

RICE UNIVERSITY

**A computational framework for evaluating
outcomes in infant craniosynostosis reconstruction**

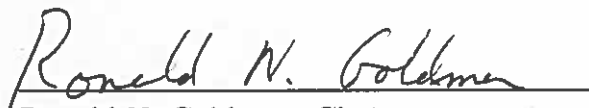
by

Binhang Yuan

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Master of Science

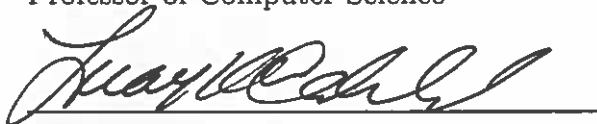
APPROVED, THESIS COMMITTEE:



Ronald N. Goldman, Chair
Professor of Computer Science



Joe D. Warren
Professor of Computer Science



Luay Nakhleh
Associate Professor of Computer Science,
of Ecology and Evolutionary Biology, and
of Biochemistry and Cell Biology

Houston, Texas

January, 2016

ABSTRACT

A computational framework for evaluating outcomes in infant craniosynostosis reconstruction

by

Binhang Yuan

Historically, surgical outcomes in craniosynostosis have been evaluated by qualitative analysis, direct and indirect anthropometry, cephalometrics, and CT craniometric analysis.

Three-dimensional meshes constructed from 3dMD images acquired on patients with synostosis at multiple times across the course of surgical treatment provide ideal raw data for a novel approach to 3D geometric shape analysis of surgical results.

We design a automatic computational framework for evaluating and visualizing the results of infant cranial surgeries based on 3dMD images. The goal of this framework is to assist surgeons in evaluating the efficacy of their surgical techniques. Feedback from surgeons in Texas Children's Hospital confirms that this framework is a robust computational system within which surgical outcomes in synostosis can be accurately and meaningfully evaluated.

We also propose an algorithm to generate normative infant cranial models from the input of 3D meshes, which are extracted from CT scans of normal infant skulls. Comparing of the head shape of an affected subject with a normal control will more clearly illustrate in what aspect the subject's head deviates from the norm. Comparing of a post-treatment subject's head shape and an age-matched control would allow

assessing of a specific treatment approach or surgical technique.

Acknowledgments

I would like to thank Professor Ronald N. Goldman, my advisor, not only for introducing the interesting problem to me but also for patiently giving me careful guidance throughout this research. I would like to thank Doctor David Y. Khechoyan from Texas Children's Hospital, providing me medical data and helpful feedback throughout this research. I also thank Professor Joe D. Warren and Professor Luay Nakhleh for serving as parts of my committee and imparting helpful suggestions.

Contents

Abstract	i
Acknowledgments	iii
List of Illustrations	vi
List of Tables	viii
1 Introduction	1
1.1 Objective Comparison Framework	1
1.2 Normative Model Generation	3
2 Related Work	4
2.1 Objective Evaluation Measurements	4
2.2 Normative Head Shape Modeling	6
3 Comparison Framework	8
3.1 Framework Overview	9
3.2 Mesh Simplification	10
3.3 Registration	12
3.4 Local Features	13
3.4.1 Curvature	13
3.4.2 Normal vectors and the Gauss Map	15
3.4.3 Surface Area	16
3.5 Visualizing local features	16
3.5.1 False Color Function	17
3.5.2 Curvature Visualization	18

3.5.3	Normal Vector Distribution	19
3.5.4	Relative Changed Surface Area	20
3.6	Application of Computational Framework	21
3.6.1	Simplification Results	21
3.6.2	Registration Results	22
3.6.3	Local Features: Comparison and Visualization	22
4	Normative Model Generation	33
4.1	Algorithm Overview	34
4.2	3D Surface Extraction	35
4.2.1	DeVIDE Operating Pipeline	35
4.2.2	Refinement by MeshLab	36
4.3	Point Set Registration	37
4.4	Averaging Algorithm	39
4.5	Pilot Experimental Results	40
4.5.1	Spacial Coherence	41
4.5.2	Model Convergence	41
5	Conclusions and Future Work	45
5.1	Conclusions	45
5.2	Future Work	45
5.2.1	Geometric Features	46
5.2.2	Implicit Normative Model	46
	Bibliography	47

