of 30 MR volume images of *normal* brains (volunteer data). All the structures in the model were segmented by hand in a slice by slice fashion. This provided the contour data on which the model was trained. Each of the members of the training set was translated, scaled and partially aligned into a common frame of reference. Currently, the alignment is only in the (x,y) plane in which the contours are defined. In 3D there remain two further orientations. These are captured by the model as *shape* variations. In order to increase the size of the training set we included a second example for each member in which the left and right sides of the brain were reflected through the mid–plane. This enabled the model to predict asymmetry on either side of the brain in unseen datasets. With a larger set training the inclusion of a left–right reflection would not be necessary.

Using these 60 examples we found that a PDM with 20 modes of variation captures 90% of the variation observed in the training set and 30 modes of variation captures 95% of the variation. In all the experiments described below we have employed 20 modes of variation. The mean shape and the first and second modes of variation are shown in figures 4 and 5 respectively. The modes represent forward–backward tilt (nodding) and change in the size of the lateral ventricles respectively.

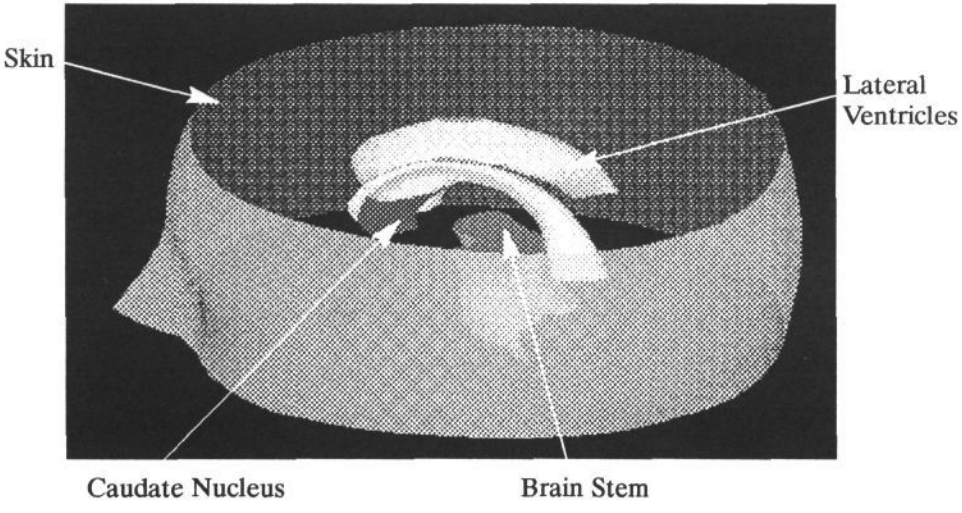

Figure 4: The PDM of the Brain – Mean Shape.

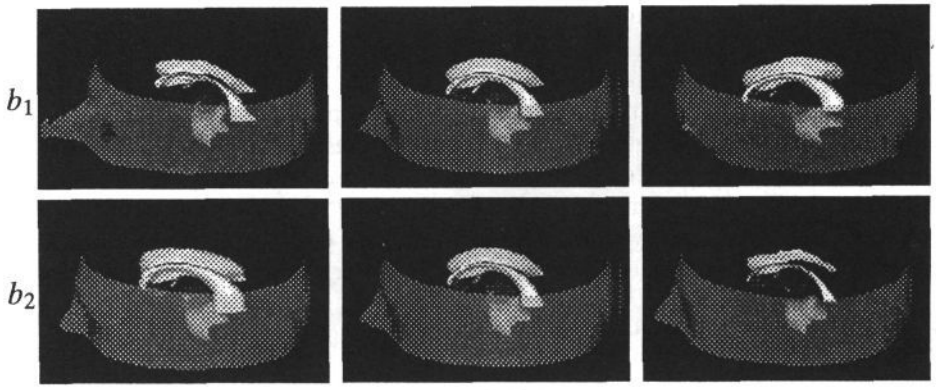

Figure 5: The PDM of the Brain : First Two Modes of Variation (forward–backward tilt and ventricular size).

