DEPARTMENT OF STATISTICS
University of Wisconsin
Medical Science Center
1300 University Ave.
Madison, WI 53706

TECHNICAL REPORT NO. 1101
January 30, 2005

Subpixel Morphometric Analysis of Corpus Callosum with Application to Autism

Thomas J. Hoffmann

Department of Statistics
University of Wisconsin-Madison
Madison, WI 53706

SUBPIXEL MORPHOMETRIC ANALYSIS OF THE CORPUS CALLOSUM WITH APPLICATION TO AUTISM

By

Thomas J. Hoffmann

A thesis submitted in partial fulfillment of the

requirements for the degree of

Bachelors of Science - Honors in the Major

(Statistics)

at the

UNIVERSITY OF WISCONSIN – MADISON

2004

Abstract

Autism is a neurodevelopmental disorder that affects the structure of the corpus callosum. Using the Wittleson paritition, there has been a consistent finding on an abnormal reduction of the anterior, midbody, and posterior of the corpus callosum [4]. We look at a different method of deforming the boundary of one corpus callosum, similar to that of Sebastian [26], only with a metric of affine distance and curvature rather than arc-length and curvature. We also look at the effect of different curvature estimates on these results. We see portions of the midbody, and splenium in the corpus callosum with more significant results, but only after a high degree of smoothing. The deformation field analysis shows no significant results.

Acknowledgements

Special thanks go to my thesis adviser Moo Chung, Ph.D., for the many discussions, ideas, and encouragement. Also to Grace Wahba, Ph.D. for her suggestions and discussions of splines. Thanks also to Fan Ding for the original level set code that was altered for the project, and to Yeufeng Lu for some dynamic time warping code that matched two functions by curvature. Finally, I would like to thank the people at the Keck imaging laboratory for their help and computer resources.

List of Figures

Contents

Chapter 1

Introduction

Autism is a neurodevelopmental disorder that affects the structure of the corpus callosum. The corpus callosum has been focused on to determine if there is any excess cortical connectivity resulting from from enlargement in posterior, temporal, and occiptal cortices [30]. Hardan et. al. have shown that the corpus callosum has an abnormal size [15], and Piven et. al have shown a different size in the midbody and posterior regions [23]. Using the Wittleson paritition, there has been a consistent finding on an abnormal reduction of the anterior, midbody, and posterior of the corpus callosum [4].

There have been many different methods of analyzing the structure of the corpus callosum - efforts that look at the voxels themselves and efforts that try to compare the boundaries of structures. In particular, In order to analyze structural morphometric differences, many studies have used an artificial partitioning of the corpus callosum, comparing volumetric metrics, with the Wittleson partition for example [35]. Recently, in a new direction, the white matter density in the corpus callosum has been compared by Chung et. al to get a more functional analysis [7]. Other efforts look to aligning the boundaries of images. Bajurucuiglu uses an interesting approach of Frenet distances to align the curves [2]. Davatzikos applies a two step method, in which he first establishes a one-to-one correspondence

between the points on one curve to the other curve, assuming one image can be translated to another by uniform scaling of length and length preserving bending. Secondly he deforms the curve by an elastic warping [9].

Here we look at a different method of deforming the boundary of one corpus callosum, similar to that of [26], only with a metric of affine distance and curvature rather than arc-length and curvature. We also look at the effect of different curvature estimates on these results. We extract a set of corpus callosums from the image, representing each by its boundary as a closed curve. After sufficient smoothing the curves are registered by a dynamic time warping algorithm based on curvature and affine distance. This is followed by a curvature and a deformation field analysis.

1.1 The data

The data consists of a total of 27 subjects, with 15 autistic subjects, and 12 control. All were right handed males, and of a similar age. The average age for the control was 17.1 ± 2.9 and for autistic was 16.1 ± 1.5, a fairly compatible age range. This data set was the same as that used in Chung et al. [7]; however, the ambidextrous subject was not used in this analysis.

Chapter 2

Extracting the corpus callosum

Before any sort of morphometric comparisons can be made on the corpus callosum, it must first be extracted from the MRI image. The corpus callosum may be one of the easier structures in the brain to extract, especially when restricting the problem to a two-dimensional sagittal slice of the brain. However, scanner artifacts and poor image quality can introduce unwanted noise to the image and blur the edges, making elementary non-manual extraction methods fail to capture the boundary of the shape accurately. There are many different segmentation methods that can be applied to extract structures in the brain, but none have become a standard technique. Extraction methods range from manual extraction to the much more desirable fully automated segmentation.

Fully automated techniques are based on information gained from other images that have been manually segmented. The methods are primarily focused on priors of intensity and a shape metrics. One method of extraction is the usage of active shape models, involving the deformation of a template shape to extract the data [13, 10]. Brejl proposes an automated two step extraction based on shape and border appearance. First he finds the approximate location of the object, and then uses a shape variant Hough transform to extract the object [5]. Several methods also use neural networks to build models on how to extract structures in

the brain [24]. With these segmentation methods, one must ensure that that the algorithm does not show bias towards control states, so that it accurately captures the shape of diseased states.

We chose to use a semi-automatic extraction technique that did not require any prior extraction results or a template, but required a little more user intervention, due to the data available. We have used the level set method of extraction, described extensively by Sethian in [28] and also by Osher [22]. This method takes as input the image, a seed point, and a parameter dependent upon the contrast, in order to extract the boundary of the corpus callosum. It is not as time-consuming as manual extraction, nor does it require the knowledge of a template; however, it requires some user initiation and some user intervention to ensure a properly segmented corpus callosum from the image. This proposed method of extraction is not meant to be confused with other level set methods that are more automated. There is also an alternative and more automatic level set algorithm is proposed by Leventon, who propagates the level set curve based on curvature and intensity priors set up from previously segmented data [17].

2.1 Level set methods

The level set method is basically a method of propagating a moving front from an initial surface known as the level set. In this context of extraction, the idea is to start with some initial curve inside the region of extraction, and propagate it with certain constraints so that an accurate estimate of the boundary of that region is

found. The front can be represented as follows:

$$\phi_t(x, y) + F(x, y) \left| \bigtriangledown \phi(x, y) \right| = 0$$

, where ϕ_t represents the signed distance of each pixel in the image to the front, and F describes the propagation of the front. ϕ_t is positive if it is inside the surface, and is negative if it is located outside the surface. Special forms of F can be reduced to a standard Hamilton Jacobian equation, and be solved that way. We restrict the propagation of F to advection, curvature, and attraction [28].

Computationally speaking, the algorithm can be speeded up by initially propagating the front with only a positive form of F [27]. The method works in an upwinding scheme, adding a pixel at a time until a certain gradient has been reached, as shown in figure 1. Using this method, one can start from a mere pixel or two, known as a seed point, and get a rough contour of the object. Then later tuning of this contour can be done by the usual form of the equation described above. Then this equation can be restricted to only the outer narrow band of pixels where the border is actually changing, rather than on the entire surface [20], as depicted in figure 2.

2.2 Level set application

2.2.1 Image processing

In practice, some amount of noise removal can facilitate the propagation of the front and yield better extractions of the curve. Simple Gaussian smoothing works, but tends to blur edges too much. Better methods involve edge detection with

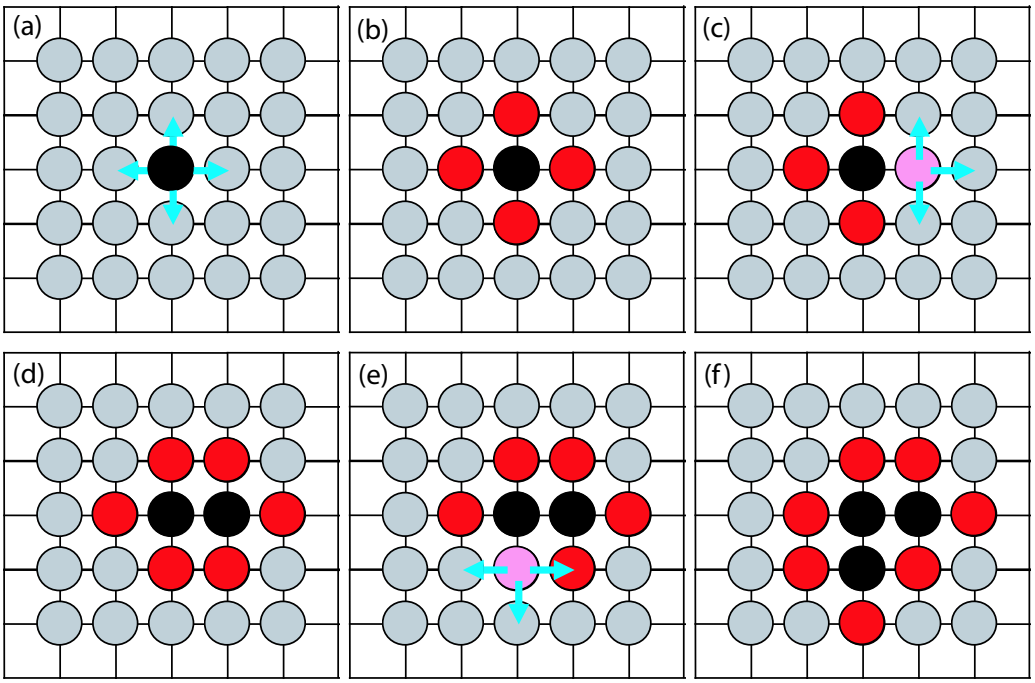

Figure 1: This illustrates how the fast marching technique propagates the moving front. **(a)** The initial value is updated "downwind" of the current frontier (just a pixel here). **(b)** New values are computed at the frontier. **(c)** The pixel on the fringe with the minimum value is chosen. **(d)** The neighbors of that minimum valued pixel are updated. **(e)** Again the minimum valued pixel is chosen. **(f)** Neighbors downwind are updated, and the process continues. [27]

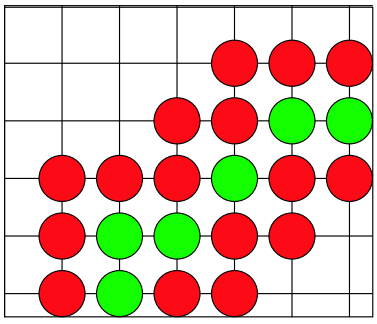

Figure 2: Figure illustrating the narrow band of pixels that need to be updated using the level set algorithm [20]. The current boundary is shown in green, and the narrow band in red surrounds the pixels.

smoothing; here we used Malladi and Sethian's min/max flow formulation of the level set equation applied to image processing [18, 19]. Results of one of the segmentations is shown in figure 3.

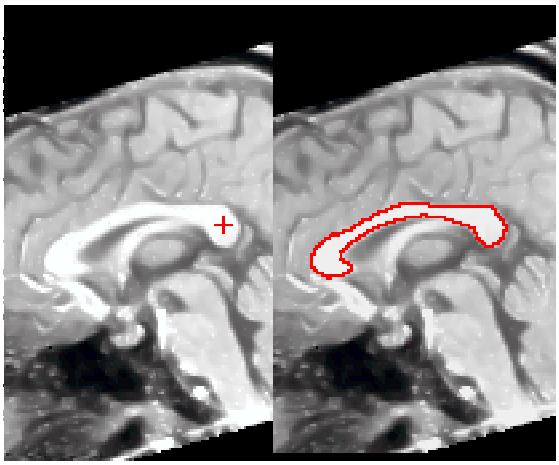

Figure 3: The image on the left shows a portion of a brain image slice with the seed points on it. The image on the right shows how the level set method extracted the image.

2.2.2 Land mines

In practice, with our implementation of the function F, we found that there were some cases where the image intensity gradient between the corpus callosum and surrounding tissues were similar enough to the gradient within the corpus callosum that the corpus callosum could not be extracted without manual intervention. Thus we introduced a notion of land mines in the data - manually inserting high image gradients to halt the flow of the object. This was only needed on 4 of the 27 subjects. Results of applying this to the worst case image in the group is shown in figure 4.