Illustrations

3.1	Outline of the evaluation and visualization framework.	9
3.2	Geometric illustration of two-dimensional curvature.	14
3.3	Mesh simplification, left: input mesh, right: simplified mesh	22
3.4	Registration. The orange meshes on the left represent the source meshes and the blue meshes on the right represent the target meshes.	24
3.5	Curvature visualization. From top to bottom: Gaussian curvature, maximal principal curvature, mean curvature and minimal principle curvature. Blue represents low curvature, while red represents high curvature.	25
3.6	Curvature comparison. From top to bottom: Gaussian curvature, maximal principal curvature, mean curvature, minimal principal curvature. Blue represents low $DC(x_i)$, while red represents high $DC(x_i)$	26
3.7	Weighted vertex normal vector distributions. The first figure is the normalized mean curvature weighted distribution for the head shape of subject 1 at time point 1. The second figure is the normalized mean curvature weighted distribution for the same subject at time point 2.	32
4.1	Framework of the normative model.	34
4.2	Pipeline of the DeVIDE software.	37
4.3	Surface extraction. Left: a surface we extracted after applying DeVIDE. Right: a surface we extracted after applying MeshLab.	38

4.4 Laplacian smoothing. Laplacian smoothing leads to good spacial coherence.	42
---	----

Tables

3.1	Registration time	23
3.2	LAFS, RPFS and AS	28
3.3	Relative changed surface area, threshold $t = 0.25$	29
3.4	Relative changed surface area, threshold $t = 0.50$	30
3.5	Relative changed surface area, threshold $t = 0.75$	31
4.1	Geometric feature differences for the averaging models between each iteration.	44

Chapter 1

Introduction

Cranial vault remodeling operations are performed on infants born with craniosynostosis and head shape dysmorphism. The basic procedure is to reposition the craniofacial skeleton in order to improve both function and aesthetics. Surgical outcomes in craniosynostosis surgery have been limited historically by subjective qualitative evaluation from the treating physician and are confounded by high inter- and intra-observer error and lack of concordance. Objective, quantitative evaluation frameworks that focus on three-dimensional aspects of normal and abnormal morphology (anatomy) are required to assess surgical outcomes accurately. In this chapter, we will briefly introduce two main methods we propose to aid in assessing surgical outcomes.

1.1 Objective Comparison Framework

Multiple medical centers have reported on the reliability of stereophotogrammetry as a useful substitute for 3D CT imaging, with the additional advantage of complete soft tissue imaging and a lack of exposure to ionizing radiation [1, 2, 3, 4]. 3dMD, a widely used medical image system, can provide 3D triangle meshes to represent the shape of a subject's head as input for further analysis [5].

However, at this point, there are no effective objective frameworks that produce clinically meaningful craniometric outcome indices.

Based on this scenario, we propose a framework that can comprehensively provide

regional and global cranial three-dimensional geometric features with corresponding visualizations to aid the treating surgeon in evaluating the severity of the presenting deformity (as it compares to normative cranial forms) and the changes that had occurred with the surgical intervention.

Our technique eliminates the need for selecting surface landmarks and registers mesh pairs automatically. This approach avoids the potential for inter- and intra-observer error. Additionally, the technique relies on stereophotogrammetry, which does not emit the harmful ionizing radiation that is delivered by CT scans. This is particularly beneficial in the pediatric patient population undergoing surgical correction for craniosynostosis as the lifetime risk of developing a malignancy after radiation exposure is higher the earlier in life a CT scan is performed [6]. This radiation free approach will allow for the capture of multiple images over time to provide longitudinal quantitative data on outcomes, information previously unavailable to surgeons. Compared to CT scans, three-dimensional surface imaging is also more capable of capturing the facial contours and the aesthetics of soft tissue. Finally, the 3dMD system has a capture speed of 1.5ms at highest resolution, eliminating the need for sedation which is often necessary to obtain a CT scan of a pediatric patient [7].

Our framework consists of the following four parts: (i) mesh decimation to simplify the input; (ii) registration to find a correspondence for mesh pairs; (iii) comparison of local surface features between mesh pairs to evaluate the quality of the surgery; (iv) visualization to illustrate this comparison. More details of this framework will be covered in Chapter 3.

