

A Heuristic Image Search Algorithm for Active Shape Model Segmentation of the Caudate Nucleus and Hippocampus in Brain MR Images of Children with FASD

A.A. Eicher*, P. Marais*, C. Warton†, S.W. Jacobson^{§†‡}, J.L. Jacobson^{§†‡}, C.D. Molteno[§], E.M. Meintjes^{†¶}

*Department of Computer Science, University of Cape Town, Rondebosch 7701, South Africa

†Department of Human Biology, University of Cape Town, Observatory 7925, South Africa

§Department of Psychiatry and Mental Health, University of Cape Town, Observatory 7925, South Africa

¶MRC/UCT Medical Imaging Research Unit, University of Cape Town, Observatory 7925, South Africa

‡Department of Psychiatry and Behavioral Neurosciences, Wayne State University, Detroit, MI

ABSTRACT

Magnetic Resonance Imaging provides a non-invasive means to study the neural correlates of Fetal Alcohol Spectrum Disorder (FASD) - the most common form of preventable mental retardation worldwide. One approach aims to detect brain abnormalities through an assessment of volume and shape of two sub-cortical structures, the caudate nucleus and hippocampus. We present a method for automatically segmenting these structures from high-resolution MR images captured as part of an ongoing study into the neural correlates of FASD.

Our method incorporates an Active Shape Model, which is used to learn shape variation from manually segmented training data. A modified discrete Geometrically Deformable Model is used to generate point correspondence between training models. An ASM is then created from the landmark points. Experiments were conducted on the image search phase of ASM segmentation, in order to find the technique best suited to segmentation of the hippocampus and caudate nucleus. Various popular image search techniques were tested, including an edge detection method and a method based on grey profile Mahalanobis distance measurement. A novel heuristic image search method was also developed and tested. This heuristic method improves image segmentation by taking advantage of characteristics specific to the target data, such as a relatively homogeneous tissue colour in target structures.

Results show that ASMs that use the heuristic image search technique produce the most accurate segmentations. An ASM constructed using this technique will enable researchers to quickly, reliably, and automatically segment test data for use in the FASD study.

KEYWORDS: Active Shape Model, Geometrically Deformable Model, ASM

1 INTRODUCTION

Fetal Alcohol Spectrum Disorder (FASD) is especially prevalent amongst residents of the Western Cape region of South Africa. This disorder affects the embryos of women who ingest alcohol whilst pregnant. Children can suffer serious central nervous system damage as a result. In order to gain a better understanding of this condition, 32 children with FASD¹ were scanned using *Magnetic Resonance Imaging (MRI)*.

This paper forms part of an ongoing study aimed at assessing the neural correlates of FASD. One topic of study is the difference in shape, volume and area between brain structures, specifically the caudate nucleus and hippocampus, of healthy subjects and those with FASD. The primary goal of this work is to find

the best algorithm to automatically segment out caudate nuclei and hippocampi from previously unseen brain volumes that form part of the study into FASD.

Manual segmentation of MR images is time-intensive and prone to inter-observer and intra-observer variability [1, 2]. Automatic segmentation of MR images is generally quick and reproducible. It is therefore desirable to automatically segment the MR images used as part of the study. This will enable researchers to quickly and objectively compare healthy specimens to those affected by FASD. In order to achieve this goal, an effective automatic segmentation algorithm must be found.

We construct a *Geometrically Deformable Model (GDM)* that effectively and efficiently assigns landmark points to the training data made available for use as part of the FASD study. We then use an *Active Shape Model (ASM)* for automatic segmentation. ASMs overcome spurious shape deformities by

Email: A.A. Eicher aeicher@cs.uct.ac.za

¹9.7 to 13.7 years of age, mean 11.6 years

learning shape information from a training data set - thereby limiting model deformation to shapes similar to those found in the training set. ASMs are not a general solution to the image segmentation problem, but work for a broad class of segmentation problems, especially in the biomedical imaging field, where volumetric images such as MRI scans are segmented [3, 4, 5, 6]. ASMs are therefore well suited to our specific needs.

ASMs are generated by performing *Principal Component Analysis (PCA)* on a set of landmark points. This process creates a statistical *Point Distribution Model (PDM)* which is used to constrain the deformation of shape models during the image segmentation process. The image segmentation process uses an image search algorithm, in which the model is expanded and deformed iteratively to fit local image information (such as an edge). We implemented various image search algorithms, in order to find the one that is most suited to our test data. These included an edge detection method, a method based on grey profile Mahalanobis distance measurement, and a heuristic image search method.

After segmentation, results were visualised and analysed not only to make use of the segmented data, but also to evaluate the effectiveness of the segmentation process. Experimental evaluation was employed to determine the most effective ASM construction method. ASMs built to use the various image search techniques were tested on sample data that are characteristic of the data found in the FASD study. Statistical analysis was conducted on the results, in order to draw meaningful conclusions. This task enabled us to choose the best ASM construction technique for use in the future of the FASD study.

Results indicate that the heuristic method generated the most accurate segmentation results. The method is not perfect - segmentation did fail on some data sets. However, this segmentation failure was usually due to extreme noise or low contrast. Thus, we can conclude that the heuristic method displays acceptable reliability for the segmentation of current and expected future data to be used in the FASD study.

The paper is structured as follows. Section 2 provides details of previous work that we build upon in this research, as well as some fundamental techniques used. Our test data are described in Section 3. Specific details of GDM construction are given in Section 4. A detailed description of our ASM construction techniques can be found in Section 5. This is followed by a discussion of our Heuristic Image Search algorithm in Section 6. Finally, we evaluate the performance of our algorithm, and present our conclusion.

2 BACKGROUND AND RELATED WORK

Many automatic segmentation methods have been proposed in recent years - each has its advantages and disadvantages. Each method is also suitable for certain types of target data. GDMs are well suited to the segmentation of 3D data where the target shapes are closed surfaces. The problem with standard GDMs

is that they can deform to arbitrary shapes that are not representative of the class of shapes that they are designed to fit. This problem is especially prevalent when segmenting noisy data, which contain many false positives. Shape priors, such as surface smoothness constraints, are sometimes used to limit deformation to a shape that is geometrically similar to the original model, but it is still possible for models to deform into suboptimal shapes [3].

ASMs were first proposed by Cootes in 1995 in an attempt to address the shortcomings of deformable models, by limiting model deformation to shapes found within the Allowable Shape Domain (ASD) [7]. ASMs achieve this by learning shape information from pre-segmented shape training sets. The first ASMs were limited to 2D, but 3D versions of the technique followed soon after [4].

In order to build an ASM, landmark points must first be assigned to training data. In 3D, it is all but impossible to assign these landmark points manually. A GDM can be employed to automatically assign 3D landmark points to volumetric data. The GDM consists of a discrete mesh of vertices, which move in 3D space in reaction to forces exerted on them by various deformation terms. These terms cause the GDM to deform to fit a series of binary training shape volumes, thereby assigning a fixed number of vertices or landmark points to the training data.

2.1 GDM Construction

The first task in creating the GDM is to develop an initial mesh, which is then deformed to fit the target structure. In the original GDM paper, [8] use a regular 20-sided icosahedron as their initial mesh. [9] use their *gravity centre* method to stitch together polygons from user-generated slices. [10] create a mesh using manually segmented data. These data are segmented on a slice-by-slice basis, based on a method used by [11]. Lee *et al* convert their original tracing into *minimal hexahedrons* and then smooth the data using a low-pass filter. It is clipped and converted into a binary volume. An initial ellipsoid model is then fitted to this binary volume, using a GDM method first proposed by [12]. The method involves deforming an initial mesh to fit a binary volume. It employs a multi-scale approach to deformation - starting with an initial low resolution mesh, and tessellating it as deformation approaches convergence. This approach is novel in that it explicitly prevents self-intersection of the deforming mesh by heavily penalising inter-polygon proximity below a certain threshold.

Deformation of the GDM is controlled by iterative minimisation of a cost function. Since there is a large parameter space to search, a naïve brute force approach is intractable. The Conjugate Gradient method, discussed by [13], is therefore employed. This method computes successive line minimisations of the cost function (also called the *objective* function). It uses the derivative of the objective function to calculate optimal direction vectors for british (called *conjugate gradients*). After each line minimisation, mesh

vertices are updated to locally optimal positions. The mesh can therefore be seen to iteratively deform to fit its target. When the difference between successive objective function evaluations is smaller than a certain threshold, the algorithm is considered to have converged, and deformation stops.

2.2 ASM Construction

ASMs are generated by performing *Principal Component Analysis (PCA)* on a set of delimited, registered training shapes, called a *Point Distribution Model (PDM)*. This process produces shape descriptors, which are used to constrain the deformation of shape models during the image segmentation process. The image segmentation process uses an image search algorithm, in which the model is expanded and deformed iteratively to fit local image information (such as an edge). After segmentation, the results must be visualised and analysed not only to make use of the segmented data, but also to evaluate the effectiveness of the segmentation process.

Typically, training data sets are manually segmented into 3D binary volumes and an isosurface is created from one of the volumes using an algorithm such as the Marching Cubes method [14]. This isosurface, containing arbitrary 3D landmark points, is then used in a GDM and deformed to fit the other segmented volumes [4]. This creates a PDM; consisting of a set of training shape isosurfaces with corresponding landmark points, which are then registered to a common coordinate frame using an algorithm similar to the Procrustes method [15, 16]. After registration, the PDM is passed to the PCA phase for analysis.

It is not always desirable to create a PDM during this stage, since PCA can be performed on a variety of non-Euclidean shape descriptors, as long as a vector of values is used to describe each training shape. Examples of these shape descriptors include: the Minimum Description Length (MDL) approach [17, 18], mapping to Spherical Harmonics (SPHARM) descriptors [4], and mapping to Spherical Wavelet Basis functions [3]. The general (simplified) algorithm followed when using these shape descriptors is as follows:

1. Manually segment the input data into a 3D binary volume of voxels
2. Map each surface voxel to parameter space, using an invertible function
3. Perform parameter space registration
4. Perform PCA on shape parameters

New shapes generated by the ASM in parameter space can then be mapped back to Euclidean space using the inverse mapping function of that in step 2. Refer to Appendix B for more detail on PCA in ASM construction.

2.3 Image Search Techniques

In order to use an ASM in image segmentation, it is necessary to deform it to fit target data. Target data usually consist of greyscale voxels derived from

some imaging process, such as MRI, and are typically noisy. Pre-processing steps such as thresholding and binary morphology may be necessary to minimise noise, thereby allowing for better boundary detection whilst fitting the ASM to the target data. Edge detectors may be used to detect candidate target boundaries within target data. Anisotropic data can be adjusted to be isotropic, however care should be taken to ensure that the target data dimensions are in proportion to those found in the training images used to construct the ASM.