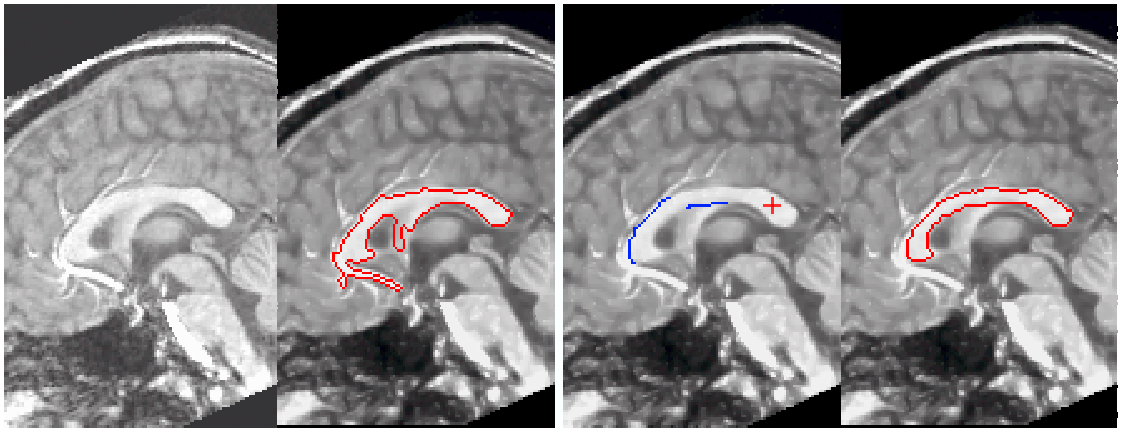

Figure 4: This image shows one of the worst case types of images to extract the corpus callosum from, as the image gradient is not sharp enough towards the left. From left to right we have the original image, the area that the level set method starts to leak out into if not corrected, the image with seed-points in red and land-mines in blue, and the final extracted structure.

2.3 From pixels to euclidean curve

The next step in preprocessing the data is converting the discrete pixelated data into a continuous domain. We follow the outer boundary of the pixels extracted from the image in the following algorithm to come up with the initial clockwise contour of the corpus callosum.

1. Formulate the closed curve into a clockwise ordering of the points, indexing these points $p_1, \cdots, p_n$.

2. Move clockwise around the *outer boundary* of the points, traversing diagonal and across as necessary, so as to touch as few points as possible and only on the outlying frontier. For each of these points p_j, if there are neighboring points $p_{j_{nbd}}$ touching a face of the pixel then average that value into the

current midpoint of the pixel; all adjacent pixels that only touch the corner are ignored.

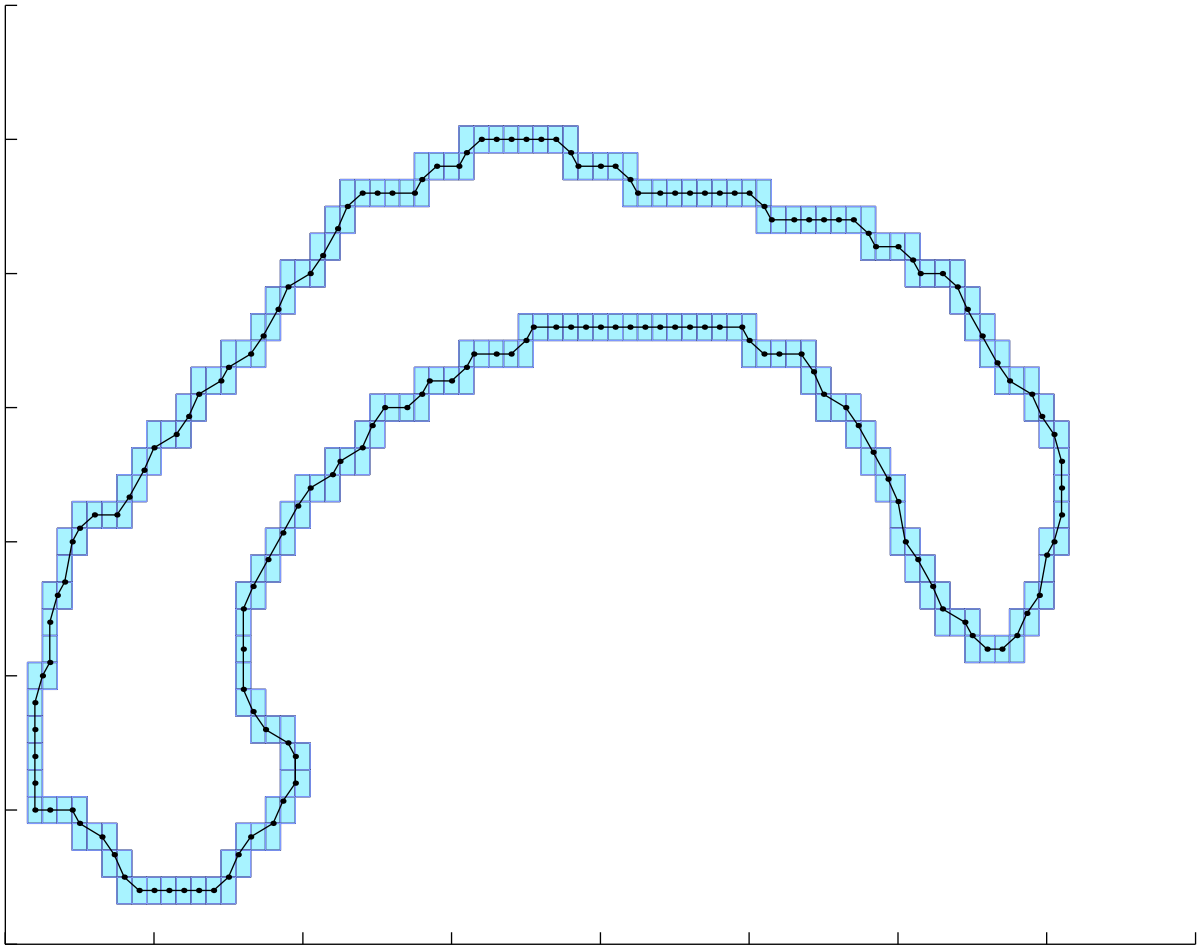

Figure 5: Initial curve fit to the pixelated data, to move from a discrete space to a continuous space.

The results of fitting the curve with such an approximation in figure 5. There is still a good amount of partial volume effect to deal with before this can be considered a reasonable approximation of the shape of an individuals corpus callosum.

However, the function of this step, is just to move the disorganized pixels into a clockwise curve in the real plane in a reasonable fashion. The data needs a good amount of smoothing to mitigate the partial volume effect before any morphometric analysis can be done.

Chapter 3

Smoothing

The pixelated image that we have extracted from the image does not provide the best estimate of the boundary of the corpus callosum, due to the partial volume effect. From figure 5 it is clearly not a smooth boundary of the image. Flynn and Jain, though referring three-dimensional curvature, state that quantization can affect estimates even more than measurement noise, and emphasize the necessity of smoothing such objects when estimating curvature [12]. Here the goals are not only to achieve a smooth boundary estimate, but also to obtain an accurate estimate of curvature for analysis and registration. Thus not only do we need to smooth the object, but methods of choosing the smoothing parameter based on the assumptions of random noise may fail to estimate a strong enough smoothing parameter.

There are many different smoothers available to attenuate the partial volume effect. They provide a smooth estimate of a closed curve in $\Re^2$ representing the shape of the corpus callosum. The principle difficulty in smoothing the curve is determining how much to smooth the curve, as all smoothing methods have a means to choose how much to filter the image. Some methods of smoothing may be better than others depending on the application. Here we explore three different methods of smoothing the curve without shrinkage: Taubin's surface fairing applied

to a two-dimensional curve, piece-wise spline smoothing, and parametrized spline smoothing. We compare each method and look at the effects of the smoothing parameter.

3.1 Taubin's Smoothing

The first method of smoothing the data is based on Gabriel Taubin's surface fairing method [29], though implemented in two dimensions here rather than in the usual three. Surface fairing is just a variant on Gaussian smoothing that prevents shrinkage. For instance, consider a circle, and apply Gaussian smoothing; clearly the circle will shrink. The closed curve is topologically equivalent to a circle; similarly, a closed curve will shrink toward the center under normal Gaussian smoothing as well. Taubin's method of smoothing prevents such shrinkage by adding an extra "unshrinking" step.

Taubin's method is formulated as follows: Let K be the circulant matrix, that is

$$K = \frac{1}{2}\begin{pmatrix} 2 & -1 & \cdots & -1 \\ -1 & 2 & \cdots & \\ & & \cdots & \\ -1 & \cdots & -1 & -2 \end{pmatrix}$$

Then if we let $\mu < -\lambda$, $0 < \lambda$ be our smoothing parameters, we define

$$f(k) = (1 - \lambda k)(1 - \mu k)$$

so that our n^{th} iteration of smoothing is just

$$x^N = f(k)^N x$$

.

Taubin provides the following suggestions in how to choose all of the parameters. First we choose k_{PB}, known as the pass-band frequency, N, the number of iterations, and the smoothing parameter λ. Then

$$\mu = \frac{\lambda}{k_{PB}\lambda - 1}$$

. Finally, Taubin suggests that one chooses $k_{PB} \in [0.01, 0.1]$ [29]. In practice, we see the results of such smoothing in figure 6 and figure 7. Since all images are pixelated to a similar extent, the smoothing parameters amongst the subjects should be roughly the same. More will be discussed in the curvature estimation section of this chapter, as the smoothness of the curve affects not only the boundary but the curvature estimation as well.

3.2 Piece-wise Smoothing Splines

An alternative method to filtering the data is to apply a smoothing spline. Smoothing splines are covered with a good theoretical basis in Wahba [31], and a little more applied in [11, 14]. A smoothing spline can be formulated so that it maintains a smooth m^{th} derivative. However the general forms of cross validation may smooth up to the m^{th} derivative, but penalize based on the fit rather than on the smoothness of the derivative. Thus although it may be a smooth derivative, it may not perform very well [32]. Here we intend to estimate curvature from the spline, which requires up to a second derivative. Hence we use quintic splines to keep a continuous second derivative. However, GCV and CV penalize the fit of the curve to the data, and assume random noise.

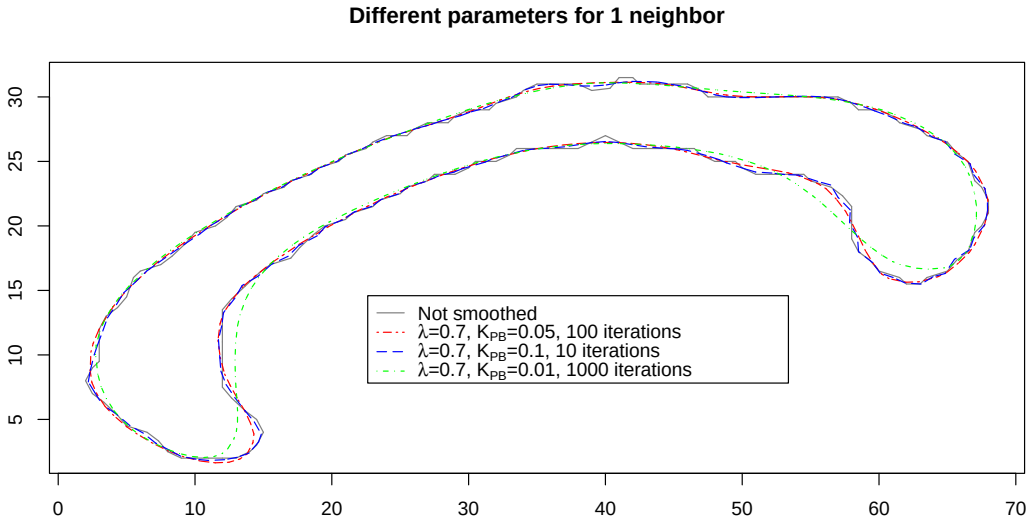

Figure 6: Various parameters for Taubin's method with a neighbor radius of one point.

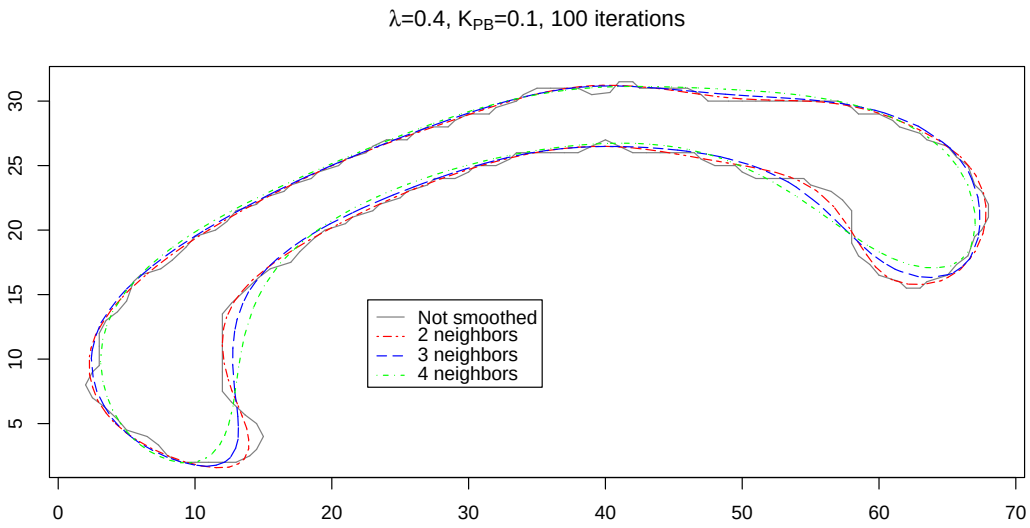

Figure 7: Various neighbor radii for Taubin's method with fixed parameters.