5 Results

Given a PDM MR brain atlas trained upon 30 examples, we would like to know two things :

i) how accurate are the interpretations which the system generates?

ii) how well can we generalise to unseen data using just 30 examples?

In order to answer both of these questions we need some quantitative measure which indicates how well the automatically derived interpretations fit the image data. Each member of the training set is represented by a set of points which are derived from hand–segmentation. We have used these as ground truth. For any given interpretation we can compute the inter–point distance between the automatic interpretation and ground truth for each point in the model. We have used the average of the inter–point distances as a measure of fit. We are particularly interested here in the ability to locate accurately the internal structures in the brain; consequently, the inter–point distances for the skin are excluded from the calculation.

To answer the first of our questions we trained the PDM on all 30 examples and used this model to interpret all 30 MR datasets. We computed the inter–point distances between the model–based interpretation and ground truth for all 30 examples. The results for the best, average and worst interpretations are shown in table 1. The average error over the whole training set is of the order of 2 voxels which indicates that the system is capable of accurate interpretation. Indeed, the average error for the worst interpretation is only 2.9 voxels.

	Average Inter–Point Distance		
	Best	**Average**	**Worst**
Leave all in	1.3	2.0	2.9
Leave one out	2.0	3.0	5.0

Table 1: Average Inter–Point Distances between Automatic Interpretations and Ground Truth.

To ascertain how well we can generalise using just 30 examples we have performed a set of leave–one–out experiments. Each member of the training set was taken in turn

and removed from the training set. The 29 examples which remained were used to train a PDM. The resulting model was then employed to interpret the dataset for the excluded member of the training set. In this way, the excluded member acted as an unseen dataset upon which the model was tested. The inter-point distances were again computed; the best, average and worst results for these leave-one-out experiments are also shown in table 1. Comparing the leave-one-out with the leave-all-in results shows clearly that the training set is not yet large enough to allow complete generalisation to unseen data. However, an average inter-point error of 3 voxels indicates that the training set does not need to be considerably larger to allow such a generalisation.

6 Conclusions

We have shown that when the PDM technique is applied to a complex, variable, biological 3D object, i.e. the brain, a compact model with 20–30 modes of variation results. This is a considerable reduction in complexity; the raw model with 2300 points. We have also shown that the ASM algorithm can be extended to cope with the PDM of the brain we present here and that the GA + ASM optimisation system can be successfully employed to locate instances of the model in unseen volume data. Both the ASM and GA are computationally tractable – an application of the ASM using 100 iterations requires approximately 30 CPU seconds and 50 generations of a GA search with a population size of 50 requires approximately 10 minutes of CPU time on a SUN SPARCstation 10. More than an hour is required to hand-segment the structures which are currently in the model. The segmentation problem we have addressed is notoriously difficult and our results seem much better than any previously reported.

There are several avenues which we intend to explore in order to improve the method. An increase in the size of the training set is the most obvious extension which will enable greater generalisation to unseen data. We anticipate the use of grey-level models at each point of the model, learnt during the training phase as described in [3,4] for the 2D case, would help improve the accuracy of the interpretation. A model with greater specificity could be generated by more accurate placement of the landmark points over the surface of the 3D objects and by removing the two orientations currently captured by the model as modes of variation. Tools to help achieve this last goal are i) sophisticated interaction software for manipulating 3D objects and identifying landmark points and ii) automatic methods for parameterising the surface of a 3D object in a consistent manner. Such automatic methods must be able to cope with the added constraint that the position of some landmarks points on the surface may have already been pre-determined (interactively say).

7 Acknowledgements

This research was funded by the UK Science & Engineering Research Council and Department of Trade & Industry. The work has been carried out in collaboration with British Technology Group; C. N. Software Ltd; Hill Centre, London Hospital Medical College; Department of Bio-Medical Physics and Bio-Engineering, University of Aberdeen; Centre for Parallel Computing, Queen Mary & Westfield College.

Thanks to Dr. Charles Hutchinson (Department of Diagnostic Radiology, University of Manchester) for providing the 3D MR datasets and for his time and expert knowl-

edge in labelling the datasets. Thanks also to George Cameron (Department of Bio-Medical Physics, University of Aberdeen) for help in segmenting the datasets.