Image search techniques are varied, although they generally search for the ideal *shape*, *scale* and *pose* parameters which best fit the ASM to the target data. In classical 2D ASM segmentation, an instance of the model, X , is defined as follows [7]:

$$X = M(s, \theta)[x] + X_c \quad (1)$$

$M(s, \theta)$ is a rotation by θ and a scaling by s . x is a vector of landmark point coordinates representing the current relative position of each point of the shape, and $X_c = (x_c, y_c, x_c, y_c, \dots, x_c, y_c)^T$ is a vector representing the uniform translation of the landmark points to a centre point in the target data. s , θ and X_c together form the scale and pose parameters, and x the shape parameters.

In order to determine the parameters which best fit the data, it is necessary to first determine the set of adjustments $dX = (dX_0, dY_0, \dots, dX_{n-1}, dY_{n-1})^T$ which will translate each landmark point closer to the target boundary. This can be done in various ways, such as region statistics-based search [3], mutual information-based coordinate descent [5], or a search using grey value intensity profiles [4]. A simple approach is to find the normal to the model boundary at each landmark point, and determine where it intersects the target boundary. The distance to move the landmark point along the surface normal is then set proportional to the edge strength at the boundary [7].

Once dX has been determined, appropriate changes to scale and pose parameters need to be found. This is done by finding the best scaling (ds), translation (dX_c, dY_c) and rotation ($d\theta$) values which map X to $(X + dX)$ [7]. The next step is to find dx , the changes to shape parameters necessary to fit the boundary. Once the appropriate changes to the scale, pose and shape parameters have been found, the model is deformed and the algorithm is repeated. This carries on until it converges to a steady state where no significant change is made between successive iterations. Shape deformation is driven by varying the ASM basis weights, within appropriate predefined limits. Because of this, we can be certain that the model deformation will be constrained to generate shapes that are within the ASD. We described this procedure in 2D for the sake of simplicity, but it is easily and intuitively extended to 3D.

3 DATA

The ASM training data used here consists of MRI brain scans that were acquired as part of a study of Fetal Alcohol Spectrum Disorder [19, 20].

All scans were acquired using a 3T Siemens Allegra MRI scanner (Siemens Medical Systems, Erlangen, Germany). High-resolution anatomical images were acquired in the sagittal plane using a 3D inversion recovery gradient echo sequence (160 slices, $TR = 2300ms$, $TE = 3.93ms$, $TI = 1100ms$, slice thickness $1mm$, in-plane resolution $1 \times 1mm^2$).

In order to use these training data, the structures of interest to this study, namely the left and right caudate nuclei and hippocampi, were manually segmented from each greyscale MRI volume by a neuroanatomist using MultiTracer software [21]. Training data was sampled at $1mm$ voxel resolution. This initial segmentation yields binary volumes representing the ROIs (caudate nuclei and hippocampi) in the training data.

4 LANDMARK POINT GENERATION

In order to capture the shape differences between the various training data, it is necessary to assign 3D landmark points to specific parts of each binary volume, thus creating mesh representations of the data. Each vertex of each mesh represents a specific point on the training data, thus generating point correspondence between training data points. Point correspondence allows for the measurement of variance in shape between training samples, and is therefore necessary to train the ASM used in the next stage of the project.

Manual assignment of landmark points is too time-consuming, and error-prone. Therefore, a discrete GDM, with a fixed number of points, was fitted to each binary training volume. Our GDM implementation is based on [12]. Certain changes have been made to better suit our target data. Our model is discussed in detail in Appendix A.

4.1 Self-proximity Term

MacDonald's self-proximity term measures the smallest distance between pairs of non-adjacent polygons. If a pair of polygons is within a certain distance threshold, a cost is assigned based on the distance. In order to explicitly prevent self-intersection, this cost increases asymptotically towards infinity as the inter-polygon distance approaches zero.

There are three problems with integrating this specific implementation into our model. Firstly, the computational cost of evaluating each pair of faces is unnecessarily high. Secondly, determining self-intersection using a distance measure is unnecessary, since there are methods to quickly test for the intersection of two triangles. Thirdly, assigning a prohibitively high cost to self-intersection creates problems with the Conjugate Gradient Method used to minimise the cost function, since the inverse partial

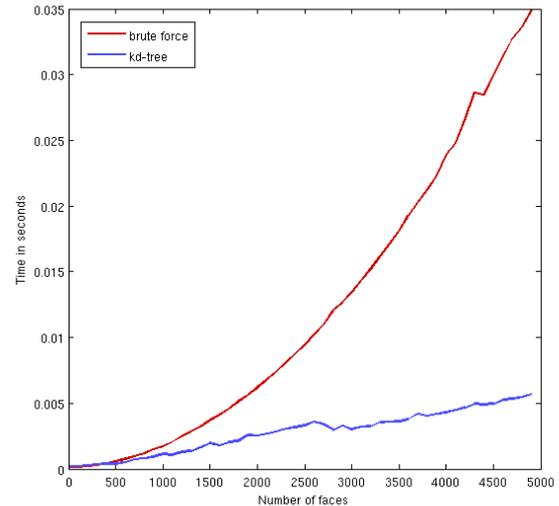


Figure 1: **Brute force vs Kd-tree.** As the number of faces in the mesh increase, the solution using the brute force takes exponentially longer, whilst the solution using the kd-tree increases linearly in time.

derivatives used as direction vectors for line minimisation result in attempts to minimise in suboptimal directions. These problems are addressed as follows.

Computational Cost

The computational cost of comparing each model face with every other model face is $O(n^2)$. This amount of computation is unnecessary, for the following reason: there is a limit on the distance that any single vertex can be translated in one iteration of deformation. Therefore, there is a limit to the change of distance between any two pairs of faces in one iteration. Since there is a limit to the change in distance between faces, only pairs of faces that are located within this distance threshold of each other stand a chance of intersecting as a result of any given iteration. Thus, for each face, it is only necessary to measure distances to other faces that are within the distance threshold. We take the distance threshold to be equivalent to the search distance used in the image search term, since vertices can not be moved further than this in one iteration. A kd-tree [22] is used to quickly identify faces that are within the defined distance threshold of each other. The kd-tree is built at the beginning of each iteration of deformation, at a negligible cost to computation (~ 0.0035 seconds on 2.66GHz Intel Core2 Duo E6750). During evaluation of the self-proximity term, the nearest non-adjacent neighbours to each face are tested for intersection with that face. Only neighbours within the prescribed distance threshold are tested. Figure 1 shows a comparison between the brute force approach and the kd-tree approach. It is clear that, as the number of faces in the mesh increase, the solution using the brute force approach runs in quadratic time, whilst the solution using the kd-tree increases linearly in time.

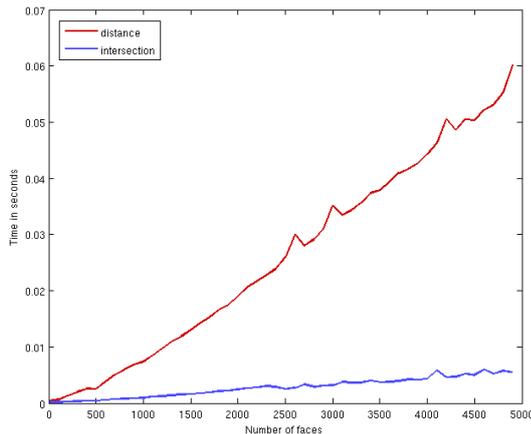


Figure 2: **Triangle intersection vs Distance measure**
As the number of faces in the model increase, the intersection test performs significantly better than the distance measure.

Determining Self-intersection

The self-proximity term is used to prevent the deforming model from self-intersecting. Therefore, the actual distance between any given pair of faces is irrelevant, as long as it is above zero. Thus, using a distance measure to determine whether pairs of faces intersect is a computationally costly process that could be replaced by a simple triangle/triangle intersection test. The method described in [23] is used to test for intersections. If two faces intersect, the self-proximity term will evaluate to 1 for the candidate face. If no two faces intersect, the self-proximity term evaluates to 0. The term's weight parameter is used to assign an actual prohibitive cost to self-intersection. In order to validate the use of the triangle intersection test, we compare our method to the distance measure method in terms of evaluation time. We use the triangle-triangle distance measure, as detailed in [24], for comparison. The results can be seen in Figure 2. It is clear that as the number of faces in the model increase, the intersection test performs significantly better than the distance measure.

Prohibitively High Cost

In order to explicitly prevent self-intersection, MacDonald *et al* set the self-proximity term to increase asymptotically towards infinity as the distance between two polygons reaches zero. Assigning such a high cost to intersecting faces prevents our model from iterating past a point where a potential intersection might take place. This is due to the fact that the cost function derivative is approximated numerically, and therefore uses discrete steps in evaluating the derivative of a continuous function. Because of the fixed step size used in derivative calculation, derivatives of functions that have a rapidly increasing gradient (such as MacDonald's implementation of the self-proximity term) can not be accurately approximated. This inaccuracy results in extremely large partial derivatives, which throw off the Conjugate Gradient Method used

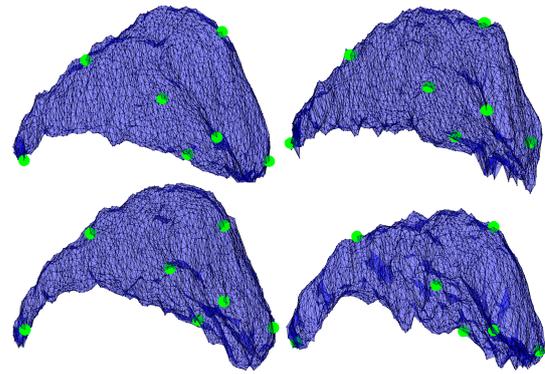


Figure 3: **Point correspondence.** Selected representative vertices (403, 489, 1371, 1657, 1794, 1883, 2729) are highlighted on each of 4 fitted, registered meshes. The meshes differ greatly in shape, but the vertices are in clear point correspondence with one another.

for cost function minimisation (discussed next). In order to address this inaccuracy problem, we avoided the use of a rapidly increasing self-proximity cost function. Instead, as discussed previously, a discrete value is assigned - based solely on whether a pair of faces intersects or not. It is important to note that this method does not explicitly prevent self-intersection, although it allows for parametrically assigning a prohibitively high cost to it.

4.2 Segmentation Quality

Segmentation quality is of particular importance to this application. It is vital to have well-defined landmark points in order to build an effective ASM. Criteria for success in this measure include a low per-vertex segmentation error, and clear point correspondence between landmark points. Figure 3 shows that point correspondence is indeed achieved by our GDM implementation.