The general form of a spline equation defined on monotonic values t_i, is

$$S(g) = \sum_{i=0}^{n} [Y_i - g(t_i)]^2 + \lambda \int_a^b g^{(m)}(x)^2 \, dx,$$

where λ is the smoothing parameter, and $m = 2$ means a cubic spline, $m = 3$ a quintic spline, $m = 4$ a septic spline, and so on. However, this formulation of a spline can only fit data where the values are monotonically increasing, so it cannot go around all the bends in our curve and reverse directions. However, we can clearly fit pieces of our curve with splines.

3.2.1 Construction of the Piece-wise Spline

We construct such a piecewise spline as follows. First, let us consider ourself in the standard $\Re^2$ with axis x and y with our closed curve on it. Define the cut-points

$\{c_{(i)}\}$ to be the set of points where, when following the curve clockwise in a circle, the curve changes direction in the x component. This set $\{c_{(i)}\}$ is ordered clockwise around the curve. Then between any set of cut points, as long as there are at least $2m + 1$ points, we can fit a spline; if there are not enough points, the spline is left undefined. However, because of this formulation, the estimation of the spline at these cut-points will not be using all available information, and will have a very poor estimate of the fitted value at this cut point. See the example in figure 8 and figure 9 for more details.

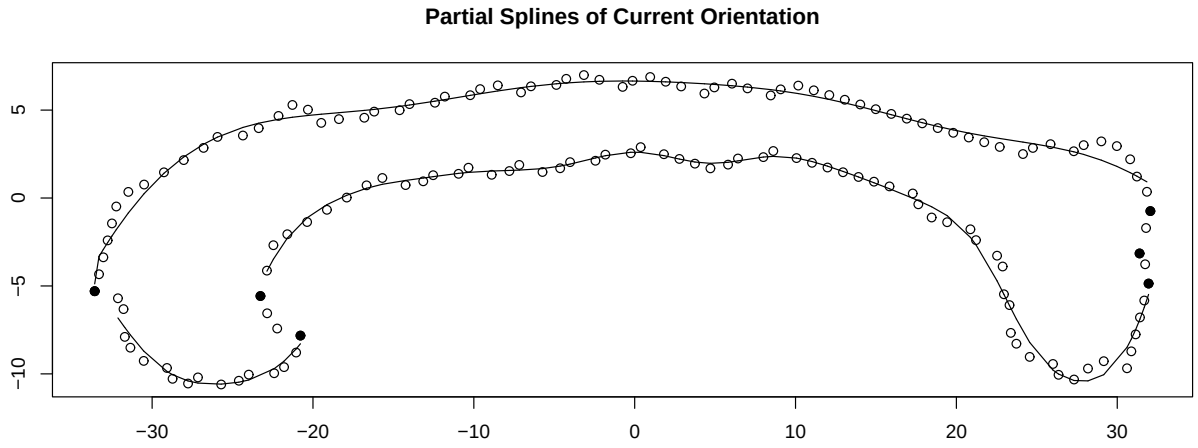

Figure 8: Partial spline fit of a curve in the standard orientation. The dark circles indicate cut points.

To remedy this situation, consider rotating the curve by $-\pi/2$ on its center, so that one will find a different set of cut-points and a different set of splines, as in figure 9. The fitted values can be back transformed to the original data to fill in the missing data, and correct other poorly estimated data in the overlapping

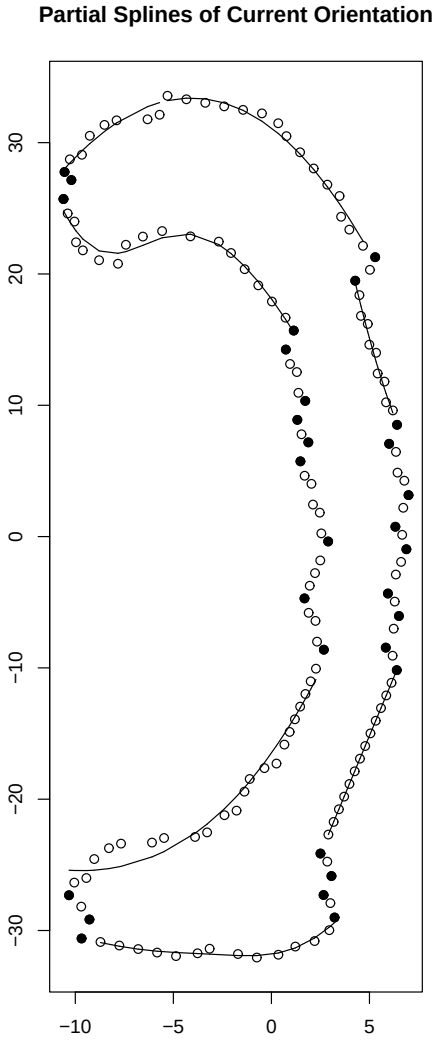

Figure 9: Partial spline fit of a curve in the rotated orientation. The dark circles indicate cut points.

regions.

So now that there are two different spline fits to the data, the fits must be combined into one smooth curve to represent the aggregate spline smoothed data. Since the fit at the cut-points is not very good, and worse yet, there may be places where the spline is defined, we introduce a weighted average of the values to determine the final estimate. It seems natural to define the weights w_i of a point p_i on the curve given by one of the splines

$$w_i = \begin{cases} 0 & \text{, if } p_i \text{ is undefined} \\ \text{minimum of the arc lengths to respective cut-points} & \text{, otherwise} \end{cases}$$

Thus when a point is not defined or right at a cut point, it will have no weight, and the maximal weight a spline can have is determined by its length. This appears to be a good method of computing the curvature, though the smoothing parameter may not be optimized correctly.

This construction can be extended in general to any number of rotations of the data for fitting more complicated surfaces and ensuring the proper fit. For instance when fitting the piece-wise spline for the corpus callosum data, we used 7 rotations, though there was not really any noticeable difference among other numbers of rotations, once beyond two or three rotations.

3.2.2 The smoothing parameter

Again the choice of smoothing parameter is very important, and the principle difficulty in applying the method. If it is too large it will be roughly equivalent to a linear regression estimate of the data, while if it is too small it will be roughly

equivalent to the data [16]. A spline assumes random noise, but here we have pixelated data, which will be a slightly different distribution. Several different choices are depicted in figure 10. Here the GCV and CV tend to leave the data a little under-smoothed, following the pixels rather than the boundary. Thus we show several other parameters in the figure, the choice used at $\lambda =$, and parameters that start to over-smooth the data. This slight over-smoothing is particularly noticeable in the areas where the curvature changes more rapidly as in the Rostrum and lower left portion of the Genu in the corpus callosum [1] when using a uniform smoothing parameter throughout the subject.

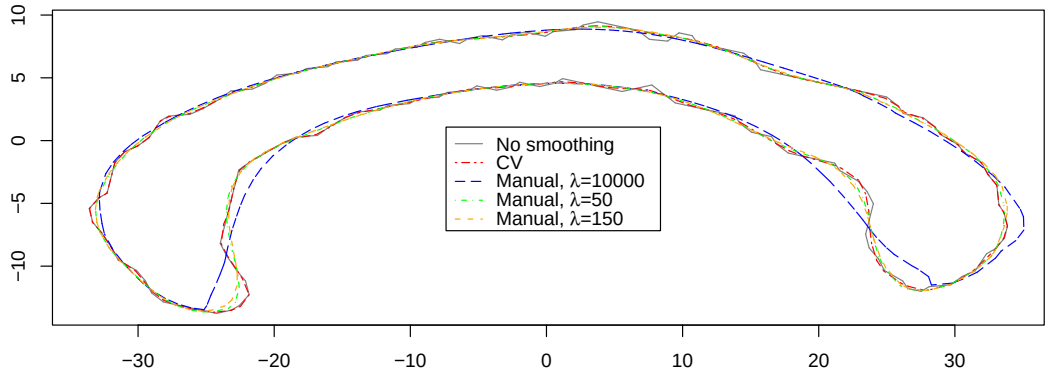

Figure 10: Here we have the results of several different smoothing parameters. The GCV smoothing parameter is not included in this figure, as it falls almost exactly on top of the CV chosen smoothing parameter.

3.3 Parametrized Spline

3.3.1 Theory

An alternative spline-smoothing method of smoothing the data has been proposed by Carew et. all [6]. First the closed curve is parametrized discretely onto a circle $C(t) = (x(t), y(t))$, where t indexes around the boundary of the curve. Then each of the x-coordinates and y-coordinates can be smoothed with two different splines, providing a smooth estimate of the boundary. As with the other spline, we will be able to calculate the curvature from the spline fit.

3.3.2 Smoothing parameter

The choice of the smoothing parameter here provides similar results to the previous method. Again we see problems with GCV and CV under-smoothing the data in figure 11.

3.4 Comparing the smoothing methods

It is difficult to assess which smoothing algorithm smooths the data the best based on visualization alone. In figure 12, we see that the smoothing results can be more or less aligned when given the proper smoothing parameters. In order to further asses the smoothing methods, we will look at how curvature can be estimated from them in the next session.

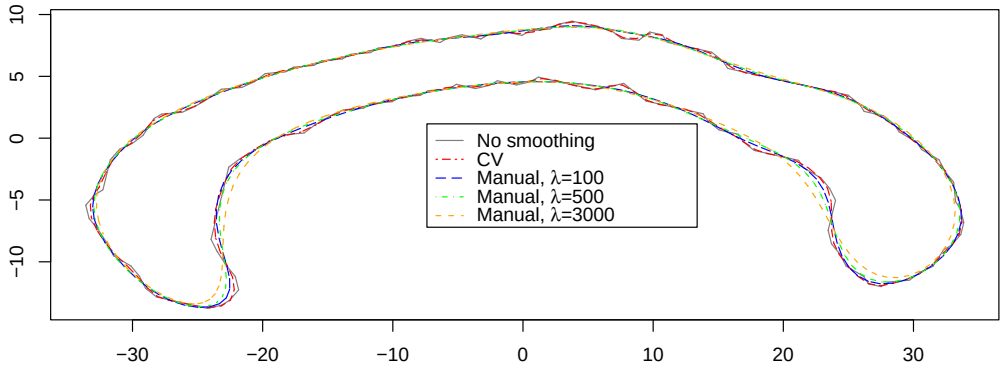

Figure 11: Results of several different smoothing parameters for the parametrized smoothing spline.

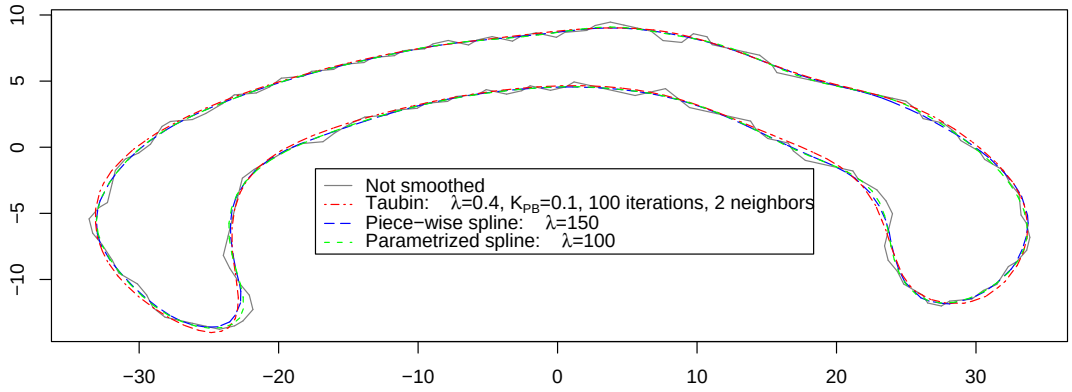

Figure 12: Similar results of smoothing a closed curve with the three smoothing methods.

Chapter 4

Curvature

A curvature estimation that is robust yet powerful enough to detect differences in shape is very difficult to obtain, especially if the data is noisy, and may be even worse if it is quantized [12]. With too much smoothing, we lose power for comparison. Yet with too little smoothing, the data becomes too noisy to detect signal, and be more likely to show false positives that may even reflect the quantization into voxels. We explore three different methods of evaluating curvature. The first method uses a least squares fit of a circle to the data to estimate curvature. The other two methods use the formulations of the splines to estimate the curvature. We also look at the effect of the smoothing parameter on the estimate.