8 References

[1] **Cootes T. F., Cooper D. H., Taylor C. J., Graham J.** *Training Models of Shape from Sets of Examples*, Proc. British Machine Vision Conference, Leeds, 1992, Springer Verlag; 9–18.

[2] **Cootes T. F., Taylor C. J.** *Active Shape Models – Smart Snakes*, Proc. British Machine Vision Conference, Leeds, 1992, Springer Verlag; 266–275.

[3] **Cootes T. F., Taylor C. J., Lanitis A., Cooper D. H., Graham J.** *Building and using Flexible Models Incorporating Grey–Level Information*, Proc. International Conference on Computer Vision, Berlin, May 1993; 242–246.

[4] **Cootes T. F., Hill A., Taylor C. J., Haslam J.** *The Use of Active Shape Models for Locating Structures in Medical Images*, Proc. 13th International Conference of Information Processing in Medical Imaging, Arizona, June 1993, 33–47.

[5] **Goldberg D. E.** *Genetic Algorithms in Search, Optimisation and Machine Learning*, Addison–Wesley, 1989.

[6] **Hill A., Taylor C. J.** *Model–Based Interpretation using Genetic Algorithms*, Image and Vision Computing, 1992, 10(5): 295–300.

[7] **Hill A., Cootes T. F., Taylor C. J.** *A Generic System for Image Interpretation using Flexible Templates*, Proc. British Machine Vision Conference, Glasgow, 1992, Springer Verlag; 267–285.

[8] **Holland J. H.** *Adaptation in Natural and Artificial Systems*, University of Michegan Press, Ann Arbor, 1975.

[9] **Kass M., Witkin A., Terzopoulos D.** *Snakes: Active Contour Models*, 1st International Conference on Computer Vision, IEEE Computer Society Press (1987); 259–268.

[10] **Lanitis A., Taylor C. J., Cootes T. F.** *A Generic System for Classifying Variable Objects using Flexible Template Matching*, Proc. British Machine Vision Conference, Guildford, 1993.

[11] **Lindley K.** *Model–Based Interpretation of Lumbar Radiographs*, MSc Thesis, University of Manchester, England, 1992.

Appendix A – Alignment of Two Sets of Points

Assume that $\underline{x}_1$ and $\underline{x}_2$ are n element vectors containing the homogeneous co–ordinates of the n points in a 3D PDM, viz $\underline{x} = [\underline{p}_1, \underline{p}_2, .., \underline{p}_n]^T$ $\underline{p}_i = (x_i, y_i, z_i, 1)$. We wish to apply a translation, scaling and rotation to the vector $\underline{x}_2$ in order to align $\underline{x}_2$ and $\underline{x}_1$ as closely as possible. Define :

$$\mathbf{Q} = \begin{bmatrix} ds & -d\theta & 0 & dt_x \\ d\theta & ds & 0 & dt_y \\ 0 & 0 & ds & dt_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{M} = \begin{bmatrix} \mathbf{Q} & \mathbf{0} & \mathbf{0} & . \\ \mathbf{0} & \mathbf{Q} & \mathbf{0} & . \\ \mathbf{0} & \mathbf{0} & \mathbf{Q} & . \\ . & . & . & . \end{bmatrix}$$

Note that in **Q** we only consider a single rotation in the (x,y) plane in which the contours for the model were generated and that we also use the approximation of small angle $\sin(d\theta) = d\theta$, $\cos(d\theta) = 1$. We can minimise the following :

$$E^2 = (\underline{x}_1 - \mathbf{M}\underline{x}_2)^T \mathbf{W} (\underline{x}_1 - \mathbf{M}\underline{x}_2)$$

to generate a least squares solution. **W** is an optional diagonal matrix of weights. The 5 simultaneous equations are given by :