Another measure of model effectiveness is the *average segmentation error*. The distances between each vertex of 10 fitted meshes and the nearest point in high resolution isosurfaces generated from the target volumes that these meshes were fitted to, was measured. The mean per-vertex error (averaged over 10 fitted meshes) was 0.395mm , with a standard deviation of 0.114 . Maximum error was 1.592mm . An error histogram is shown in Figure 4. Since the mean per-vertex error is less than the 1mm resolution of the target data, we conclude that the fit was exceptionally good.

Overall, segmentation results were excellent. The GDM model succeeds in capturing the shape of the training data volumes, and is sufficient to provide the ASM with reliable landmark points.

5 ACTIVE SHAPE MODEL CONSTRUCTION

The human brain contains both a left and right caudate nucleus, and a left and right hippocampus. We therefore constructed four ASMs, each focusing on a

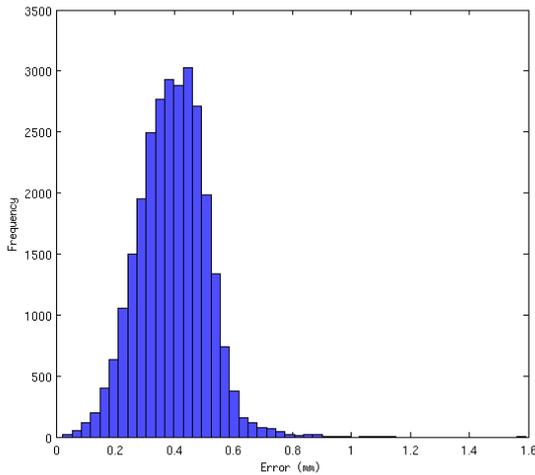


Figure 4: **Segmentation error.** A histogram showing the frequency of per-vertex segmentation error. The mean per-vertex error is less than the $1mm$ resolution of the target data.

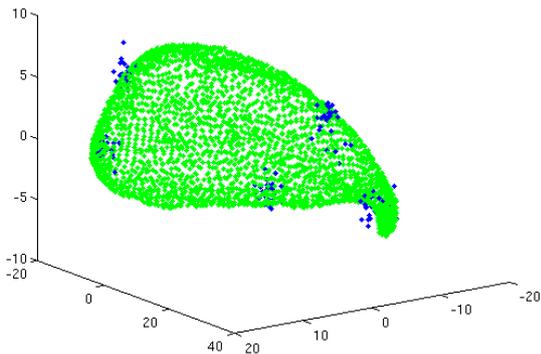


Figure 5: **Scatter Plot of Left Caudate ASM.** Selected vertices are displayed in blue, showing point correspondence between aligned training shapes.

specific structure: the left caudate nucleus, right caudate nucleus, left hippocampus, and right hippocampus. Our training data consisted of 30 left and right caudate nucleus volumes, and 26 left and right Hippocampal volumes. These volumes were manually segmented by an expert neuroanatomist, as described in Section 3. 3D landmark points were generated as described in the previous chapter, using a GDM to fit manually segmented training data for each structure. Scatter plots showing point correspondence between selected vertices of each training shape, for the left caudate nucleus, can be seen in Figure 5.

Each of the 4 ASMs is constructed in the same general fashion. Therefore, from here on, we will discuss specifically the construction of the left caudate nucleus ASM. Any differences between the construction of this ASM and others will be discussed as they arise.

	LC%	RC%	LH%	RH%
λ_1	71.8	65.4	61.8	43.8
λ_2	8.6	14.0	6.7	14.0
λ_3	3.9	5.0	5.0	6.4
λ_4	2.5	2.1	3.5	5.5
λ_5	1.6	1.3	3.3	4.3
λ_6	1.3	1.3	2.7	3.7
λ_7	1.2	1.3	2.4	2.5
λ_8	1.0	1.1	1.8	2.3
λ_9	0.8	0.9	1.5	1.9

Table 1: **Eigenvalues as a percentage of total variation.** Percentages are calculated as $P_i = \frac{\lambda_i}{\lambda_T} \times \frac{100}{1}$ for the i th eigenvector of the covariance matrix. λ_T is the total variance. *LC*, *RC*, *LH* and *RH* represent the four ASMs.

5.1 Training Shape Registration and PDM Generation

The first step in creating the ASM was to align the input training shapes, each consisting of 3D landmark points, to common coordinate axes. The next step was to create shape descriptors from the aligned shapes using PCA. The 9 eigenvectors with the highest eigenvalues are retained, thus giving $t = 9$ modes of variation, a feature vector $P = (p_1 p_2 \dots p_9)$, and a vector of 9 basis weights with which to control shape deformation, $b = (b_1 b_2 \dots b_9)$. Table 1 shows the variation ascribed to the first 9 eigenvalues of the 4 ASMs. By summing the eigenvalues of the first 9 eigenvectors of the left caudate nucleus ASM, we can see that these 9 modes account for 92.79% of the total variation in the model. Including more modes of variation did not significantly improve the accuracy of the model. Indeed, using only $t = 3$ accounts for 84.31% of the total variation.

By varying the basis weights in b , new shapes can be generated by the ASM. Figure 6 shows the effect of varying the first basis weight, b_1 , of our left caudate nucleus ASM by -3 to 3 standard deviations.

5.2 Standard Segmentation Algorithms

Our algorithm is a variation of the ASM segmentation algorithm described by [7]. This algorithm consists of 3 main stages: *initialisation*, *image search* and *parameter adjustment*. We consider the algorithm to have converged when the sum of adjustments between successive iterations is below a certain threshold.

In order to find the most effective algorithm for the image search stage, two of the most commonly used methods were implemented. These algorithms were applied to our test data, and their weaknesses analysed. Based on this analysis, we devised our own heuristic image search algorithm.

Standard Image Search: Edge Detection

In their original paper on ASMs, Cootes *et al* use Algorithm 1 as their image search method.

Some problems exist with using only edge detection to seek boundaries. One obvious problem is that

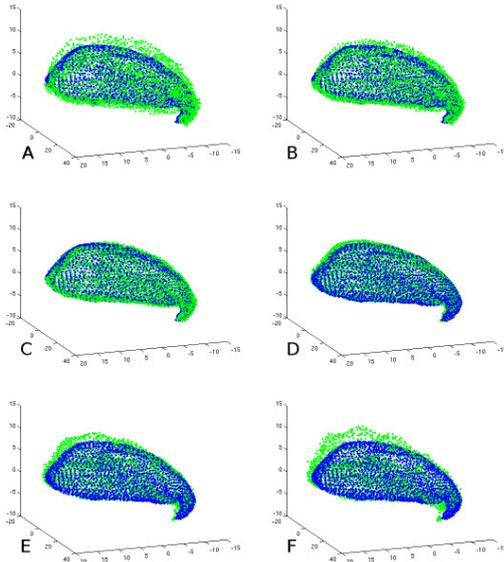


Figure 6: **Variance represented by basis weight b_1 .** Blue vertices represent the mean shape. Green vertices are displaced by varying the first basis weight by -3 to 3 standard deviations (A-F).

Algorithm 1

1. For each vertex in the ASM boundary:
 - (a) Sample the surrounding grey scale intensities along the local surface normal.
 - (b) Use an edge detector to detect the strongest edge along the sampled 1D grey profile.
 - (c) The adjustment for that vertex is calculated to be in the direction of the strongest edge, and of a magnitude proportional to the edge strength.

Cootes' Image Search

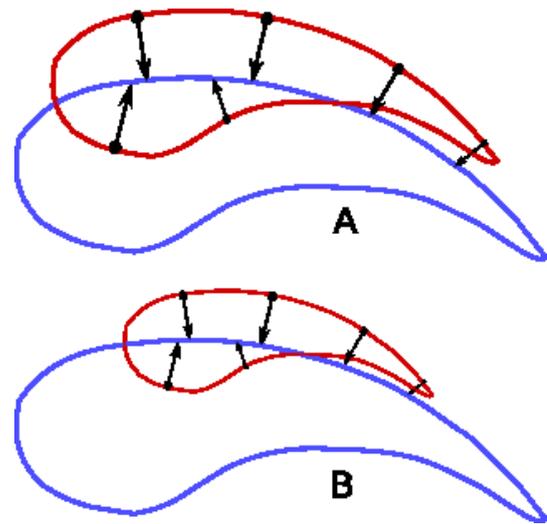


Figure 7: **Boundary Detection Problem.** Step A shows vertices from both the top and bottom of a 2D ASM being attracted to the same boundary. This causes the model to be scaled as shown in step B.

the ASM will be attracted to false positives - detecting boundaries that are not part of the target structure. If there are not too many false positives, the restriction of ASM deformation would prevent the shape from deforming outside of the ASD. However, too many false positives could result in undesirable deformation. Another problem is illustrated in Figure 7. When a valid target boundary is completely to one side of a badly initialised ASM, both the internal and external vertices of the model will be attracted to the same boundary. This target boundary is not a false positive, but it causes the model to scale itself inappropriately to attempt to fit both its internal and external vertices to the same boundary - resulting in a very small ASM.

A similar problem tends to occur in narrow regions of the target structure, such as the tail of the caudate nucleus. It is difficult to initialise the tail accurately, and in most cases the ASM's tail will be initialised next to the target tail. This presents a problem in that model boundaries will be attracted to the same target boundary - resulting in poor tail segmentation results.

Standard Image Search: Grey Profile Mahalanobis Distance

[4] describe an image search technique incorporating the use of statistical matching of grey profiles in target volumes to grey profiles sampled from training volumes. In order to use this technique, an extension must be made to the ASM generation stage. After landmark points have been allocated to a training volume, the local greyscale neighbourhood must be sampled at each of these points. In our case, fitted GDM vertices represent landmark points. A 1D greyscale profile is sampled in both directions along the local surface normal at each one of these vertices.

The grey profile Mahalanobis distance image search method relies on the relative homogeneity of the local greyscale neighbourhood surrounding each

corresponding vertex in the training volumes. When there is a high variance between grey profiles for corresponding vertices, the model will not capture the characteristics of the greyscale neighbourhood of those vertices very well. In this case, the Mahalanobis distance measure is not very effective in matching a vertex that is positioned in a target volume to its corresponding model vertex, as stored in the ASM.

6 HEURISTIC IMAGE SEARCH METHOD

In order to address the limitations of the edge detection and grey profile Mahalanobis distance image search methods, and to take advantage of various general characteristics of our two target structures, we devised a heuristic method for the image search stage of caudate nucleus and hippocampus segmentation.

6.1 Assumptions

The heuristic image search method is based on the following assumptions about the characteristics of the caudate nucleus and hippocampus. These assumptions are based on the nature of the MRI data discussed in Section 3.