4.1 Least Squares Estimation of Curvature

4.1.1 Theory

The first estimation of curvature is based on the idea of curvature on a circle, which is just the reciprocal of the radius. First, let $p_1, p_2, \cdots, p_n$ represent the points on the closed curve so that $p_0 \equiv p_n, p_1 \equiv p_{n+1}$, and so on. Then the curvature at point $p_i = (x_i, y_i)$ is determined by fitting a least squares estimate of a circle through its

neighbors $p_{i-k}, \cdots, p_i, \cdots, p_{i+k}$ which we define to be of a neighborhood width k. Then we can formulate an equation for our circle with center (a,b) for each (x_j, y_j) in the neighborhood as

$$(x-a)^2 + (y-b)^2 = (x_j - a)^2 + (y_j - b)^2.$$

Simplifying, we can formulate a matrix of all of these equations that can be solved by the generalized inverse for the center of our circle

$$2\begin{pmatrix} x_i - x_{i-k} & y_i - y_{i-k} \\ \cdots & \cdots \\ x_i - x_{i+k} & y_i - y_{i+k} \end{pmatrix}\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} x_i^2 - x_{i-k}^2 + y_i^2 - y_{i-k}^2 \\ \cdots \\ x_i^2 - x_{i+k}^2 + y_i^2 - y_{i+k}^2 \end{pmatrix}.$$

So that the curvature is just

$$\kappa(p_i) = ((x_i - a)^2 + (y_i - b)^2)^{-1/2}.$$

The least squares estimate is very sensitive to any sort of noise on the curve that you are estimating from. Thus, to get a better estimate of the curvature, the data must be smoothed sufficiently. When fitting the Corpus Callosum data with the least squares estimate of a neighborhood width of 2 as defined above (for a total of 5 point), Taubin's smoothing was first method used in conjunction with least squares estimation. The results of the curvature fit on respective choices of parameters are as depicted in figure 13 and figure 14. These results do not look as smooth as the two methods generated from the smoothing splines, shown in figure 15 or figure 16.

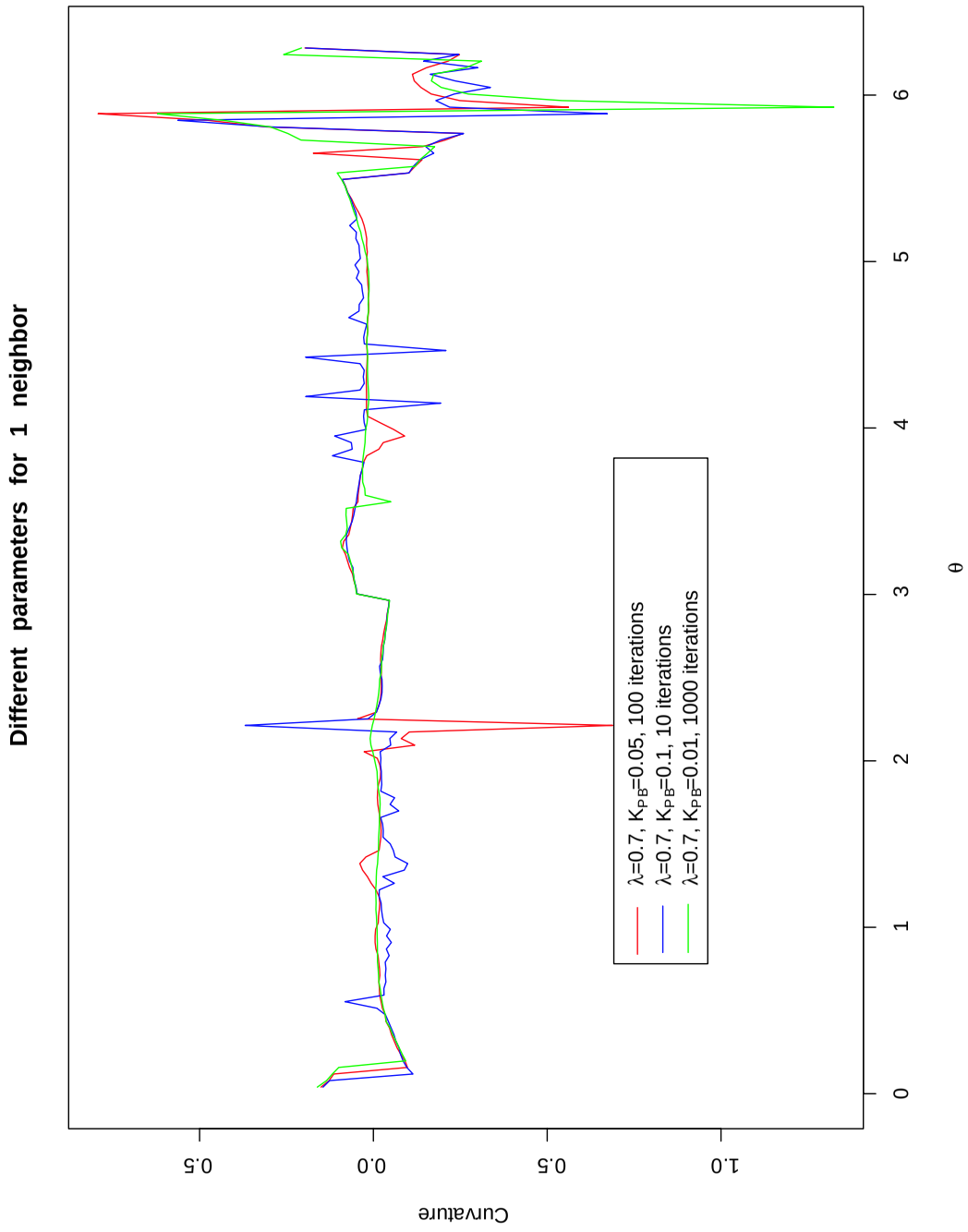

Figure 13: Least squares estimation on curves smoothed by Taubin's method.

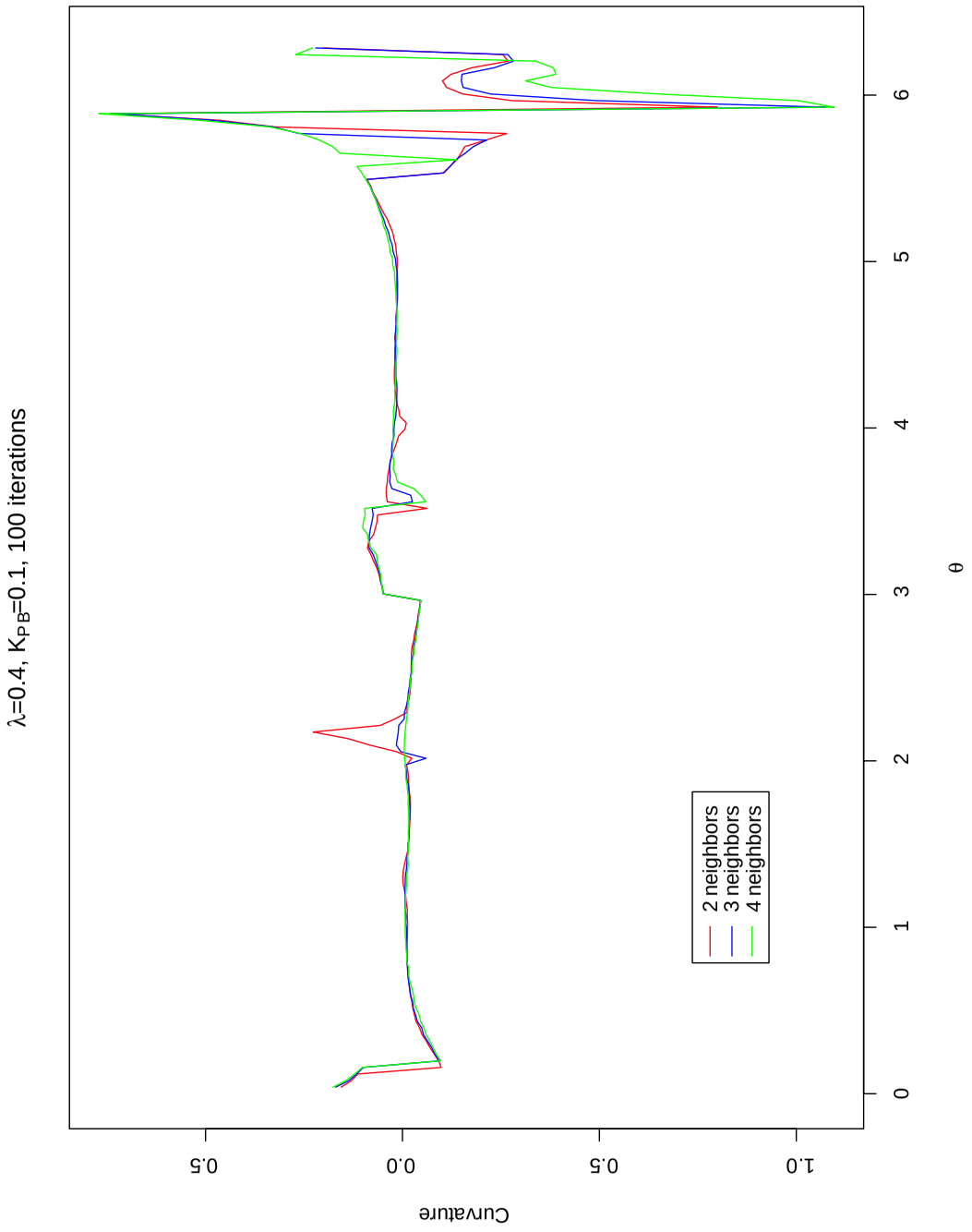

Figure 14: Least squares estimation on curves smoothed by Taubin's method.

4.2 Piece-wise spline curvature

One benefits of using the spline is that the estimate of curvature can be obtained from the spline used to smooth the data. From the fitted spline we can estimate the first and second order derivatives, and use that information to calculate the curvature [34]:

$$\kappa = \frac{y''}{(1 + (y')^2)^{3/2}}.$$

Results of this estimation are shown in figure 15. Here we clearly see that as the smoothing parameter λ of the curve increases and the curve is smoothed more, there is a corresponding smoother estimate of curvature.

4.3 Parametrized spline curvature

Curvature for the parametrized spline is computed similarly to that of the piece-wise spline. We use the following formula to compute the curvature [34]:

$$\kappa = \frac{\frac{d\theta}{dt}}{\sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dx}{dt}\right)^2}}$$

Again we see a good correspondence between the smoothness of the curvature estimate, as depicted in figure 16

4.4 Curvature comparison

The results of all of the curvature estimators are shown in figure 17. It is difficult to assess which curvature estimate is the best here, but the two spline methods appear to be more stable. The spline methods show a good correspondence among