1.2 Normative Model Generation

In addition to the comparison framework applied between the pre-surgery mesh and the post-surgery mesh to illustrate the head shape change following the operation, surgeons still prefer to check how the pre-surgery mesh and the post-surgery mesh diverge from a normal infant’s cranial shape. For this purpose, we would like to generate a robust series of normal cranial shapes from which age-, gender-, and race-matched controls may be extracted. This comparison will provide more accurate information for evaluating treatments.

Since 3DMD systems have been in use only for a short time, it is difficult to collect enough 3D meshes of normal infants’ head shapes. On the other hand, multiple medical centers have accumulated plenty of medical CT images, which would provides good alternative input for generating normative models.

Our algorithm for generating a normative model consists of three parts: (i) extracting the 3D mesh representing the head shape with complete soft tissue from the CT scans; (ii) applying a registration algorithm to find a correspondence map between pairs of meshes; (iii) averaging the related mesh points to generate a normative model.

Our algorithm makes two main contributions. Our algorithm generates a normative model targeting the infants’ head shape with complete soft tissue starting from CT scan data. The advantage of the model with soft tissue is that the comparison method in Chapter 3 can be extended to a comparison between abnormal and normal head shape, so the extended framework naturally inherits the merits of the original framework. Additionally, our framework scales well. Given a new sample of a normal head shape, the normative model can be updated with low overhead. We will introduce our method for generating a normative model in Chapter 4.

Chapter 2

Related Work

The measure of cranial morphology has evolved from linear anthropometrics, to CT craniometric vector analysis, to three-dimensional vector analysis, and finally to three-dimensional shape analysis on stereophotogrammetric images without manual landmark registration. In this chapter, we will survey related work in two areas: objective evaluation measurement and normative head shape modeling.

2.1 Objective Evaluation Measurements

Initial attempts to evaluate surgical outcomes involved categorizing the need for surgical revision using the Whitaker classification [8]. This and other qualitative methods are limited by inter- and intra-observer variability.

Direct linear anthropometry uses anatomical landmarks to make cranial measurements. However, performing these measurements on infants and children is logistically impractical and time-consuming [9, 1, 2, 10].

CT-based vector analysis provides a more sophisticated means of assessing cranial form, where measurements are taken from a central fixed point within the skull to an anatomical or pre-defined point on the outer surface of the skull [11, 12]. This technique is limited to analysis in two dimensions based only on a combination of linear or angular indices and performs poorly in characterizing overall, three-dimensional cranial form. Three dimensional vector analysis allows for better three-dimensional

quantification of cranial morphology; however, the algorithm requires user-selected manual input for selecting landmarks [1, 13, 14].

With the recent advances of 3D imaging, several researchers have tried to use mathematical metrics to evaluate the surgical outcome in Plastic Surgery. [15] considers curvature for quantitative assessment of head shape. Curvature is an important local feature for shape analysis. The limitation of [15] is that the authors consider only curvature of two-dimensional curves generated by some specific axial plane cut. Three-dimensional surface curvature would give better local information about the shape of the head as we will describe in Chapter 3.

[16] uses normal vectors for head shape analysis based on stereophotogrammetry. The basic idea of considering normal vectors is appropriate, but their further mathematical analysis of normal vectors is somewhat rudimentary. The authors group the surface normal vectors into several bins determined by azimuth angles and elevation angles, and simply generate histograms by the number of surface normal vectors in each bin. In contrast, we consider the magnitude of each vertex normal vector, where the magnitude is computed from the mean curvature at the vertex, as the weight to construct the histogram.

[17] tries to develop a quantitative computer-based method for measuring cleft lip severity. However, this solution is based on some machine learning algorithms, which take the doctors' subjective opinions as the ground truth for the training data set. As a result, the accuracy of their computational assessment will be determined by the doctors' subjective evaluation of cleft lip severity. In comparison, our approach is to provide some objective metrics based on mathematical modeling to help the surgeons evaluate the severity of the disease and the quality of their surgery.

2.2 Normative Head Shape Modeling

Accurately modeling the shape of the normal human skull plays an important role in aiding diagnosis, surgical treatment, and postoperative analysis. Various researchers have constructed such models based on different data sets.

By introducing the concept of virtual healing, Brief et al [18] first propose creating a 3D norm, which allows comparison between the current head shape of a subject and a typical normal shape considering factors such as age and sex. The limitation of this method is that it needs manual landmarking for the averaging algorithm.