- The main body of the caudate nucleus is relatively homogeneous in greyscale intensity. Based on empirical observation of test data, this intensity lies between the mean grey matter and white matter intensities for the entire brain. Specifically, the caudate nucleus is around 1.25 times the intensity of the mean grey matter intensity.
- The tail of the caudate nucleus, making up around 20% of the total structure, is darker than the body. Specifically, our test data indicates that it is around 0.7 times the intensity of the body.
- The tissue surrounding the caudate nucleus is mostly grey matter of a lighter intensity than the caudate itself.
- The entire hippocampus is relatively homogeneous in greyscale intensity. Empirical observation again shows that this intensity lies between the mean grey matter and white matter intensities for the entire brain. However, the hippocampus is slightly darker, at around 1.17 times the intensity of the mean grey matter intensity.
- The tissue surrounding the hippocampus is mostly grey matter of a lighter intensity, but some surrounding areas are darker than the hippocampus. Thus, the only assumption that can be made about colour of the surrounding tissue is that it is generally of a different intensity to the hippocampus.
- Because both the caudate nucleus and hippocampus are relatively homogeneous in intensity, they contain few internal edges. However, since the surrounding tissue is of a different intensity, there will be a pronounced edge along large portions of the boundaries of both structures.

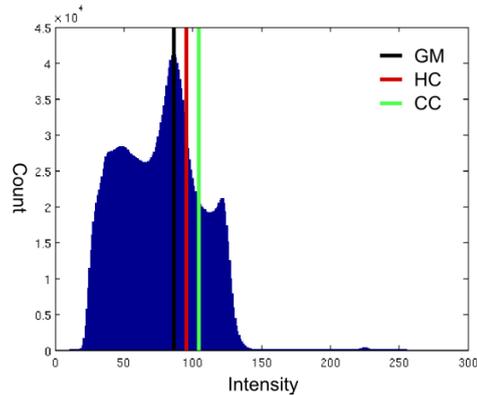


Figure 8: **Histogram of Typical Brain Tissue Intensities.** *GM*, *CC*, and *HC* refer to mean grey matter colour, caudate colour and hippocampus colour, respectively.



Figure 9: **MRI Scan of caudate nucleus.** The sagittal MRI scan shows a manually-traced left caudate nucleus in red.

- The tissue surrounding both structures is generally not homogeneous in intensity, and contains many edges.

Figure 8 shows a histogram of brain tissue intensities, with the mean colour of the caudate nucleus and hippocampus marked. Figures 9 and 10 show MRI scans of the tissue surrounding the caudate nucleus and hippocampus.

6.2 Search Function

Our heuristic image search function was designed to take advantage of the characteristics mentioned in the previous section. The goal of the function, as with the others, is to find an appropriate boundary in a given grey profile. The function works similarly to the naïve edge detection method, in that it evaluates the probability of being a part of the boundary, for each voxel in a given grey profile. The difference is that it not only takes into account edge strength of a given voxel, but also the characteristics of the (8-connected) surrounding voxels. Characteristics taken into account include the difference in colour between surrounding voxels and the target structure, as well as the number and strength of edges present in the surrounding voxels. The Heuristic Edge evaluation function, $H_E()$, is

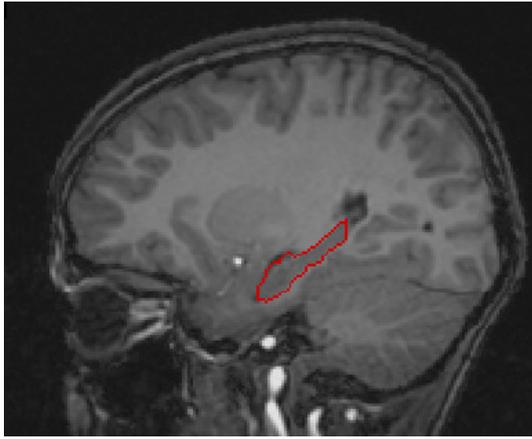


Figure 10: **MRI Scan of hippocampus.** The sagittal MRI scan shows a manually-traced left hippocampus in red.

made up of 3 weighted terms, formulated as follows. Figure 11 provides an illustration of how the heuristic Edge evaluation function works.

$$H_E(p_v, i, s) = a.E_S(p_v, i) - b.S_E(p_v, i, s) - c.D_C(p_v, i, s) \quad (2)$$

$E_S()$, $S_E()$, and $D_C()$ refer to Edge Score, Sample Edges, and Difference in Colour, respectively. These terms will be discussed in detail in the following section.

Once the voxel with the highest heuristic Edge score in p_v has been detected, the per-vertex adjustment is computed in a similar fashion to the edge detection image search method. Vertex v is moved along surface normal in the direction of the voxel with the highest Heuristic Edge score, for a distance proportional to the Heuristic Edge score of that voxel.

Edge Score

The Edge Score term, $E_S(p_v, i)$, uses a simple 1D convolution kernel ($[0.25 \ 0.5 \ 0.25]$) to find the edge strength of a given voxel, i , as it appears in a grey profile, p_v . This term operates in a similar fashion to the edge detection image search method mentioned earlier. The term is weighted by the constant a . This weighting allows for the adjustment of the contribution that each term has on the final score. This concept is discussed in further detail in Section 6.4.

Sample Edges

The Sample Edges term, $S_E(p_v, i, s)$, sums the total strength of all edges that are present in a sample that appears immediately to the right of voxel i in grey profile p_v . The sample length is specified as s . This term effectively measures inhomogeneity in the sample to the right of the voxel in question. This measure is based on the assumption that the caudate nucleus and hippocampus are relatively homogeneous, and is therefore negatively weighted by the constant $-b$.

The sample is taken to the right hand side of voxel i because of the direction in which grey profiles are

sampled along local surface normals. The surface normal orientation is such that the left hand side of grey profiles always point towards the outside of the target structure, and the right hand side always points towards the inside.

This term, when negatively weighted and combined with the previous Edge Score term, detects edges that are bordered to the right by homogeneous regions - as one would expect to find around the borders of the target structures.

Difference in Colour

The Difference in Colour term, $D_C(p_v, i, s)$, measures the difference between the mean colour of a sample - also taken to the *right* of the i th voxel - and the mean colour of the target structure being segmented. The mean colour of the target structure is determined via a lengthy process that is explained in detail in Section 6.3.

Based on the assumption that the caudate nucleus is generally darker than the surrounding areas, the Difference in Colour term only penalises the Heuristic Edge score if the sample mean is lighter than the mean caudate colour. The caudate tail is also known to be darker than the rest of the structure, so the mean caudate colour is multiplied by 0.7 when the vertex, v , corresponding to the grey profile p_v , is part of the tail.

The area surrounding the hippocampus can be both lighter and darker than the Hippocampal mean colour. So, instead of only penalising the Heuristic Edge score for the sample mean being lighter, in the case of the hippocampus, the Difference in Colour term calculates the absolute value of the difference between the sample mean and the hippocampus mean - thus incurring a penalty for the sample mean being either darker *or* lighter.

The Difference in Colour term is also negatively weighted, using the constant $-c$.

6.3 Target Colour Determination

The heuristic edge detection method relies largely on the knowledge of the mean tissue intensity of the target structure. In order to determine this intensity, we employ a lengthy algorithm, based mainly on a method devised by Worth *et al*, and to a lesser degree on the method employed by Xia *et al* [25, 26]. These methods are discussed next. Following this discussion, we describe each step of our method in detail. We then evaluate the effectiveness of our method.

Target Volume Preprocessing

In order to remove scanning artifacts and other spurious voxels from the target volume, a flood fill is first performed on the background area, outside of the skull boundary. The flood fill isolates the volume contained in the skull from the rest of the voxels, allowing for a more accurate global histogram.

The next step is to crop the volume to begin on the skull boundary. This is achieved by casting rays from

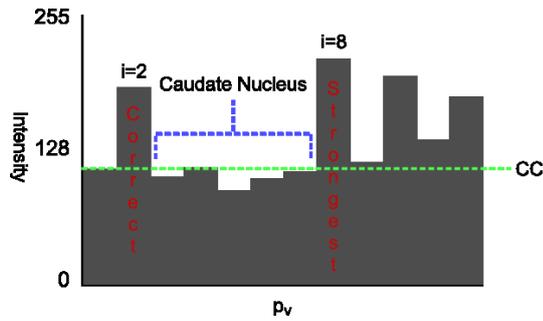


Figure 11: **Heuristic Edge evaluation.** p_v is a typical grey profile sample, taken from a caudate nucleus target volume. CC indicates the mean caudate colour, as detected by our algorithm. Using the naïve edge detection method, voxel $i = 8$ would have been incorrectly chosen (it is on the wrong side of the target structure), since it has the strongest edge value. However, the heuristic edge detection method takes into account the homogeneity of the voxels $i = 3$ to $i = 7$ (to the *right* of the sample), as well as the fact that the mean colour of these voxels is close to that of the caudate nucleus in this target volume - resulting in $i = 2$ being correctly chosen as the boundary of the target structure.

the original target volume borders, vertically and horizontally, towards the centre of the volume. The point in each axis at which the rays first intersect with the skull is taken as the cropping plane for that axis. The cropped volume allows faster processing of subsequent steps, as the volume to be processed becomes smaller.

The final preprocessing step is to remove strong edges using a Sobel filter. A 2D Sobel filter is convolved with each slice of the target volume to identify edges. Edges with scores above a certain threshold are then removed, leaving behind areas of relatively homogeneous intensity, and containing only weak gradients.

These preprocessing steps are illustrated in Figure 12.

Global Peak Detection

The first step in peak detection is to calculate the global histogram of voxel intensities for the target volume - with the background voxels thresholded out. This histogram is initially quite jagged, and must be smoothed to be useful. Smoothing is achieved by convolving the histogram with a discrete Gaussian convolution kernel. This smoothing step is repeated a number of times, depending on the width of the histogram. The width of the histogram is proportional to the range of intensities present in the target volume. If the volume contains only a small range of intensities, then smoothing the histogram too many times will cause too much loss of information, rendering it useless. Thus, the range of intensities is first determined, and the histogram is then smoothed accordingly.

The next step is identify peaks in the smoothed histogram. This is done by first convolving the histogram with a discrete Gaussian first derivative kernel. This convolution produces an approximation of the first derivative of the histogram. The first deriva-

tive approximation is then smoothed using the original Gaussian kernel. Peaks can now be identified by negative-going zero crossings of the first derivative function.