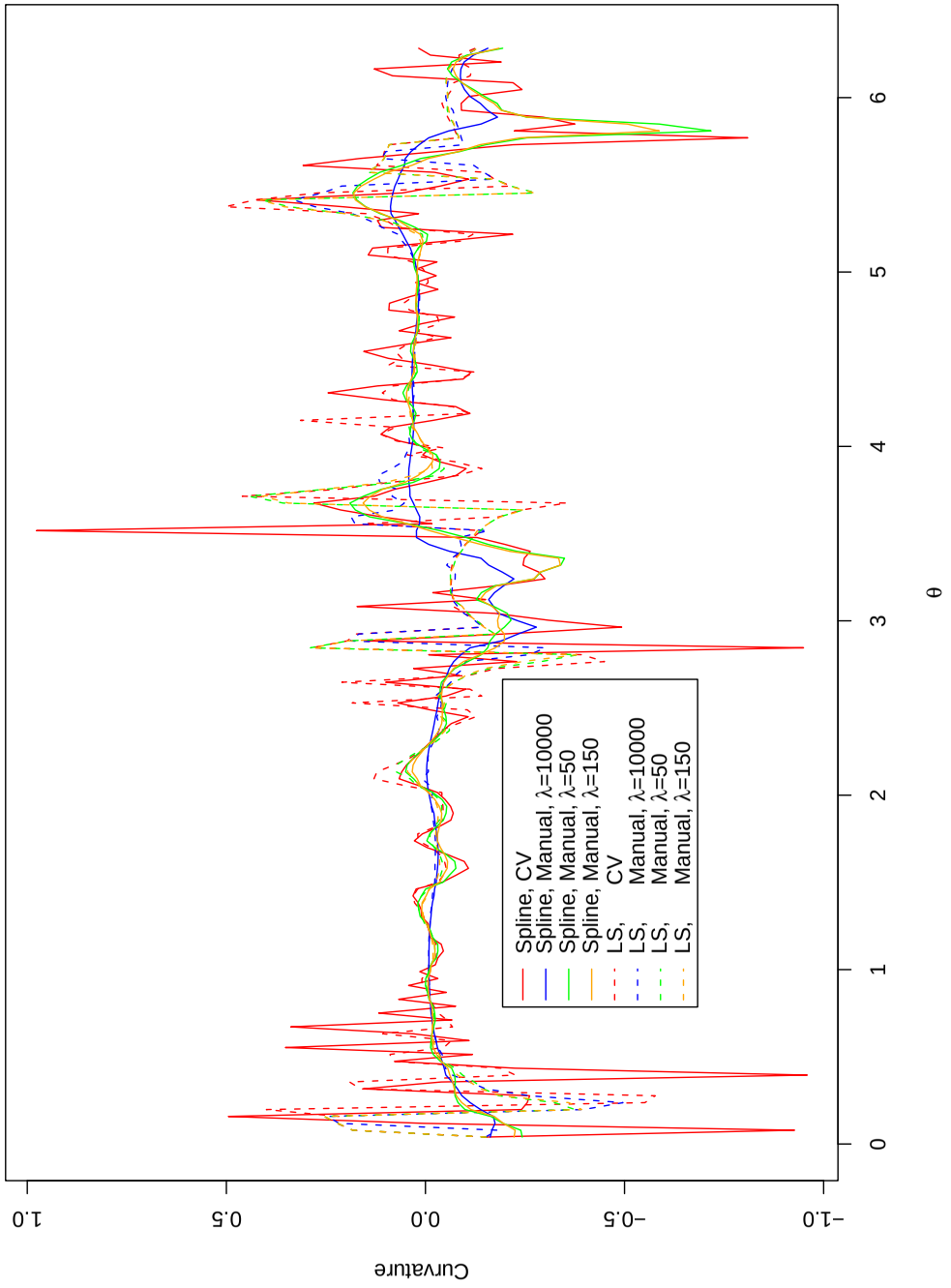

Figure 15: The curvature of one subject after being smoothed by the piece-wise spline smoothing method. We have the results of using the derivative of the spline and from least squares estimation (LS).

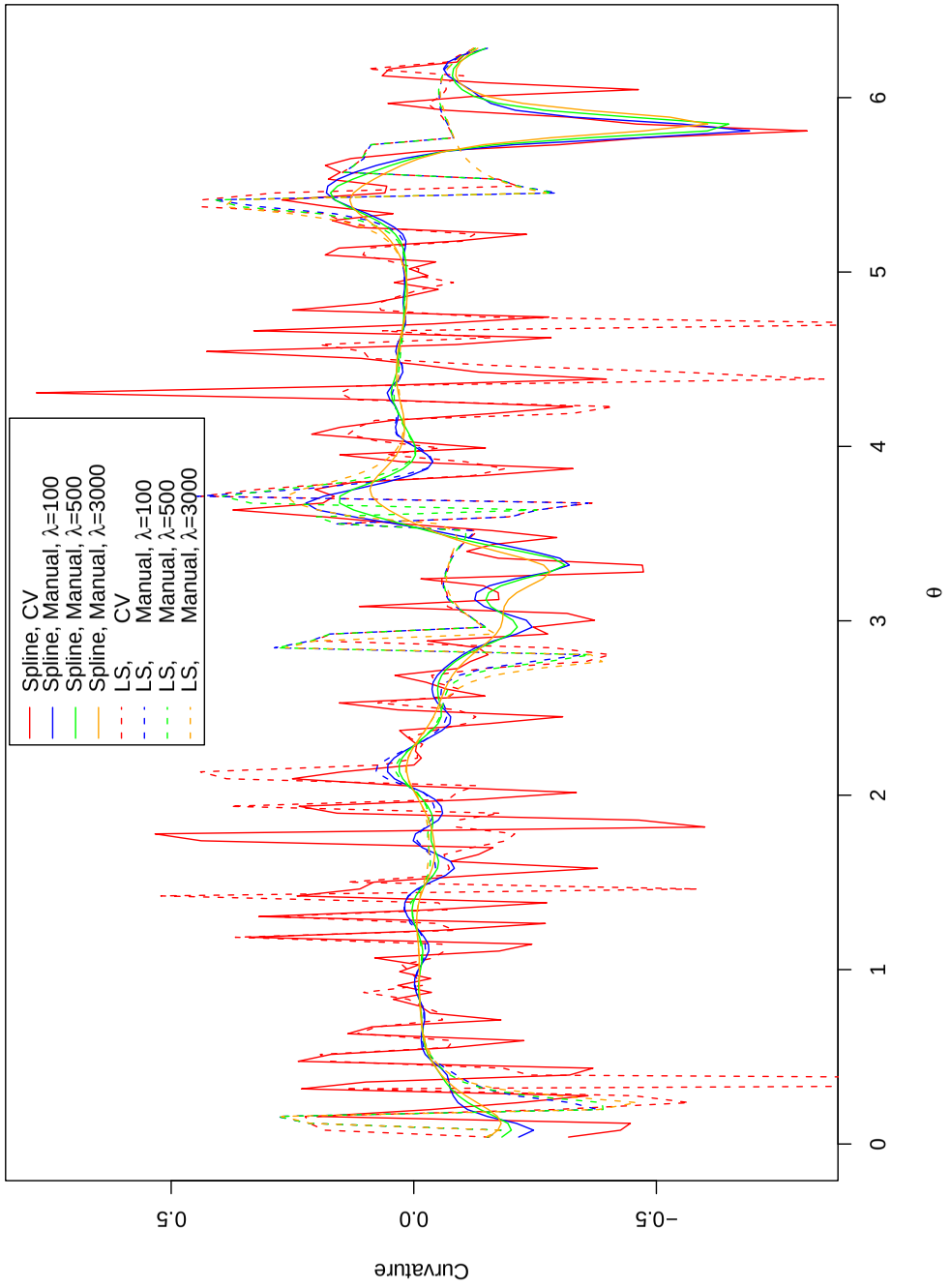

Figure 16: The curvature of one subject after being smoothed by the parametrized spline smoothing method. We have the results of using the derivative of the spline, and also from the least squares estimation.

the smoothing parameter, the smoothness of the curve, and the smoothness of the curvature. As for choosing amongst the two spline methods, the parametrized spline may seem a slightly more natural approach to some. However, with automated smoothing parameter choices, the piece-wise spline can be formulated to choose a different parameter for each piece, thereby obtaining a more spatially adaptive filter.

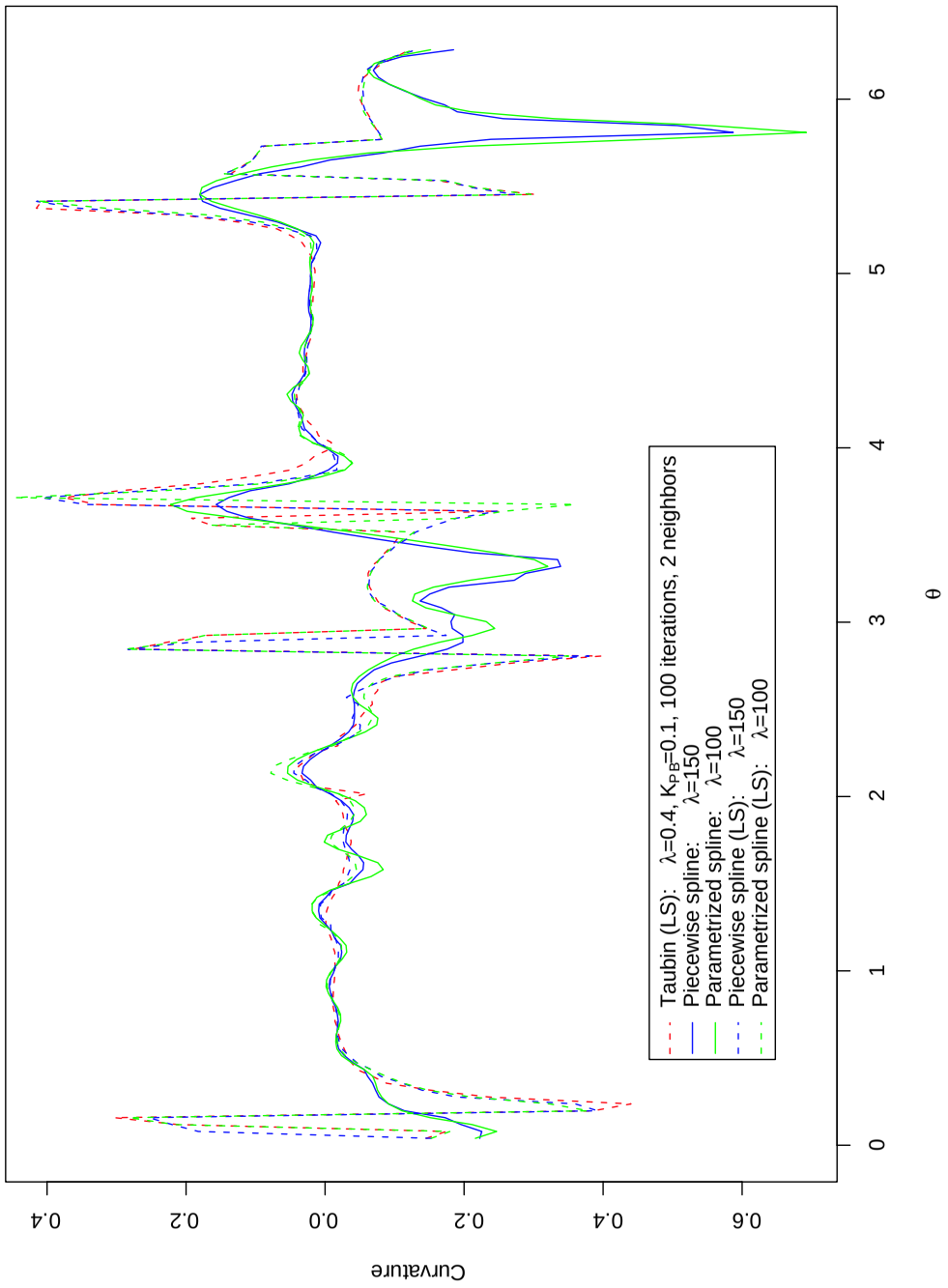

Figure 17: The curvature of one subject measured with the three different methods.

Chapter 5

Registration

Before any morphometric analysis can be considered on the smoothed curves, they must be registered so that one corpus callosum curve is deformed to another curve, aligning homologous regions of the curve. An optimal alignment between curves is desired, but such an alignment is not well-defined as different instances require different features of curves to be matched. In the brain there are no obvious features on the corpus callosum to align all of the points on one curve to another curve, though there are certain extremas of curvature and overall shape that allows our eye to distinguish the shape and match portions of curves together.

Here we present a dynamic time-warping algorithm to deform and align curves, based on curvature and an affine distance from one curve to the other to stabilize. These costs are invariants that do not change under rotation or translation. For a theoretical approach to dynamic time-warping see Bellman and Dreyfus [3], and for applications to curve matching see Wang and Gasser [33], Sebastian [25, 26]. The dynamic time-warping algorithm minimizes our distance metric to be based on curvature and affine distance.

5.1 Pre-registration Affine Alignment

One method of obtaining an affine alignment involving only scaling and rotation is to pick some number of control points and use least squares estimation to align those points. We use a method that begins with this as an initial step to align the curves, choosing those control points computationally by extrema of curvature and some knowledge of the shape.

The next step intends to refine this alignment by minimizing the distance from one curve to the other. Let $C_1, \cdots C_n$ represent the points of the template curve, and $\bar{C}_1, \cdots, \bar{C}_2$ represent the points of the other curve to be aligned. Then we define the distance between the curves to be

$$\sum_{i=1}^{n} \min_{j \in \{1,\cdots,m\}} d(C_i, \bar{C}_i)$$

, where $d(C_i, \bar{C}_i)$ is the euclidean distance between two points.

The refined alignment is found by doing a search on a small space of perturbations of scaling and rotation from the original alignment. This approximates the solution to the problem, but the curves are already in a close alignment from the first step; here we are merely refining this alignment.

The basic algorithm behind our search is as follows:

1. Begin with the original curve.

2. Generate slight perturbations to this curve as successor states (rotations, translations, and scaling here, but could be other affine transformations).