Marcus et al [19] describe a method to generate a three-dimensional surface point cloud representing a subject's head shape. For comparison, the study focuses on computing the average of a group of three dimensional vector analysis measurements. Similarly, Marcus et al [14] apply three dimensional vector analysis (3DVA) software to generate a set of point clouds and use the average and standard deviations for the age and gender bins of point clouds to create a normative 3D model. The risk of such approaches is that the vector analysis measurement is subject to error sources common in traditional anthropometric measurement, such as soft-tissue compression, poor patient compliance, and inaccurate landmark identification.

Subramanyan and Dean [20] propose a procedure to average 3D anatomical structures by encoding the entire surface as a series of B-spline space curves, and then applying average landmark configuration, average space curve generation, and averaging internal tile curves to generate the final model. The method is intricate and still requires manual landmarks.

Recent research has illustrated that the use of computer-assisted design and computer-assisted manufacturing (e.g. 3D printing) to create a prefabricated template will help to achieve standardized, objective and precise correction of craniosyn-

ostosis [21, 22, 23, 24]. Burge et al [23] generate a fronto-orbital bandeau template from the CT scans of multiple children with normal skulls and uses this template to optimize surgical outcomes. Khechoyen et al [22] reveal that using such bandeau templates leads to better conformity between the reconstructed supra-orbital bar and the ideal normal bandeau shape and reduces the duration of operations. These templates focus on modeling the skull shape to aid surgeries. On the other hand, we generate models representing head shape with complete soft tissue to help to objectively evaluate the quality of the surgery by comparing the normative model with 3dMD meshes of the subjects' head.

McComb [25] introduces a framework similar to our approach. However, there are two main differences: (i) he aims at modeling the skull while we are targeting the head shape with complete soft tissue covering the skull; (ii) the registration algorithms and averaging methods are fundamentally different.

Chapter 3

Comparison Framework

In this chapter, we will introduce our mathematically sophisticated automatic objective computational framework to evaluate and visualize the results of infant cranial surgery. The framework consists of the following four parts: (i) mesh decimation to simplify the input; (ii) registration to find a correspondence for mesh pairs; (iii) comparison of local surface features between mesh pairs to evaluate the quality of the surgery; (iv) visualization to illustrate this comparison.

The 3D triangle mesh captured by the 3dMD camera consists of a large number of vertices and faces, which includes a lot of redundancy and makes it very time-consuming for further manipulation and analysis. To solve these problems, we use a mesh simplification algorithm to preprocess the data.

To compare local surface features, we need to find a correspondence between mesh pairs.

Differential geometry provides several local features that can be used to describe the shape of the head. These features include normal vectors, surface areas, maximal and minimal principal curvatures, and mean and Gaussian curvatures.

Additionally, we also show how to visualize these features and describe some experimental results generated by our framework.

3.1 Framework Overview

The outline of our evaluation and visualization system for infant cranial surgeries is summarized in Figure 3.1.

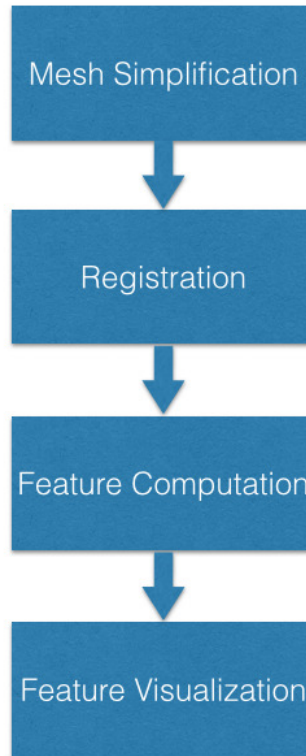


Figure 3.1 : Outline of the evaluation and visualization framework.

The input for this system is the 3D triangle mesh captured by the 3dMD medical image system. Each 3D mesh contains about 10^5 vertices and about 2×10^6 faces. This large number of vertices and faces would make registration and computation extremely time-consuming. So we apply a mesh simplification algorithm as part of our framework. The mesh simplification algorithm removes redundancy in the input. We will discuss the mesh simplification algorithm in Section 3.2.