Because of the inherently jagged nature of the histogram and its approximate first derivative, many peaks will be identified by looking for zero crossings. In order to eliminate false positives, the area of each of these peaks must be determined. Peak area is determined in a similar way to Worth's method of adding the absolute values of the heights of the first derivative of the histogram on each side of the negative-going zero crossing [25]. Instead of adding the heights, we sum the area under the curve at each discrete point in the histogram, between the two zero crossings. Using this method, we can express the area of a peak as a proportion of the total histogram area - simply calculated by summing the area under the curve at each point in the full histogram. Peaks with areas that are proportionally much smaller than the histogram itself, are discarded as false positives. Our empirical testing showed that valid peaks are usually greater than $\frac{1}{30}$ th of the total histogram area.

At this point, there should be 2-3 peaks of interest left, since the dark cerebrospinal fluid can sometimes be thresholded out during the process. The grey matter peak is always the second last peak of the remaining group. This grey matter value should be saved for later comparison to the local grey matter colour, since this is the peak of interest in our segmentation algorithm. Figure 12 illustrates the peak detection process.

Local Peak Detection

Now that the global grey matter intensity has been determined, a similar process must be undertaken to find the grey matter intensity local to the area surrounding the target structure. Since both the caudate nucleus and hippocampus consist entirely of grey matter, determining this intensity will provide the necessary information for the Difference in Colour term mentioned previously.

The first step in determining local intensity is to isolate the area surrounding the target structure. Worth's method involves a lengthy heuristic procedure involving locating the Corpus Callosum, and using this as a landmark to find the caudate nucleus. This procedure is unnecessary in our algorithm, as we already have information regarding the mean location of target structures, as recorded in our training shapes. Indeed, our method already relies on this location to initialise the ASM prior to segmentation. Using the mean location and orientation of the training shapes we can reliably determine the bounding boxes surrounding both the caudate nuclei and hippocampi in a given target volume. This is done based on the previously mentioned assumption that target data will be captured in a similar orientation to the training data used to create the ASM.

After the target structure has been isolated, the algorithm proceeds to calculate peaks using the same

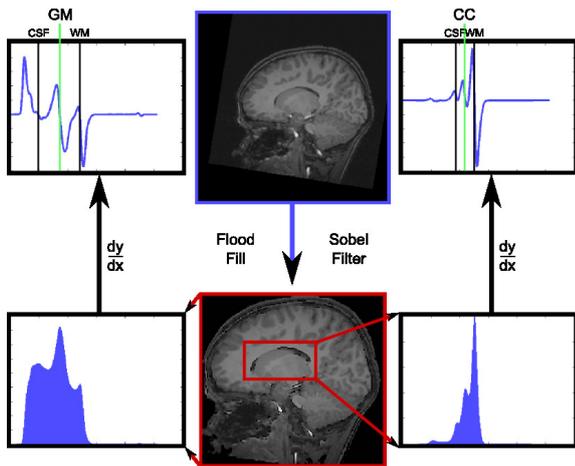


Figure 12: **Caudate Colour Determination.** *CC* indicates the final caudate colour. *CSF*, *GM* and *WM* indicate cerebrospinal fluid, grey matter and white matter, respectively. The processing steps illustrated here are discussed in detail in section 6.3.

	CN	H
<i>N</i>	62	52
Success W/O GM Guess	47%	98%
Success W GM Guess	82%	98%
Mean Intensity	88.95	80.96
Min Intensity	40	36
Max Intensity	130	109
False Positives	4	1
False Pos Rate	6%	2%
Detected False Pos	2	1
Undetected False Pos	2	0
Undetected False Pos Rate	3%	0%

Table 2: **Results of Evaluation of Target Colour Determination.** *CN* refers to caudate nucleus and *H* refers to hippocampus.

method as was used to determine global peaks. Again, the second last peak that remains after filtering is taken to be the local grey matter intensity. This peak is checked against the global grey matter peak to ensure that it is of a slightly higher intensity.

In some cases, the target volume is of such poor contrast that determination of the local grey matter peak fails. In these situations, local grey matter intensity is estimated based on global grey matter intensity. Based on our previous assumptions, grey matter belonging to the caudate nucleus and hippocampus would be estimated at 1.25 and 1.17 times the global grey matter intensity, respectively.

Evaluation

Caudate nucleus and hippocampus colour determination algorithms were tested in the following fashion. The algorithm was run on each unsegmented target volume, and the global grey matter and target structure colours were recorded. These colours were then compared to the mean target structure colour in the corresponding segmented volume. The results are summarised in Table 2.

GM Guessing refers to the estimation of target structure intensity, based on the detected global grey matter intensity. When the algorithm fails to determine the local grey matter intensity (often due to poor contrast data), or the algorithm determines an intensity that is outside of an acceptable range, the intensity is estimated according to the previously mentioned assumptions about the ratio between target structure and global grey matter intensity.

A colour estimate is deemed to be a false positive when the algorithm *does not fail* (i.e. falsely indicates success), but provides an estimate that is outside of the acceptable range of intensities for a particular target structure - based on the mean target structure intensity, as determined from manually segmented test data. False positives are detectable in most cases, by measuring whether the ratio between global grey matter intensity and local grey matter intensity is within acceptable limits. In very few cases (2 out of 62, for the caudate nucleus, and 0 out of 52, for the hippocampus), these false positives went undetected, and would therefore cause poor segmentation results.

The results show that the Target Colour Determination algorithm works satisfactorily for caudate nucleus target structures, with an 82% success rate. In cases where the algorithm fails, the ASM segmentation method would have to be altered to not rely on the knowledge of the target structure colour. This can be achieved by assigning a weighting of 0 to the Difference in Colour term. Fortunately these failures are mostly detectable, and can be compensated for.

The algorithm works exceptionally well in determining the colour of target hippocampi. With a 98% success rate, and only 1 false positive (which was detectable), the results show that the algorithm is very reliable when applied to these structures.

6.4 Parameter finding using a Genetic Algorithm

The heuristic edge detection algorithm relies on the use of various parameters to control segmentation. These parameters are detailed below.

- Profile length (*l*). This parameter determines the length of sampled grey profiles, *p_v*. Typical values range between 3 and 5.
- Sample length (*s*). This determines the sample length, as detailed in Section 6.2. Typical values range between 3 and 7.
- Term weighting (*a, b, c*). These parameters control relative term weighting, as detailed in Section 6.2. Typical values range between 0.5 and 6.0.
- Maximum edge value (*m*). This parameter controls the maximum expected boundary edge value. This varies according to target structure, and is used to control the proportional adjustment of vertices in the search function. The caudate nucleus tends to have stronger boundaries than the hippocampus, and will thus require

higher values of m . Typical values range between 50 and 500.

In order to maximise segmentation accuracy, it is necessary to find the best set of parameters for each target structure. Parameters that work well for caudate nucleus segmentation do not necessarily work as well for hippocampus segmentation. In order to test a set of parameters, a segmentation must be run using them. Since the target volume data vary in brightness, contrast, and signal-to-noise ratio, it is not enough to merely test a set of parameters on one target volume. A set of parameters must therefore be used in multiple segmentation runs to test its effectiveness. This process is considerably time consuming, and it is not feasible to experiment with all possible parameter combinations, as the search space is fairly large.

Implementation

Since exhaustive exploration of the parameter search space was infeasible, we use a Genetic Algorithm (GA) to find the best set of parameters for each target structure. The GA was implemented in MATLAB, and uses Roulette Wheel selection. Individual genes in our GA chromosomes simply represent segmentation parameters. The chromosomes therefore consist of 6 genes each. Genes consist of integer indexes ranging from 1 to the maximum number of discrete intervals in the range of values for their respective parameters. For example, the chromosome [1 2 3 6 5 4] represents the parameter values [$l = 3$ $s = 4$ $a = 1.5$ $b = 3.0$ $c = 2.5$ $m = 200$]. The fitness function takes a chromosome, evaluates it, and returns a scalar fitness value. Segmentation overlap was used as our fitness metric. A separate fitness function was used to evaluate the two target structure types, since caudate nucleus segmentation is slightly different to hippocampus evaluation.

Results

Optimum results were achieved with a crossover rate of 70% and a mutation rate of 20%. A population size of $n = 20$ was used, and the algorithm ran for 100 generations. The fittest 10 individuals were recorded for each target structure. Each of the 10 fittest chromosomes were then evaluated in a full test run of 30 and 26 target volumes, in the case of the caudate nucleus and hippocampus, respectively. For each target structure, the set of parameters that generated the best overall results was used in the final evaluation.

7 EVALUATION

7.1 Metrics

In this section, we discuss the background and our implementation of the quantitative metrics used for evaluation. These metrics include commonly-used segmentation measures such as *overlap* and *segmentation error*. Metrics such as *volume difference* and *surface area difference* offer an easy-to-calculate, but naïve

measure of segmentation success. They are not widely used, and are therefore excluded from this evaluation.

Overlap

The overlap metric, ε , also known as Dice's coefficient, is used to measure the fraction of possible voxels shared by two binary segmentations. It is formulated as follows [26].

$$\varepsilon = \frac{2V_{12}}{(V_1 + V_2)} \quad (3)$$

V_1 and V_2 are the scalar volume measures of the two binary segmentations in question. V_{12} is the scalar volume of the intersection of V_1 and V_2 . Thus, ε represents the ratio of actual overlap to possible overlap. For our purposes, the overlap metric is used to measure the success of our segmentation algorithm in identifying a target structure in a target data volume. Thus, V_1 will represent the scalar volume measure of the manually segmented ground truth data volume, G . V_2 thus indicates the scalar volume measure of the automatically segmented data volume, A . V_{12} is the scalar volume measure of the intersection of A and G , $A \cap G$, and can also be represented as $V_{A \cap G}$. This metric is used widely in segmentation literature, and can therefore be used as an objective comparison of segmentation results between similar studies [4, 27, 28].

False positive and false negative ratios can be calculated in a similar fashion. False positives, or over-estimated voxels, are those voxels that have been erroneously identified as being part of the target structure. These voxels can be described as the relative complement of A , given G , or $A - G$. The volume of false positives, V_{A-G} , and the ratio of false positives to volume of overlap, $\frac{V_{A-G}}{V_{A \cap G}}$, are also commonly-used metrics. False negatives, or missed voxels, can be described as the relative complement of G , given A , or $G - A$. Similarly, the volume of false negatives, V_{G-A} , and the ratio of false negatives to volume of overlap, $\frac{V_{G-A}}{V_{A \cap G}}$, are frequently used. False positives and negatives can also be expressed somewhat more intuitively as a proportion of the target volume, formulated as $\frac{V_{A-G}}{V_G}$ and $\frac{V_{G-A}}{V_G}$, respectively.

Segmentation Error

Average segmentation error, is a measure of the mean distance between each vertex of a fitted mesh, M_A , and the closest point in a mesh generated from a manually segmented ground truth data volume, M_G . The *mean-squared distance*, is used more often than mean distance, as squaring the difference generates more noticeable results.