3. Choose the best of the perturbed and original curves in terms of our equation above.

4. Replace the original curve with this best found curve. Repeat the procedure until a minimum has been reached.

5. Reduce the size of the perturbation, and repeat until a desired tolerance has been met.

The affine alignment behaves stable enough. The results of aligning all subjects is shown in figure 18.

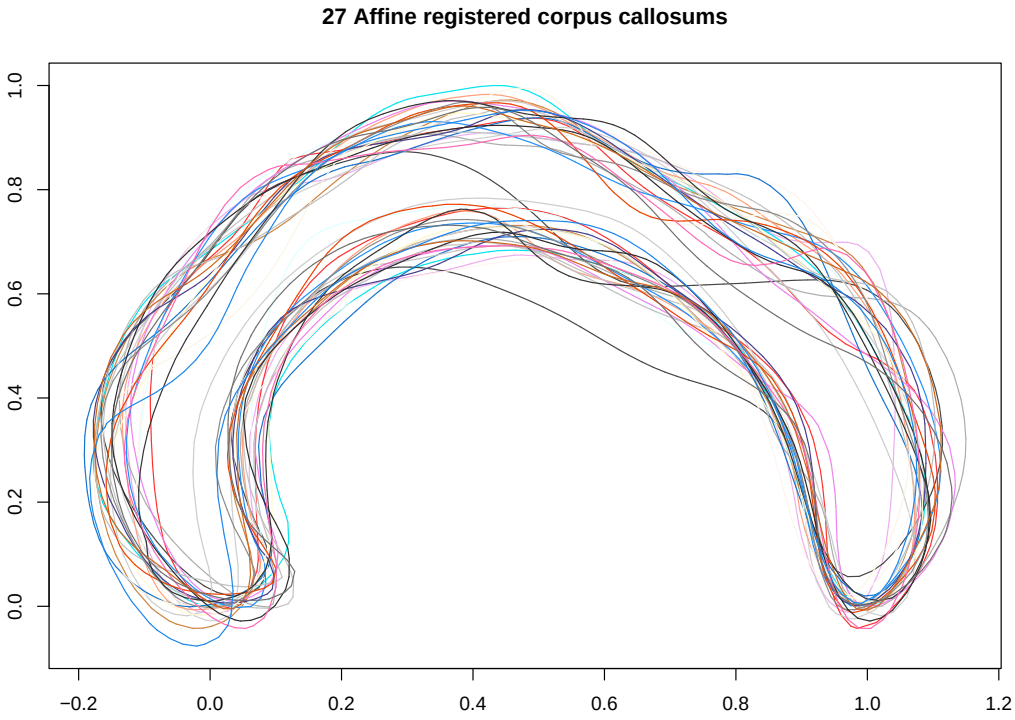

Figure 18: Plot of the affine registered corpus callosum curves (random colors).

5.2 Deformation by dynamic time warping

First consider that we are just working on two curves, rather than two closed curves. Let $C_1, \cdots C_n$ represent the points of the template curve, and $\bar{C}_1, \cdots, \bar{C}_m$ represent the points of the other curve to be aligned. Then let $\kappa_1, \cdots, \kappa_n$ and $\bar{\kappa_1}, \cdots, \bar{\kappa_m}$ be the discrete respective corresponding curvatures. Let $\alpha(C_i, \bar{C}_j)$ be the euclidean distance between the two points, so that it is penalizing the affine distance from one curve to another. Finally let the cost be defined as

$$d(C_i, \bar{C}_j) = \alpha(C_i, \bar{C}_j) + \eta \|\kappa_i - \bar{\kappa_j}\|$$

, where η is a chosen constant controlling the weight. Then we have an additive metric.

If we had a one to one mapping, then C_i would map to $\bar{C}_i$ in our curve, but the curves are of different length. If we introduce one shift map $s(i)$, then we can define a map from $C_{s(i)}$ to $\bar{C}_i$. However this allows one point in C mapping to $\bar{C}$, but not vice versa. So we modify this formulation to incorporate a second shift map $\bar{s}(i)$ [25]. Hence our formulation of the cost for these parameters is just

$$D(C, \bar{C}, s, \bar{s}) = \sum_{i=0}^{I} d\left(C_{s(i)}, C_{\bar{s}(i)}\right)$$

. Then we estimate the shift maps by minimizing D:

$$\zeta(C, \bar{C}) = \min_{s, \bar{s}} D(C, \bar{C}, s, \bar{s})$$

Now, we need only go to $I = \max(m, n)$, because the cost of $s(i+1) = s(i) = \bar{s}(i) = \bar{s}(i+1)$ is greater than $s(i) = \bar{s}(i)$, so minimization will not halt.

This formulation can be minimized by dynamic programming, when formulated recursively [8]. Since we have a positive monotonic metric, we rewrite the

minimization as:

$$\zeta(C_{s[0,i]}, \bar{C}_{\bar{s}[0,i]}) = \min_k \left(\zeta(C_{s[0,k]}, \bar{C}_{\bar{s}[0,k]}) + \zeta(C_{s[k,i]}, \bar{C}_{\bar{s}[k,i]})\right)$$

Now we clarify that we are actually working on a closed curve, rather than just a curve. The above formulation is not sufficient enough, as it does not tell us where the map begins. Sebastian suggests an efficient algorithm to find the optimal map between two curves by fixing a point on one curve, and then choosing the minimum cost of all maps starting at every point from the second curve [26]. Instead, we pick the area of highest curvature in the Rostrum of the corpus callosum [1] as our starting point, with the assumption that these points should be matched between the two curves anyway. Sebastian also suggests using curvature and arc-length which are more intrinsic curve properties, whereas here we use curvature and affine distance to stabilize instead. Now with this formulation, we can apply the dynamic time warping method to deform one curve to the other.

5.3 Constructing the template-registered curve

The results of using the above metric provides a fairly stable algorithm that does not often map one point in one curve to many points in another curve. The affine cost helps stabilize errors in curvature estimation or different curvature estimation metrics used. An example of this map is shown in figure 19. However, the points mapped to each other as a result of this mapping are not in a bijective relationship, so it will not be easily analyzable. Thus we reconstruct the registered curve to give us a bijective mapping. The curve is reconstructed from the affine registered curves as follows. If one point maps to one point, then neither point is altered. If one

point maps to several points, we interpolate those several points to find the middle value. This results in a bijective correspondence between the curves. This curve is to be referred to as the template-registered curve.

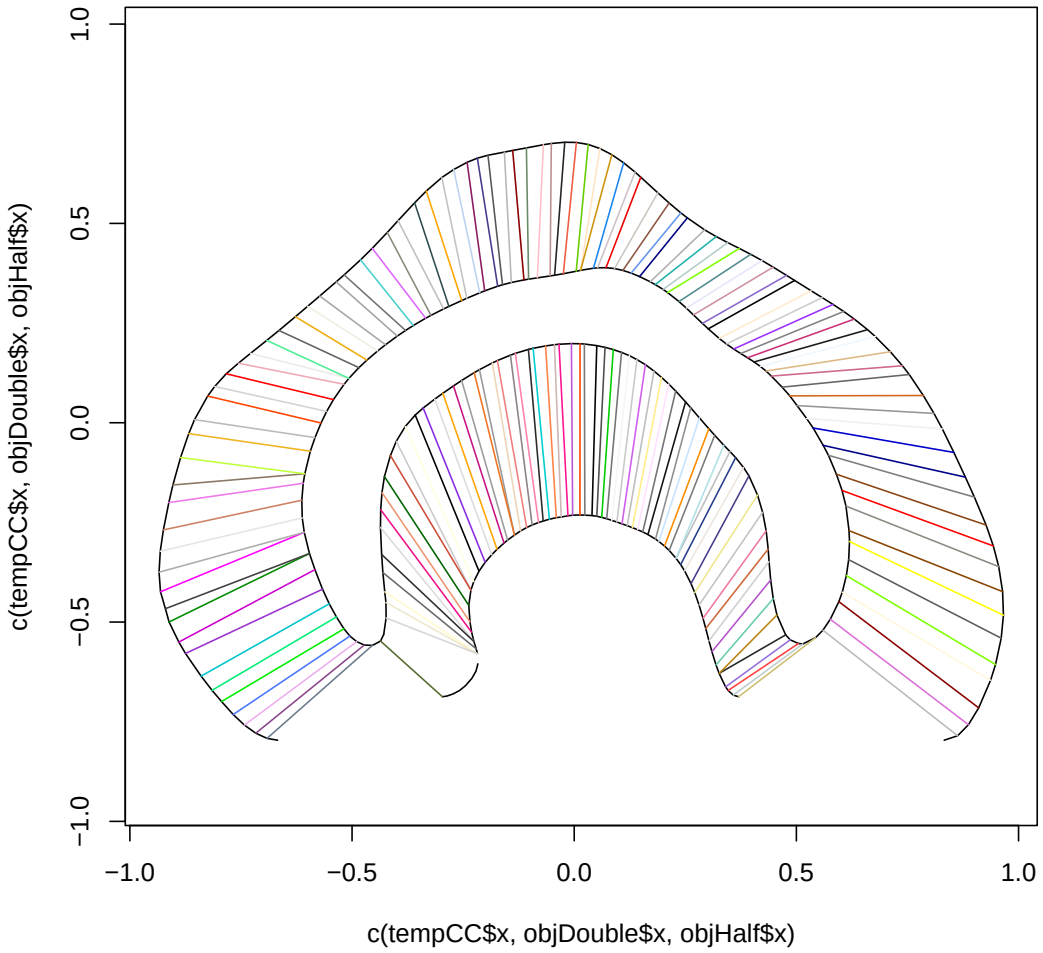

Figure 19: The map of one curve to the other using the dynamic time warping algorithm. For display purposes, the template curve is cut, scaled up on the upper piece and down on the lower piece, and translated appropriately. Points mapped to each other are indicated by randomly colored lines.

Chapter 6

Morphometric analysis

Given a method of constructing a deformation mapping from one corpus callosum to another corpus callosum, we can now construct a deformation mapping amongst all subjects. We choose one control subject that has a larger number of points to serve as a template curve, and then deformed all other subjects to that curve. From that the atlas curve and template-registered curves where formed, as described in the previous section. Thus we map a series of points from one curve to a series of points on the other curve. With such a mapping, we can analyze the differences in curvature and deformation field. This data set was the same as that used in Chung et al. [7]; however, here we restrict our analysis in assuming that there is no effect of age and IQ.

6.1 Curvature Metric Analysis

In order to adequately analyze the curvature amongst the subjects, we look at the results of applying the tests on different smoothing parameters. When smoothing well beyond the GCV criterion, we get much more stable results.

6.2 Piece-wise spline, $\lambda = 150$

Here we work with the template-registered curves. With the curves being registered, at each point on the template curve, we have a corresponding deformed point from all over curves. Thus at each atlas point we have one group of points representing the control group with corresponding estimated curvatures, and one group of points representing the autistic group with corresponding estimated curvatures. We begin with a curve with smoothing parameter $\lambda = 150$ (slightly above GCV and CV) with the piece-wise spline smoothing, spline curvature estimation method. From looking at the variance maps of the two curves in figure 20 and figure 21 we see that though the variance maps look fairly constant in most areas, the lower right Rostrum of the corpus callosum shows more variance that the other areas. This is partially due to the variance in that area, but may also be in part due to the registration algorithm. When comparing the two groups for equal variance, we get the p-values shown in figure 22.

We use Welch's t-test on all points, to make up for any inconsistencies in variance, so we are less likely to produce false positives. The results of this smoothing parameter are shown in figure 23 and figure 24.

6.3 Piece-wise spline, $\lambda = 10000$

When we use a much higher smoothing parameter, we get better results. Here we again use the piece-wise spline smoothing with the spline curvature estimation, but with a heavy smoothing parameter of $\lambda = 10000$. The variance results are much smaller and more stable as shown in figure 25 and figure 26, as to be expected

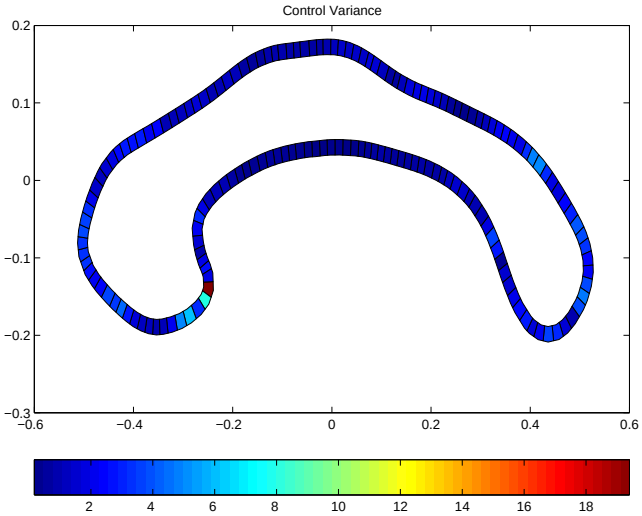

Figure 20: Control variance map ($\lambda = 150$).