After simplifying the mesh, we need to register mesh pairs. Our evaluation system is capable of comparing local information for three types of the mesh pairs: (i) the pre-surgery 3D mesh and the post-surgery 3D mesh of the same subject, (ii) the pre-surgery 3D mesh and the normative head model, (iii) the post-surgery 3D mesh and the normative head model. For any of these comparisons, we need a correspondence map, so that for any vertex in the source mesh, we know the corresponding vertex in the target mesh. For this purpose, we need a registration process, and this procedure will be covered in Section 3.3.

Using the correspondence map, we can compare local features for 3D mesh pairs. The local features include normal vectors, surface areas, and three-dimensional curvatures. We will elaborate these features in Section 3.4.

We will discuss how to combine these local features to evaluate and visualize the surgery in Section 3.5. This section includes methods for visualizing curvature, computing normal vector distribution and calculating relative changed surface area.

Finally, we will describe some experimental results generated by our framework in Section 3.6. Surgeons in Texas Children’s Hospital suggests that the outcome of our objective framework can clearly quantify and display changes in cranial form to inform surgical decisions and postoperative care.

3.2 Mesh Simplification

[26] describes a way of decimating a mesh while maintaining the overall appearance of the object. They call their method *quadric error metric decimation* because the decimating cost is approximated in a quadratic term $\Delta(v) = v^T Q v$, where v are the homogeneous coordinates for any vertex and Q is a symmetric matrix characterizing the geometric error. We use this method in our framework to simplify input.

The fundamental operation of quadric based mesh decimation is *edge collapse*, which merges two vertices into one, removes the edges connecting to these two vertices and adds new edges connecting to the new vertex. An edge collapse can only occur on a *valid pair* (v_1, v_2) , which means either (v_1, v_2) is an edge or $\|v_1 - v_2\|$ is less than a threshold parameter. The simplification algorithm can be summarized by the following steps:

1. Compute the Q matrices characterizing the geometric error for all the initial vertices.
2. For all valid pairs (v_1, v_2) , compute the optimal contraction target \bar{v} for (v_1, v_2) . Take the error $\bar{v}^T (Q_1 + Q_2) \bar{v}$ of this target vertex as the cost of contracting that pair.
3. Place all the pairs in a heap keyed on cost with the minimum cost pair at the top.
4. Iteratively remove the pair (v_1, v_2) of least cost from the heap, contract this pair, and update the costs of all valid pairs involving v_1 and v_2 , until the number of vertices is less than a user-defined maximum number of vertices.

This algorithm is quite efficient. Given a mesh with roughly 10^5 vertices, decimating 90% of the vertices takes only about 2 seconds in our system environment, which will be described in Section 4.1. So this algorithm is a good choice for our computational framework.

3.3 Registration

In our computational framework, we use the coherent point drift method described in [27] to implement registration.

The input to this method is two point sets. The registration assigns correspondences between two sets of points and recovers the transformation that maps one point set to the other.

In [27], the alignment of two point sets is considered as a probability density estimation problem, where the first point set (*source*) represents the Gaussian mixture model cluster centroids and the second point set (*target*) represents the data points. The correspondence probability between two point sets is defined as the posterior probability of the Gaussian mixture model cluster centroid given the data points. The method uses the EM (Expectation-Maximization) algorithm to estimate the parameters (including the representation of the transformation) of this Gaussian mixture model.

The computational complexity of this method can be reduced to linear with the help of a fast Gauss transform. The coherent point drift registration code used for this part of our framework is the implementation in MATLAB by Andriy Myronenko ([27]).

The output of this registration process includes a transformation on the target and the correspondence map between the source and the target. The transformation relocates the target to the same position and resizes the target to the same scale as the source to prepare the way for the correspondence map. Given the source point set $X_{N \times 3} = \left(x_1, \dots, x_N \right)^T$, and the target point set $Y_{M \times 3} = \left(y_1, \dots, y_M \right)^T$,

the transformation can be represented as:

$$T(y_j; R, t, s) = sRy_j + t \quad (3.1)$$

where R is a 3×3 rotation matrix, t is a 3×1 translation vector, s is a scaling parameter and y_j is any point of the target point set.

The correspondence map can be represented as a function:

$$y_j = f(x_i) \quad (3.2)$$

where x_i is any point in the source, and y_j is the corresponding point in the target. With this function, for any vertex in the source, we know the corresponding vertex in the target, and this correspondence is still valid for the mesh underlying this point set.