One advantage of using these metrics is that, as well as measuring the total difference between all vertices, it is also possible to measure the difference between subsets of vertices - thereby allowing closer scrutiny of surface overlap in localised regions, such as the tail of the caudate nucleus, for example.

Hausdorff distance can also be used to measure how successfully the ASM has been fitted to target

Algorithm 2 Evaluation of ASM Segmentation Ability

1. For $i = 1..N_s$
 - (a) Construct ASM from N_s training shapes, leaving shape i out
 - (b) For each segmentation method, M (Edge Detection, grey profile Mahalanobis distance, heuristic method, heuristic method with multisampling):
 - i. Initialise ASM
 - ii. Segment unsegmented target data volume i , using segmentation method M
 - iii. Evaluate segmentation metrics for target volume i , method M

data [3]. This metric measures the maximum error between the boundaries of the ground truth mesh, M_G and the fitted mesh, M_A .

It is also useful to measure the *standard deviation of distances* between vertices of M_A and the target, M_G . This reflects the regularity of the segmentation results. Results with a high standard deviation would indicate an erratic segmentation, which is undesirable.

7.2 Results

ASMs based on the four target structures were constructed for this evaluation. These included ASMs for left and right caudate nuclei, as well as ASMs for left and right hippocampi. For each target structure, ASMs were constructed using varying numbers of training shapes. This was done to evaluate the effect that adding more training data to the ASMs had on the segmentation results. The caudate ASMs were constructed with 10, 20 and 30 training shapes. The hippocampus ASMs were constructed with 10, 20 and 26 training shapes. Thus, 12 ASMs were evaluated in total (4 structures \times 3 variations in the number of training shapes used). Using specific subsets of training shapes could potentially introduce bias into the experiment. Subsets were chose at random in order to minimize this. This bias effect could also be minimised by creating several small subsets of training shapes, and averaging the results obtained from them.

For each of these ASMs, Algorithm 2 was used to evaluate the ability of the ASM to segment unseen test data. As mentioned, data were considered unseen since each ASM was built using leave-one-out construction.

Statistical Significance Testing

In order to facilitate certain further statistical analysis, normality testing was performed on the results. This was done to each result set by first finding the mean and standard deviation of a Gaussian distribution that most closely matched the histogram of results, and then measuring the correlation coefficient to determine how strongly the results are correlated to the standard normal distribution.

Normality testing was performed on the overlap

	<i>Left</i>	<i>Right</i>
ED	0.7766	0.9420
GPM	0.7410	0.6683
H	0.8971	0.9390

(a) Caudate Nuclei

	<i>Left</i>	<i>Right</i>
ED	0.5439	0.8930
GPM	0.5811	0.7692
H	0.8838	0.9458

(b) Hippocampi

Table 3: Normality Testing. *ED*, *GPM*, and *H* refer to edge detection, grey profile Mahalanobis, and heuristic method. Correlation coefficient values indicate a strong correlation ($0.5 < r < 1.0$) between all results histograms and the standard normal distribution. We can therefore conclude that our results are normally distributed.

metric results of segmentation using each image search method, for each target structure. Correlation results are summarised in Table 3.

Chauvenet’s Criterion was used to identify and remove outliers in our data [29]. According to Chauvenet, an experimental result can be rejected as an outlier if the probability of obtaining the result (based on the number of standard deviations from the mean) is less than $\frac{1}{2n}$, where n is the number of results.

Out of 232 results, 37 outliers were identified - giving an outlier percentage of $\frac{37}{232} \times \frac{100}{1} = 15.9\%$. 20 of the 37 outliers occurred due to segmentation failure of target volume 6 - this accounts for $\frac{20}{232} \times \frac{100}{1} = 8.6\%$ of the total number of of results.

We use a *paired difference test* for statistical significance, aiming to disprove the null hypothesis that the means of two sets of segmentation results are not statistically different. Again, we used the overlap metric as our basis for comparison of results, since this metric is the most representative of the effectiveness of the image search methods in question. Since outliers have been eliminated, not all sets of results could be paired for comparison. Thus, we performed the dependent t-test only on samples that had corresponding pairs in all results sets. The test was performed using a significance level of 5%. A summary of results for ASMs with the largest number of training shapes is presented in Table 4.

Segmentation Results

Table 5 lists a summary of segmentation results for ASMs with the largest number of training shapes (i.e. $N_s = 30$ for caudate nuclei, and $N_s = 26$ for hippocampi).

Overlap Results Beginning with the overlap metric, it is clear, from Table 4 and Table 5, that for all four target structures, the heuristic method significantly outperformed both the ED and the GPM methods.

	μ	σ	n	$t\text{-stat}$	$p(T_j=t)$
ED-H	0.08	0.06	29	7.18	¡0.001
GPM-H	0.15	0.09	29	9.12	¡0.001

(a) Left Caudate Nuclei

	μ	σ	n	$t\text{-stat}$	$p(T_j=t)$
ED-H	0.07	0.07	25	5.13	¡0.001
GPM-H	0.12	0.12	25	5.13	¡0.001

(b) Right Caudate Nuclei

	μ	σ	n	$t\text{-stat}$	$p(T_j=t)$
ED-H	0.19	0.11	26	9.23	¡0.001
GPM-H	0.26	0.13	26	10.45	¡0.001

(c) Left Hippocampi

	μ	σ	n	$t\text{-stat}$	$p(T_j=t)$
ED-H	0.13	0.06	23	10.33	¡0.001
GPM-H	0.15	0.08	23	8.81	¡0.001

(d) Right Hippocampi

Table 4: **Results of Paired Difference Test.** *ED*, *GPM*, and *H* refer to the image search methods: edge detection, grey profile Mahalanobis, and heuristic, e.g. ED-H refers to the difference between edge detection and heuristic image search methods. μ and σ refer to the mean difference and standard deviation of differences. n is the sample size, while $t\text{-stat}$ refers to the t-test statistic value. $p(T_j=t)$ refers to the two-tail p value. All differences are statistically significant.

The GPM method performs worst out of the three image searches. When compared to initialisation statistics, we noted that the GPM method actually generates a worse segmentation over time. All other methods tend to improve their mean overlap as segmentation proceeds. This indicates a fundamental flaw with the use of the GPM method in this context. This is possibly due to boundary inhomogeneities in our training data.

False Positive and False Negative Ratios As previously discussed, *False Positive Ratio* and *False Negative Ratio* refer to the ratio of false positives and false negatives to the volume of overlap, respectively. Our results indicate that although the ED method generated acceptable overlap results, it had a tendency to over-segment the data. This is apparent from the fact that mean false positive ratios were very high, whilst mean false negative ratios remained low for all target structures. In fact, in all target structures except the left hippocampus, the ED method generated the lowest mean false negative ratios. The same method also generated the highest mean false positive ratios in all four target structures.

The GPM method generated poor mean false positive and false negative ratios for all target structures. This indicates that the area segmented by this method was more or less uniformly offset from the target structure. This lack of precision once again indicates a flaw in the use of the GPM method.

The heuristic method significantly outperformed

	<i>O</i>	<i>FP</i>	<i>FN</i>	<i>MSD</i>	<i>HD</i>	<i>SDD</i>
ED	0.74	0.49	0.13	0.97	3.76	0.63
GPM	0.67	0.26	0.37	1.79	4.09	0.83
H	0.82	0.16	0.2	0.75	2.7	0.49

(a) Left Caudate Nuclei

	<i>O</i>	<i>FP</i>	<i>FN</i>	<i>MSD</i>	<i>HD</i>	<i>SDD</i>
ED	0.75	0.42	0.16	0.84	3.28	0.59
GPM	0.69	0.28	0.32	1.28	3.72	0.73
H	0.81	0.14	0.21	0.76	2.73	0.48

(b) Right Caudate Nuclei

	<i>O</i>	<i>FP</i>	<i>FN</i>	<i>MSD</i>	<i>HD</i>	<i>SDD</i>
ED	0.57	0.69	0.32	2.19	4.83	1.05
GPM	0.5	0.46	0.51	2.82	4.76	1.04
H	0.76	0.3	0.21	0.73	3.1	0.56

(c) Left Hippocampi

	<i>O</i>	<i>FP</i>	<i>FN</i>	<i>MSD</i>	<i>HD</i>	<i>SDD</i>
ED	0.65	0.78	0.16	1.7	4.62	0.97
GPM	0.64	0.43	0.33	1.59	3.97	0.84
H	0.79	0.25	0.19	0.72	2.82	0.52

(d) Right Hippocampi

Table 5: **Summary of Mean Segmentation Results.** *ED*, *GPM*, *H*, and *HM* refer to the image search methods: edge detection, grey profile Mahalanobis, and heuristic. *Overlap (O)*, *False Positive (FP) Ratio*, *False Negative (FN) Ratio*, *Mean Squared Distances (MSD)*, *Hausdorff Distance (HD)* and *Standard Deviation of Distances (SDD)* segmentation metric means are displayed in the corresponding columns.

both the ED and GPM methods in terms of mean false positive ratios for all structures, indicating a uniform tendency to avoid over- and under-segmentation. However, the ED method generated slightly lower false negative ratios for three of the structures.

Segmentation Error Metrics As expected, mean squared distances are approximately inversely proportional to overlap. Results for the GPM method show the highest mean squared distances in most cases, indicating that the model boundary was situated relatively far from the target boundary. Again, the ED method gives moderate results, whilst the heuristic method gives the best results - resulting in an average mean squared distance of under $1mm^2$ in all cases.

The Hausdorff distance results are much the same, indicating that the maximum segmentation error was, in most cases, lowest for the heuristic method without multisampling. The ED and GPM methods fared the worst in all cases.

Standard Deviation of Distances indicates a large fluctuation in model-vertex-to-target-boundary distances for the ED and GPM methods. This indicates erratic segmentation results, with some vertices situated close to target boundaries, and some situated far away. Again, the heuristic method showed reliable consistency in segmentation, with the lowest mean standard deviation of distances results in all cases.

FreeSurfer Validity Study

A recent study was done into the validity of using the FreeSurfer software package for measurement of brain volumes in children with fetal alcohol syndrome [30]. This study was done using the same MRI test data that we used in our experiments.

The study reported inter-observer correlations for manual tracings of target structures ranging from $r = 0.94$ to $r = 0.99$, with a median of $r = 0.98$. This inter-observer correlation indicates the high reliability of the manually delimited target volumes used in our study.

The FreeSurfer package automatically reconstructs an entire cortical surface, and corresponding sub-cortical volumes from an MRI brain volume. This process is time-consuming, as target structures can not be segmented individually. Results from the FreeSurfer study indicate that manual tracings are highly correlated with automatically segmented volumes. Correlation coefficient values ranged from a worst case of $rs = 0.74$ for the right hippocampus to a best case of $rs = 0.89$ for the total caudate, with a median of 0.83.