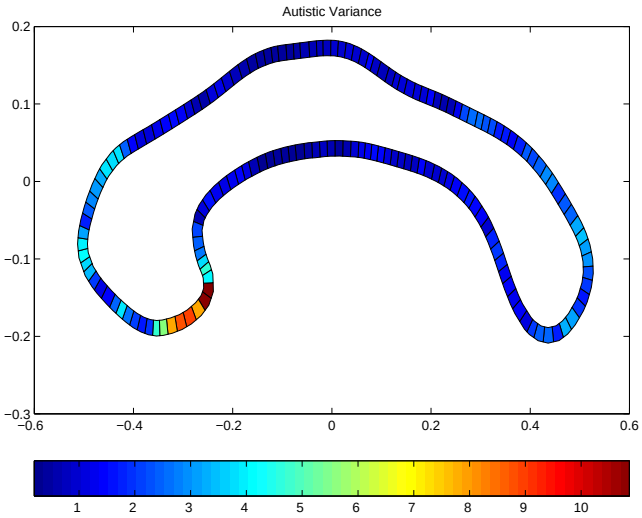

Figure 21: Autistic variance map ($\lambda = 150$).

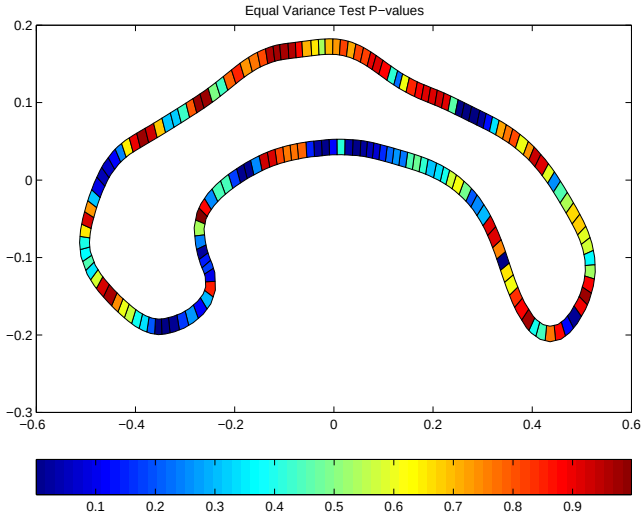

Figure 22: P-value for equal variance test ($\lambda = 150$).

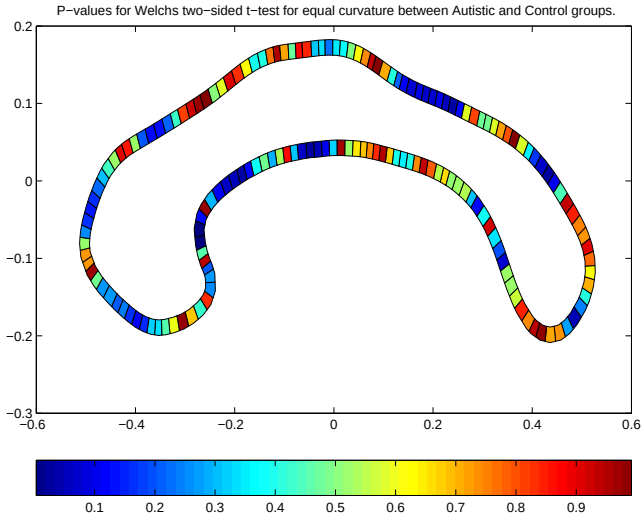

Figure 23: P-value for significant difference in curvature, without multiple comparison correction. ($\lambda = 150$)

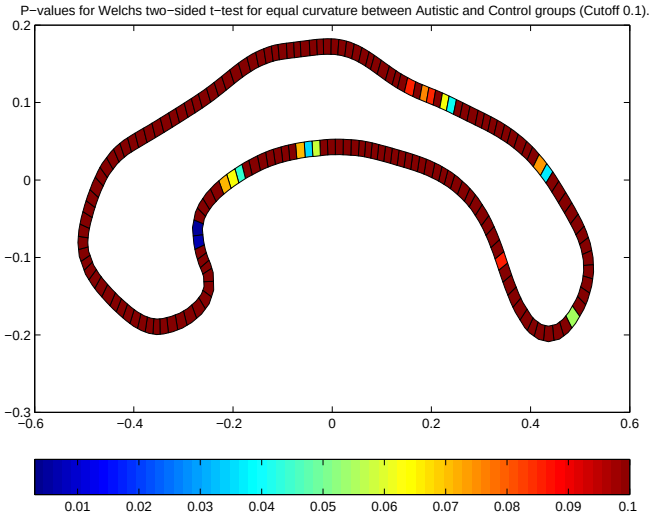

Figure 24: P-value for significant difference in curvature with cutoff at 0.1, but without multiple comparison correction. ($\lambda = 150$)

with a much higher smoothing parameter. Finally, when looking at the results of Welch's t-test in figure 27 and figure 28, we see a much smoother p-value map. We also have the t-value map in figure 29 to show what direction the curvature change is in. There are a few regions of the midbody and the splenium that show very low p-values, indicating

6.4 Deformation field analysis

The deformation field analysis is based on a point-wise basis. To calculate the deformation field for one subject, we take the all of the affine registered curves, which have the same number of points in one to one correspondences, and average them to form an "atlas" curve. Then to calculate the deformation field for each curve, we take the difference of each to the atlas curve. Once this has been constructed, we

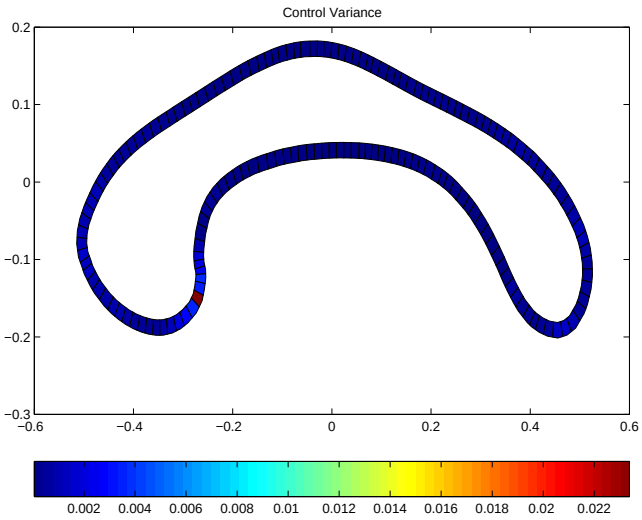

Figure 25: Control variance map ($\lambda = 10000$).

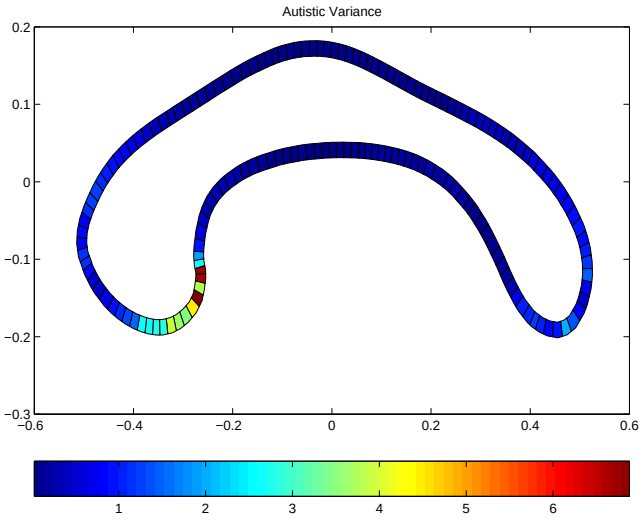

Figure 26: Autistic variance map ($\lambda = 10000$).

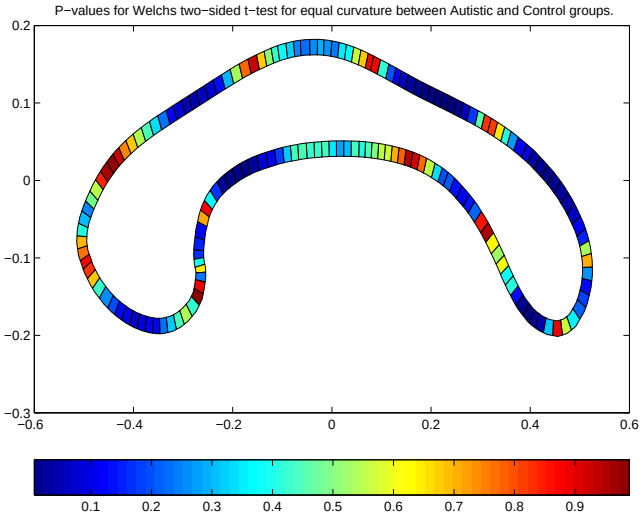

Figure 27: P-value for significant difference in curvature, without multiple comparison correction. ($\lambda = 10000$)

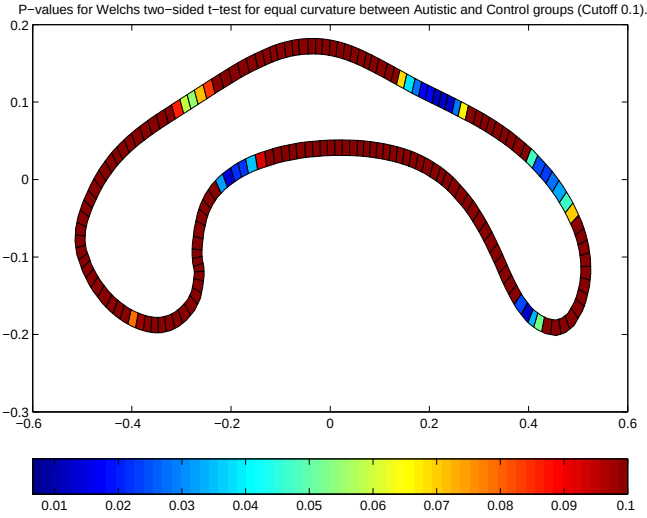

Figure 28: P-value for significant difference in curvature with cutoff at 0.1, but without multiple comparison correction. ($\lambda = 10000$)

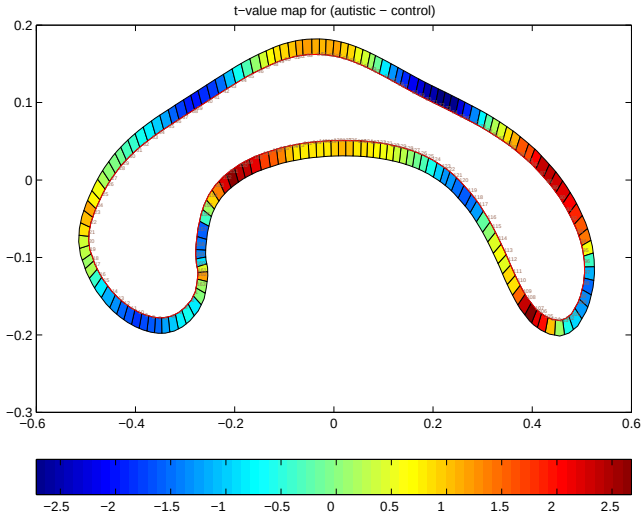

Figure 29: t-map for curvature difference. ($\lambda = 10000$)

can analyze the deformation field on a point-wise basis using Hotelling's t-squared statistic [21]. Again, we will look at the effect on the analysis of choosing two different smoothing parameters.

6.4.1 Piece-wise spline, $\lambda = 150$

Looking at the figures of the p-values along the deformation field in figure 30 and figure 31, we see no highly significant results, especially if we take into consideration any sort of multiple comparison testing.

6.4.2 Piece-wise spline, $\lambda = 10000$

Again, when looking at the figures of the p-values along the deformation field in figure 33 and figure 34, we see no highly significant results. The results are at the most slightly stronger for this smoothing parameter, rather than $\lambda = 150$.