$$\left.\begin{aligned} \partial E^2/\partial ds &= 0 \\ \partial E^2/\partial d\theta &= 0 \\ \partial E^2/\partial dt_x &= 0 \\ \partial E^2/\partial dt_y &= 0 \\ \partial E^2/\partial dt_z &= 0 \end{aligned}\right\} \text{ giving } \begin{bmatrix} X_2 & -Y_2 & W & 0 & 0 \\ Y_2 & X_2 & 0 & W & 0 \\ Z_2 & 0 & 0 & 0 & W \\ D_1 & 0 & X_2 & Y_2 & Z_2 \\ 0 & D_2 & -Y_2 & X_2 & 0 \end{bmatrix} \begin{bmatrix} ds \\ d\theta \\ dt_x \\ dt_y \\ dt_z \end{bmatrix} = \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ E_1 \\ E_2 \end{bmatrix}$$

where :

$$X_i = \sum_{k=0}^{n-1} w_k x_{ik} \quad Y_i = \sum_{k=0}^{n-1} w_k y_{ik} \quad Z_i = \sum_{k=0}^{n-1} w_k z_{ik} \quad W = \sum_{k=0}^{n-1} w_k$$

$$D_1 = \sum_{k=0}^{n-1} w_k (x_{2k}^2 + y_{2k}^2 + z_{2k}^2) \quad D_2 = \sum_{k=0}^{n-1} w_k (x_{2k}^2 + y_{2k}^2)$$

$$E_1 = C(x_1, x_2) + C(y_1, y_2) + C(z_1, z_2) \quad E_2 = C(y_1, x_2) - C(x_1, y_2)$$

$$C(x_i, y_j) = \sum_{k=0}^{n-1} w_k x_{ik} y_{jk}$$

Appendix B – Changes in Shape Parameters

Points in the model reference frame $(\underline{y})$ are transformed into the image frame by:

$$\mathbf{Q} = \begin{bmatrix} s\cos(\theta) & -\sin(\theta) & 0 & t_x \\ \sin(\theta) & s\cos(\theta) & 0 & t_y \\ 0 & 0 & s & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{M} = \begin{bmatrix} \mathbf{Q} & \mathbf{0} & \mathbf{0} & . \\ \mathbf{0} & \mathbf{Q} & \mathbf{0} & . \\ \mathbf{0} & \mathbf{0} & \mathbf{Q} & . \\ . & . & . & . \end{bmatrix}, \quad \underline{x} = \mathbf{M}\underline{y}$$

Assume that we have such a set of image model points $\underline{x}$ and an associated set of displacements $\underline{\delta x}$ suggested by the image data. By setting $\underline{x}_1 = \underline{x} + \underline{\delta x}$ and $\underline{x}_2 = \underline{x}$ we can apply the method in Appendix A to compute $ds, d\theta, dt_x, dt_y, dt_z$ thus aligning approximately $\underline{x}$ with $\underline{x} + \underline{\delta x}$. In order to compute changes to the shape parameters $\underline{db} = [db_1, db_2, .., db_m]^T$ we use :

$$\underline{db} = \mathbf{P}^T\{\mathbf{M}_2\mathbf{M}_1(\underline{x} + \underline{\delta x}) - \underline{y}\},$$

$$\mathbf{Q}_1 = \begin{bmatrix} 1 & 0 & 0 & \hat{t}_x \\ 0 & 1 & 0 & \hat{t}_y \\ 0 & 0 & 1 & \hat{t}_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{Q}_2 = \begin{bmatrix} \hat{s}\cos(\hat{\theta}) & -\sin(\hat{\theta}) & 0 & 0 \\ \sin(\hat{\theta}) & \hat{s}\cos(\hat{\theta}) & 0 & 0 \\ 0 & 0 & \hat{s} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\hat{s} = 1/(s * ds)$$
$$\hat{\theta} = -(\theta + d\theta)$$
$$\hat{t}_x = -(t_x + dt_x)$$
$$\hat{t}_y = -(t_y + dt_y)$$
$$\hat{t}_z = -(t_z + dt_z)$$

Where **P** is the matrix of eigenvectors generated by the Principal Components Analysis and $\mathbf{M}_2\mathbf{M}_1$ transforms the image points back into the model reference frame.