Since overlap is directly proportional to correlation, we can compare the FreeSurfer correlations with our overlap results. Table 6 shows median and maximum results for our heuristic method that are similar to the FreeSurfer correlations, although minimum values were significantly lower for most structures. These results indicate that our heuristic method is a viable alternative to FreeSurfer. Also, when segmenting individual target structures, our method takes around 5

	<i>Min</i>	<i>Max</i>	<i>Median</i>
LC	0.65	0.88	0.83
RC	0.65	0.88	0.82
LH	0.66	0.82	0.78
RH	0.72	0.85	0.79

Table 6: **Min, Max and Median Overlap for the Heuristic Method.** *LC, RC, LH* and *RH* represent the four target structures: left caudate nucleus, right caudate nucleus, left hippocampus, right hippocampus.

to 10 *minutes*, whereas FreeSurfer typically takes 24 to 36 *hours* on equivalent hardware (due to the whole-brain segmentation). Thus, when used for rapid segmentation of individual target structures, our method may be preferable to the use of FreeSurfer.

GPM Image Search Failure

The GPM image search method failed to detect boundaries effectively. This failure was probably due to the relative heterogeneity of organ boundaries in our training data. The GPM method relies on a strong correlation between greyscale intensities of corresponding boundary points in training data. If this correlation is not strong enough (due to noise or contrast factors, for example) the GPM method fails to create a good model of the boundary surrounding a target shape. Without a decent model of the boundary, the GPM image search fails to attract vertices to boundaries, and produces a bad segmentation.

8 CONCLUSION AND FUTURE WORK

We attempt to find the best algorithm for the automatic segmentation of the caudate nucleus and hippocampus from our MRI test data, within the context of a study into the neural correlates of FASD.

We use a modified GDM to create point correspondence in our data. We then implement an ASM that incorporates various popular image search methods to find target structure boundaries. In addition to these methods, we develop our own heuristic image search method - tailored to work particularly well with the abovementioned target structures.

We use a wide range of segmentation metrics to evaluate the effectiveness of the various image search methods when applied to the data from the study into FASD. The results follow a standard normal distribution. Statistical analysis shows that our heuristic method is consistently better at segmenting the target data than other tested image search methods. The edge detection method produces average results, although it displays a tendency to over-segment the target data. The grey profile Mahalanobis method fares the worst of the three. This failure is ascribed to heterogeneity in training data. The heuristic method shows the lowest standard deviation in results, again displaying reliable consistency in segmentation. The other methods produce somewhat more erratic results. Future work may include experimenting with image

search methods not attempted in this research, such as the method suggested by [31].

Results from the heuristic method are favourable when compared to results of automatic segmentation of the same data using the FreeSurfer software package. Although the FreeSurfer results are slightly better, the ASM approach allows for independent target structure segmentation, and is therefore much less computationally demanding. Thus, individual target segmentation using the heuristic method is much faster, and may be preferable to the use of FreeSurfer. Both FreeSurfer and the heuristic method provide a lower level of accuracy when compared to manual segmentation. Since the FreeSurfer method has been used in numerous studies, such as the one mentioned in the previous section, and our results are comparable to those produced by FreeSurfer, we conclude that the heuristic method provides sufficient accuracy to be used in similar FASD-related studies.

Recent state-of-the-art techniques that make use of multi-atlas segmentation, have demonstrated comparable, and sometimes better overlap results to our ASM method [32]. Multi-atlas segmentation is generally more computationally demanding, but with advances in the availability of computational power, this becomes less of an issue. Future work in FASD segmentation could include research into the efficacy of these methods on our subject data.

In order to improve segmentation results, whilst maintaining most of the speed benefit, it may be possible to use the heuristic ASM method as the initial stage in a semi-automatic segmentation process - the fast initial automatic segmentation would then be manually corrected by neuroanatomists. This would result in dramatically improved accuracy, at a marginal cost of a few minutes of segmentation time.

All-in-all, our method will allow for more rapid segmentation of the caudate nucleus and hippocampus structures, thereby accelerating research into FASD. This important research will allow us to better understand the neural correlates of this debilitating disorder.

ACKNOWLEDGMENTS

I would like to thank Patrick Marais and Ernesta Meintjes for their excellent work in supervising and co-supervising this project.

I am also thankful to the following people for their contributions to this endeavour: Chris Warton, for performing manual tracing of the MRI test data; Sandra Jacobson, Joseph Jacobson and Chris Molteno for proposing development of an automated algorithm for segmentation of hippocampi and caudate nuclei to streamline morphometric analyses of children with FASD, as well as planning, implementing and coordinating the MRI test data collection and manual tracing.

This work was funded by NIH grants from NIAAA R01AA016781 and Fogarty R03TW007030, a Focus Area grant (FA2005040800024) from the National Re-

search Foundation of South Africa, the South African Research Chairs Initiative of the Department of Science and Technology and National Research Foundation of South Africa, and seed money grants from Wayne State University and the Joseph Young, Sr., Fund from the State of Michigan.

REFERENCES

- [1] P. Freeborough, N. Fox and R. Kitney. "Interactive algorithms for the segmentation and quantitation of 3-D MRI brain scans". *Computer Methods and Programs in Biomedicine*, vol. 53, no. 1, pp. 15–25, 1997.
- [2] Y. Xia, K. Bettinger, L. Shen and A. Reiss. "Automatic Segmentation of the Caudate Nucleus From Human Brain MR Images". *Medical Imaging, IEEE Transactions on*, vol. 26, no. 4, pp. 509–517, 2007.
- [3] D. Nain, S. Haker, A. Bobick and A. Tannenbaum. "Multiscale 3-d shape representation and segmentation using spherical wavelets". *Medical Imaging, IEEE Transactions on*, vol. 26, no. 4, p. 598, 2007.
- [4] A. Kelemen, G. Szekely and G. Gerig. "Elastic model-based segmentation of 3-D neuroradiological data sets". *Medical Imaging, IEEE Transactions on*, vol. 18, no. 10, pp. 828–839, 1999.
- [5] A. Tsai, W. Wells, C. Tempany, E. Grimson and A. Willsky. "Mutual information in coupled multi-shape model for medical image segmentation". *Medical Image Analysis*, vol. 8, no. 4, pp. 429–445, 2004.
- [6] D. Shen, E. Herskovits and C. Davatzikos. "An adaptive-focus statistical shape model for segmentation and shape modeling of 3-D brain structures". *Medical Imaging, IEEE Transactions on*, vol. 20, no. 4, pp. 257–270, 2001.
- [7] T. Cootes, C. Taylor, D. Cooper, J. Graham et al. "Active shape models-their training and application". *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.
- [8] J. Miller, D. Breen, W. Lorensen, R. O'Bara and M. Wozny. "Geometrically deformed models: a method for extracting closed geometric models from volume data". *SIGGRAPH Comput. Graph.*, vol. 25, no. 4, pp. 217–226, 1991. ISSN 0097-8930. doi: <http://doi.acm.org/10.1145/127719.122742>.
- [9] A. Ghanei, H. Soltanian-Zadeh and J. Windham. "A 3D deformable surface model for segmentation of objects from volumetric data in medical images". *Computers in Biology and Medicine*, vol. 28, no. 3, pp. 239–253, 1998.
- [10] J. Lee, S. Kim, D. Jang, T. Ha, J. Kim, I. Kim, J. Kwon and S. Kim. "Deformable model with surface registration for hippocampal shape deformity analysis in schizophrenia". *Neuroimage*, vol. 22, no. 2, pp. 831–840, 2004.

- [11] J. Pantel, D. O’Leary, K. Cretsingher, H. Bockholt, H. Keefe, V. Magnotta and N. Andreasen. “A New Method for the In Vivo Volumetric Measurement of the Human Hippocampus With High Neuroanatomical Accuracy”. *Hippocampus*, vol. 10, no. 6, 2000.
- [12] D. MacDonald, N. Kabani, D. Avis and A. Evans. “Automated 3-D Extraction of Inner and Outer Surfaces of Cerebral Cortex from MRI”. *Neuroimage*, vol. 12, no. 3, pp. 340–356, 2000.
- [13] H. William, S. Teukolsky, W. Vetterling and B. Flannery. *Numerical Recipes in C: The art of scientific computing*. Cambridge university press New York, NY, USA, 1988.
- [14] W. Lorensen and H. Cline. “Marching cubes: A high resolution 3D surface construction algorithm”. *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pp. 163–169, 1987.
- [15] J. Gower. “Generalized Procrustes Analysis.” *Psychometrika*, vol. 40, no. 1, pp. 33–51, 1975.
- [16] F. Rohlf and D. Slice. “Extensions of the Procrustes method for the optimal superimposition of landmarks”. *Syst. Zool*, vol. 39, no. 1, pp. 40–59, 1990.
- [17] R. Davies, C. Twining, T. Cootes, J. Waterton and C. Taylor. “A minimum description length approach to statistical shape modeling”. *Medical Imaging, IEEE Transactions on*, vol. 21, no. 5, pp. 525–537, 2002.
- [18] R. Davies, C. Twining, P. Allen, T. Cootes and C. Taylor. “Shape discrimination in the hippocampus using an MDL model”. *Information Processing in Medical Imaging*, vol. 4, 2003.
- [19] E. Meintjes, J. Jacobson, C. Molteno, J. Gatenby, C. Warton, C. Cannistraci, H. Hoyme, L. Robinson, N. Khaole, J. Gore et al. “An fMRI study of number processing in children with fetal alcohol syndrome”. *Alcoholism: Clinical and Experimental Research*, vol. 34, no. 8, pp. 1450–1464, 2010.
- [20] S. Jacobson, M. Stanton, N. Dodge, M. Pienaar, D. Fuller, C. Molteno, E. Meintjes, H. Hoyme, L. Robinson, N. Khaole et al. “Impaired Delay and Trace Eyeblink Conditioning in School-Age Children With Fetal Alcohol Syndrome”. *Alcoholism: Clinical and Experimental Research*, 2010.
- [21] R. Woods. “Multitracer: a Java-based tool for anatomic delineation of grayscale volumetric images”. *NeuroImage*, vol. 19, no. 4, pp. 1829–1834, 2003.
- [22] J. L. Bentley. “Multidimensional binary search trees used for associative searching”. *Commun. ACM*, vol. 18, pp. 509–517, September 1975. ISSN 0001-0782. doi: <http://doi.acm.org/10.1145/361002.361007>. URL <http://doi.acm.org/10.1145/361002.361007>.
- [23] T. Moller. “A fast triangle-triangle intersection test”. *Journal of graphics tools*, vol. 2, no. 2, pp. 25–30, 1997.
- [24] D. Eberly. *3D game engine design: a practical approach to real-time computer graphics*. Morgan Kaufmann Pub, 2001.
- [25] A. Worth, N. Makris, M. Patti, J. Goodman, E. Hoge, V. Caviness and D. Kennedy. “Precise segmentation of the lateral ventricles and caudate nucleus in MR brain images using anatomically driven histograms”. *Medical Imaging, IEEE Transactions on*, vol. 17, no. 2, pp. 303–310, 1998.
- [26] Y. Xia, Q. Hu, A. Aziz and W. Nowinski. “A knowledge-driven algorithm for a rapid and automatic extraction of the human cerebral ventricular system from MR neuroimages”. *Neuroimage*, vol. 21, no. 1, pp. 269–282, 2004.
- [27] P. Yushkevich, J. Piven, H. Hazlett, R. Smith, S. Ho, J. Gee and G. Gerig. “User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability”. *Neuroimage*, vol. 31, no. 3, pp. 1116–1128, 2006.
- [28] M. de Bruijne, B. van Ginneken, W. Niessen, J. Maintz and M. Viergever. “Active shape model based segmentation of abdominal aortic aneurysms in CTA images”. In *Proc. SPIE*, vol. 4684, pp. 463–474. 2002.
- [29] J. Taylor. *An introduction to error analysis: the study of uncertainties in physical measurements*. Univ Science Books, 1997.
- [30] S. Jacobson, C. Warton, N. Dodge, F. de Guio, C. Molteno, J. Jacobson and E. Meintjes. “Validity of FreeSurfer for measurement of brain volumes in children with fetal alcohol syndrome”. In *Proceedings of 16th Annual Meeting of the Organization for Human Brain Mapping (OHBM)*. 2010.
- [31] B. Van Ginneken, A. Frangi, J. Staal, B. ter Haar Romeny and M. Viergever. “Active shape model segmentation with optimal features”. *Medical Imaging, IEEE Transactions on*, vol. 21, no. 8, pp. 924–933, 2002.
- [32] H. Wang, S. Das, J. Suh, M. Altinay, J. Pluta, C. Craige, B. Avants and P. Yushkevich. “A learning-based wrapper method to correct systematic errors in automatic image segmentation: Consistently improved performance in hippocampus, cortex and brain segmentation”. *NeuroImage*, vol. 55, no. 3, pp. 968–985, 2011.