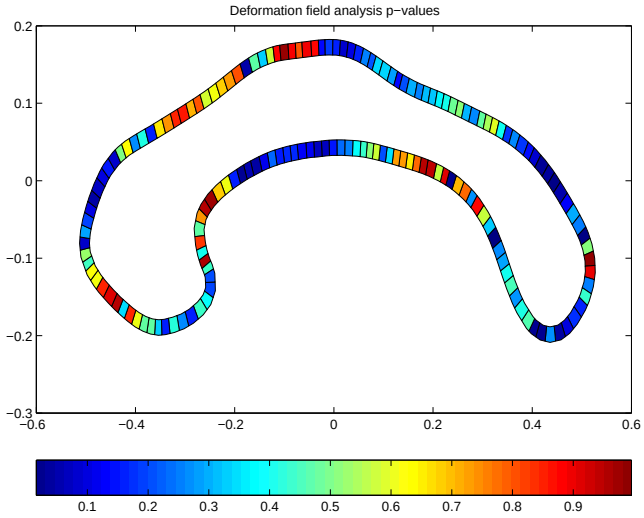

Figure 30: P-values of the deformation field. ($\lambda = 150$)

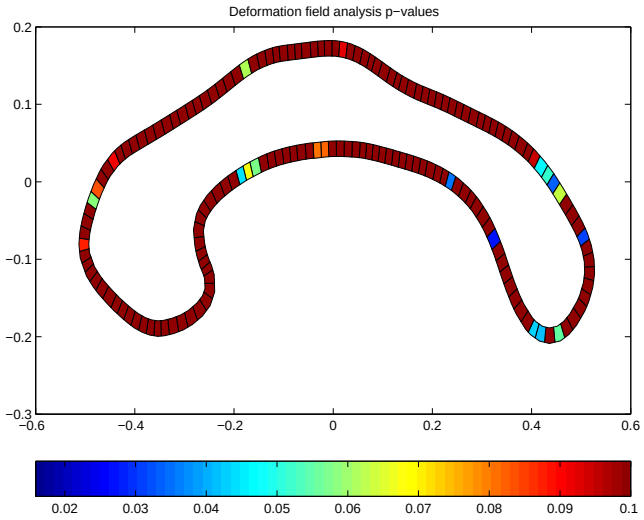

Figure 31: P-values of the deformation field, with a cut-off at 0.1. ($\lambda = 150$)

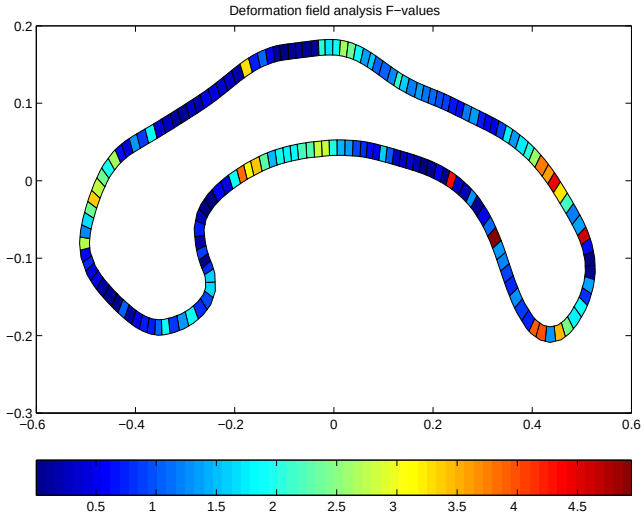

Figure 32: F-values of the deformation field. ($\lambda = 150$)

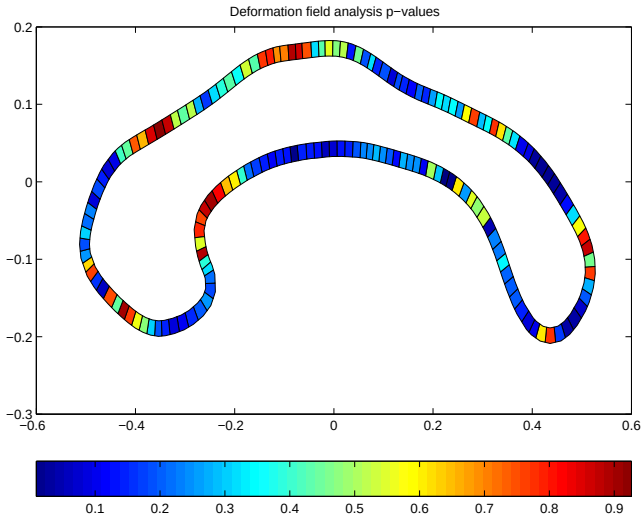

Figure 33: P-values of the deformation field. ($\lambda = 10000$)

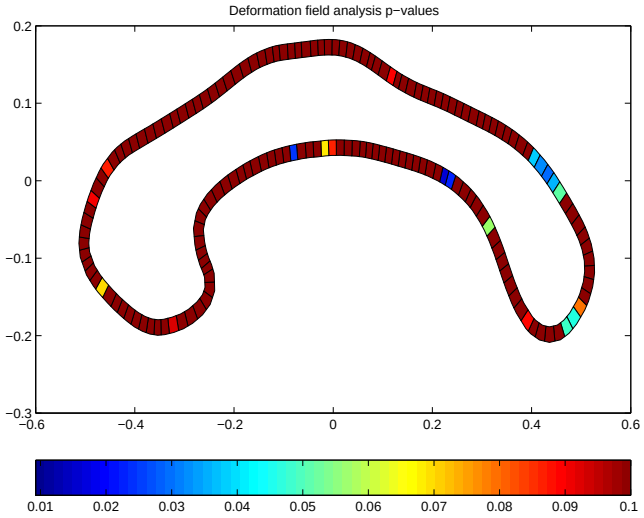

Figure 34: P-values of the deformation field, with a cut-off at 0.1. ($\lambda = 10000$)

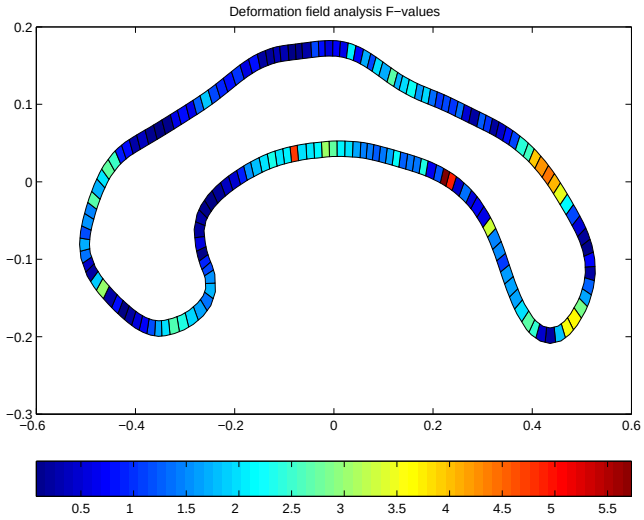

Figure 35: F-values of the deformation field. ($\lambda = 10000$)

6.5 Conclusion

We only see moderately significant results in the highly smoothed curvature region. Smoothing the curve well beyond the GCV or CV is critical in getting a good estimate of curvature in this pixelated case. However, we are not able to see much of any significance in the deformation field despite finding something in the highly smoothed region. This suggests that the deformation field is not as demanding on how smooth the curve is as the curvature is. This also shows how slight variabilities in methods used to check for such differences might get different results, depending on what was used as prior estimators.

Bibliography

[1] *Medical gross anatomy atlas images.* Retreived May 1, 2004 from http://www.med.umich.edu/lrc/coursepages/M1/anatomy/html/atlas/n1a5p8.html.

[2] M. Bakircioglu, U. Grenander, N. Khaneja, and M. Miller, *Curve matching on brain surfaces using Frenet distances.*, Hum Brain Mapp, 6 (1998), pp. 329–33.

[3] R. E. Bellman and S. E. Dreyfus, *Applied dynamic programming*, Princeton U Press, NJ, 1962.

[4] P. Brambilla, A. Hardan, S. U. di Nemi, J. Perez, J. C. Soares, and F. Barale, *Brain anatomy and development in autism: review of structural MRI studies.*, Brain Res Bull, 61 (2003), pp. 557–69.

[5] M. Brejl and M. Sonka, *Object localization and border detection criteria design in edge-based image segmentation: automated learning from examples.*, IEEE Trans Med Imaging, 19 (2000), pp. 973–85.

[6] J. D. Carew, R. K. Dalal, G. Wahba, and S. B. Fain, *Estimating arterior wall shear stress*, Tech. Report 1088, University of Wisconsin, 1210 W Dayton St., Dec 2003.

[7] M. K. Chung, K. Dalton, A. L. Alexander, and R. J. Davidson, *White matter density of corpus callosum in autism: 2d voxel based morphometry*, tech. report, University of Wisconsin, 2004.

[8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*, McGraw Hill, 2001.

[9] C. Davatzikos, J. L. Prince, and R. N. Bryan, *Image registration based on boundary matching*, Trans. on Medical Imaging, (1996), pp. 212–215.

[10] C. Davatzikos, X. Tao, and D. Shen, *Hierarchical active shape models, using the wavelet transform.*, IEEE Trans Med Imaging, 22 (2003), pp. 414–23.

[11] R. L. Eubank, *Nonparametric regression and spline smoothing*, Marcel Dekker, Inc., NY, 2 ed., 1999.

[12] P. J. Flynn, *On reliable curvature estimation*, tech. report, Michigan State University, 1989.

[13] B. v. Ginneken, A. F. Frangi, J. J. Staal, B. M. t. H. Romeny, and M. A. Viergever, *Active shape model segmentation with optimal features.*, IEEE Trans Med Imaging, 21 (2002), pp. 924–33.

[14] P. J. Green and B. W. Silverman, *Nonparametric regression and generalized linear models: A roughness penalty approach*, Chapman & Hall, NY, 1994.

[15] A. Y. Hardan, N. Minshew, and M. S. Keshavan, *Corpus callosum size in autism*, Neurology, (2000), pp. 1033–1036.

[16] R. A. Irizarry, *Choosing smoothing parameters for smoothing splines by minimizing an estimate of risk.* February 2004.

[17] M. Leventon, O. Faugeraus, and W. Grimson, *Level set based segmentation with intensity and curvature priors*, (2000).

[18] R. Malladi and J. A. Sethian, *Image processing via level set curvature flow*, in National Academy of Sciences of the United States of America, vol. 92, July 1995, pp. 7046–7050.

[19] ——, *Image processing: Flows under min/max curvature and mean curvature*, Graphical Models and Image Processing, 58 (1996), pp. 127–141.

[20] ——, *An $O(N \log N)$ algorithm for shape modeling*, in Appllied Mathematics, no. 18, National Academy of Sciences of the United States of America, September 1996, pp. 9389–9392.

[21] B. F. J. Manly, *Multivariate statistical methods: a primer*, Chapman & Hill, 1994.

[22] S. Osher and R. Fedkiw, *Level set methods*, (2000).

[23] J. Piven, J. Bailey, B. J. Ranson, and S. Arndt, *An mri study of the corpus callosum in autism*, American Journal of Psychiatry, 154 (1997), pp. 1051–1056.

[24] W. Reddick, J. Glass, E. Cook, T. Elkin, and R. Deaton, *Automated segmentation and classification of multispectral magnetic resonance images of brain using artificial neural networks.*, IEEE Trans Med Imaging, 16 (1997), pp. 911–8.

[25] T. B. Sebastian, J. J. Crisco, P. N. Klein, and B. B. Kimia, *Constructing 2d curve atlases*.

[26] T. B. Sebastian, P. N. Klein, and B. B. Kimia, *On aligning curves*, IEEE transactions on pattern analysis and machine intelligence, 25 (2003), pp. 116–24.

[27] J. A. Sethian, *Fast marching methods*, SIAM Review, 41 (1999), pp. 199–235.

[28] ——, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge U, 2 ed., 1999.

[29] G. Taubin, *A signal processing approach to fair surface design*, Computer Graphics, 29 (1995), pp. 351–358.

[30] P. M. Thompson, K. L. Narr, R. E. Blanton, and A. W. Toga, *Mapping structural alterations of the corpus callosum during brain development and degeneration*, in The Corpus Callosum, M. Lacoboni and E. Zaidel, eds., Kluwer Academic Press, 1999.

[31] G. Wahba, *Spline models for observational data*, vol. 59 of CBMS-NSF Regional Conference Series, SIAM, 1990.

[32] G. Wahba and Y. Wang, *When is the optimal regularization parameter insensitive to the choice of the loss function?*, Commun. Statist.-Theory Meth., 19 (1990), pp. 1685–1700.

[33] K. Wang and T. Gasser, *Alignment of curves by dynamic time warping*, Annals of Statistics, 25 (1997), pp. 1251–1276.

[34] E. W. Weisstein, *Curvature*. http://mathworld.wolfram.com/Curvature.html, 1999.

[35] S. F. Witelson, *Hand and sex differences in the isthmus and genu of the corpus callosum: a postmortem morphological study*, Brain, (1989), pp. 799–835.