APPENDIX A - LANDMARK POINT GENERATION

Mesh Initialisation

The initial mesh is generated from one of the target data volumes. The Matlab isosurface routine is used to convert the binary volume into a mesh of vertices, edges and faces. This mesh is then smoothed twice using the Laplacian smoothing technique described below.

$$v_i = \frac{1}{N} \sum_{j=1}^N v_j \quad (4)$$

Each vertex, v_i , in the mesh is replaced by the mean of its 1-ring neighbouring vertices, v_j . N is the number of neighbouring vertices to v_i .

The next step is to automatically register the smoothed mesh with the target volume. This ensures that the mesh is fitted as closely as possible to the target, prior to commencing deformation. Registration is done using the Procrustes technique [15]. This technique iteratively rotates, scales and translates a candidate mesh to fit a given target mesh. An isosurface is therefore created from the binary target volume using the isosurface routine. The initial mesh is then registered against this target mesh - effectively the same as registering against the target volume itself.

After registration, our initial mesh is positioned well enough relative to the target volume to commence deformation. MacDonald's approach is slightly different since it starts with an initial low resolution mesh, which is tessellated as it deforms to fit the target. This multi-resolution approach is good for situations where the initial mesh is not well registered. It also provides a computation speedup, as the searchable parameter space is initially much smaller. The advantage of using a single resolution approach, and starting with an initial model that is similar in shape to the target volume, is that the model will deform in such a way that ensures the placement of particular vertices in similar positions on each target. This is essential to our formulation, as the generation of point correspondence between training data volumes is the purpose of this stage of the project. MacDonald's multi-resolution approach is able to generate loose point correspondence, but this is not sufficient for our purposes.

Cost Function

The cost function is similar to MacDonald's formulation [12]. It has the following formulation:

$$C(M) = (A * T_{image} + B * T_{stretch} + C * T_{bending} + D * T_{self-proximity}) \quad (5)$$

$C(M)$ is the cost function evaluating mesh M . It consists of 4 terms, each with its own weighting parameter (A, B, C and D). Ideal weightings for each

term proved to be difficult to estimate, as this process relied on trial-and-error. The output of each term is summed into a total cost value for the mesh.

Image Term

The Image term measures the sum of distances from each vertex to its nearest boundary, in both directions along the local surface normal. As in MacDonald's formulation, search distances were thresholded in order to prevent vertices from being attracted to incorrect boundaries. The equation for the Image term follows:

$$T_{image} = \sum_{v=1}^{n_v} d_B(M_v, \bar{N}_v, t) \quad (6)$$

$d_B(M_v, \bar{N}_v, t)$ measures the distance from vertex v , which belongs to deformable mesh M , along its local surface normal, \bar{N}_v , to the nearest boundary of the target volume. This distance measure is thresholded by t .

The surface normal for a specific vertex is calculated as the mean of the normals of the faces surrounding that vertex:

$$\bar{N}_v = \frac{1}{n_f} \sum_{f=1}^{n_f} \bar{N}_{v,f} \quad (7)$$

$\bar{N}_{v,f}$ denotes the normal of face f , neighbouring vertex v . n_f denotes the number of faces neighbouring vertex v .

Boundaries are detected by sampling voxels in the target volume at uniform intervals from a vertex, along the surface normal in both directions. When a change in voxel value from zero to non-zero (or vice versa) is detected, a boundary has been reached.

Stretch Term

In MacDonald's formulation, the stretch term measures the deviation in edge length between the deforming mesh and an *ideal* model. This regularisation force prevents the model from stretching or contracting into a shape that is jagged, or very different from the ideal model. We found that enforcing a predefined ideal edge length for each edge was too restrictive, since this approach prevented overall scaling of the deformable model. We opted instead for an ideal proportional edge length. The equation for the stretch term is formulated as follows:

$$T_{stretch} = \sum_{e=1}^{n_e} \left| \frac{\|M_e\| - \|Im_e\|}{\bar{M}_{edge}} \right| \quad (8)$$

M_e refers to the vector that represents edge e of mesh M . Im_e refers to the vector that represents edge e of the ideal mesh Im . n_e is the number of edges in the mesh. \bar{M}_{edge} is the average length of an edge in mesh M . It can be represented as follows:

$$\bar{M}_{edge} = \frac{1}{n_e} \sum_{e=1}^{n_e} \|M_e\| \quad (9)$$

The use of a proportional edge length allows our model to be uniformly scaled, but still constricts deformation to shapes that are similar to the ideal model.

Bending Term

Our bending term is implemented similarly to MacDonald's. The equation is as follows:

$$T_{bending} = \sum_{e=1}^{n_e} |a(M, e) - a(Im, e)| \quad (10)$$

$a(M, e)$ measures the angle between the faces adjacent to edge e , belonging to mesh M . Im again refers to the ideal mesh. This term also measures deviation from an ideal mesh. The angles between the normals of adjacent faces are measured, and the absolute difference between each angle in the deforming mesh and the ideal mesh is summed.

We use the initial, smoothed mesh as our ideal mesh. The stretch and bending terms therefore penalise deviation from the initial mesh configuration. Since our initial configuration is similar to our target shape, the GDM is prevented from deforming to shapes that are dissimilar to the target.

APPENDIX B - PRINCIPAL COMPONENT ANALYSIS IN ASM CONSTRUCTION

The PCA phase in ASM construction involves capturing the statistics of the m aligned training shapes. Each shape is described by a vector of values, which we will call a *shape description vector*. If a PDM was used in the previous stage, each shape description vector will consist of a combination of n 3-D landmark coordinates. These coordinates are projected onto $3n$ -D shape space, thereby giving one $3n$ -D vector of values for each of the m shapes. If the shape surface was reparametrised to shape descriptors, then each shape description vector will consist of a set of shape descriptor parameters. The objective of using PCA in at this stage is to find the principal modes of variation of training shapes within the ASD.

The first step of PCA is to calculate the mean shape description vector \bar{x} :

$$\bar{x} = \frac{1}{m} \sum_{i=1}^m x_i \quad (11)$$

Each shape description vector x_i is then described terms of its difference from the mean, such that:

$$dx_i = x_i - \bar{x} \quad (12)$$

The next step in PCA is to construct the covariance between each dimension across all the adjusted shape description vectors dx_i . These covariance values are stored in an $N \times N$ *covariance matrix*, where N is the total number of dimensions present in each shape description vector (e.g. $N = 3n$ for PDM-based vectors). The covariance matrix is constructed as follows:

$$C^{N \times N} = (c_{i,j}, c_{i,j} = cov(Dim_i, Dim_j)) \quad (13)$$

It can be shown that the eigenvectors of the covariance matrix with the highest eigenvalues describe the most significant modes of variation between the variables used to construct the covariance matrix [7]. Thus, the next step in PCA is to find the unit eigenvectors p_k of the covariance matrix. For an $N \times N$ covariance matrix, there exist exactly N eigenvectors. The eigenvectors $p_k (k = 1, \dots, N)$ satisfy the following equation (where λ_k is the k th eigenvalue of C):

$$Cp_k = \lambda_k p_k \quad (14)$$

Each eigenvalue indicates the amount of variance explained by its corresponding eigenvector. It is generally the case that a significant amount of variance can be explained by a small number of modes, t [7]. This enables us to approximate instances in a space of N dimensions by using only t dimensions - without losing much information as a result of the approximation. An appropriate value chosen for t should balance variation with model compactness. If t is too low, the ASM will not be able to represent finer variations in shape. Conversely, if t is too large, the ASM will contain too many parameters - creating a large parameter search space. The variance represented by t modes can be evaluated in proportion to the total variance λ_T , calculated by summing the eigenvalues:

$$\lambda_T = \sum_{k=1}^N \lambda_k \quad (15)$$

The *feature vector*, $P = (p_1 p_2 \dots p_t)$, is then created as a matrix of the first t eigenvectors. Any shape in the ASD can now be approximated by adding the mean shape description vector to the product of the feature vector and a transposed vector of basis weights $b = (b_1 b_2 \dots b_t)^T$ [7]:

$$\tilde{x} = \bar{x} + Pb \quad (16)$$

By varying the values of the weights in b , we can generate new shapes that are not part of our training set, but are also within the ASD. This is the fundamental concept on which ASMs are based. If landmark points were reparametrised prior to PCA, these will need to be mapped back into Euclidean coordinates to be of use when fitting the ASM to target data.