3.4 Local Features

In this section, we introduce the local surface features in our evaluation system.

3.4.1 Curvature

Intuitively, curvature is the amount by which a geometric object deviates from being flat. The curvature of a smooth curve at each point can be defined as the radius of curvature (i.e. the reciprocal of the radius) of the osculating circle. Figure 3.2 illustrates the geometric meaning of two-dimensional curvature.

Given a plane parametric curve represented as:

$$\begin{cases} x = x(t) \\ y = y(t) \end{cases} \quad (3.3)$$

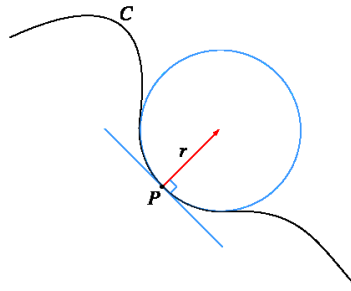


Figure 3.2 : Geometric illustration of two-dimensional curvature.

the formula for the curvature K is given by:

$$K = \frac{\left| \frac{dx}{dt} \cdot \frac{d^2y}{dt^2} - \frac{dy}{dt} \cdot \frac{d^2x}{dt^2} \right|}{\left(\left(\frac{dx}{dt} \right)^2 + \left(\frac{dy}{dt} \right)^2 \right)^{\frac{3}{2}}} \quad (3.4)$$

For surfaces embedded in three dimensions, the definition of curvature is more complicated. Suppose S is a surface in three-dimensional space and P is a point on S . For every unit direction e_θ in the tangent plane of P , we can use the normal vector n_p and the tangent vector e_θ to determine a plane containing P . Cutting S by this plane generates a curve in this plane. Then we can use Formula (4) to compute the curvature for this curve; this curvature is called the *normal curvature*. Among all normal curvatures, the maximal value is called the *maximal principal curvature*, denoted by K_1 , and the minimal value is called the *minimal principal curvature*, denoted by K_2 .

Furthermore, there are two additional important notions of curvature. The *Gaussian curvature* K_G is the product of the two principal curvatures and the *mean curvature* K_M is the average of the two principal curvatures, see Formula (5) and Formula (6).

$$K_G = K_1 \cdot K_2 \quad (3.5)$$

$$K_M = \frac{K_1 + K_2}{2} \quad (3.6)$$

[28] gives an algorithm to compute curvatures for 3D triangle meshes. Many other researchers have improved the accuracy and efficiency for this algorithm. See [29] and [30].

We use the popular computer graphics API, the Point Cloud Library (PCL) [31], to implement our system for curvature computation.

3.4.2 Normal vectors and the Gauss Map

Given a 3D triangle mesh, we can compute face normal vectors for triangle faces and vertex normal vectors for vertices.

Face normal vectors can be computed by the cross product of two nonparallel vectors on that face. Given a triangle with vertices ABC , the unit face normal vector $N_{\Delta ABC}$ can be computed by the following formula:

$$N_{\Delta ABC} = \frac{(A - C) \times (B - C)}{|(A - C) \times (B - C)|} \quad (3.7)$$

There are a great number of different ways to compute unit vertex normal vectors. Since each vertex may belong to many faces, one solution is to take the normalized vector sum of the unit face normal vectors, where the sum is over all faces containing the vertex. Mathematically, this unit vertex normal vector can be represented by the following formula:

$$n_A = \frac{\sum_{\Delta k} N_{\Delta k}}{\left| \sum_{\Delta k} N_{\Delta k} \right|} \quad (3.8)$$

where Δk represents any triangle face that contains vertex A , $N_{\Delta k}$ represents the unit normal vector of face Δk and n_a represents the unit normal vector of vertex A . By this method, we can get the direction of the normal vector for this vertex. Additionally, it is known that the magnitude of a vertex normal vector is the discrete mean curvature of the surface at this vertex.

In differential geometry, the *Gauss map* maps a surface in Euclidean space R^3 to the unit sphere S^2 . Specifically, for each point p of the surface S , the result of the Gauss map is the unit normal vector $N(p) \in S^2$. The Gauss map can be used to analyze the overall geometric character of the surface. [16] uses a discrete representation of the Gauss map by using a discrete representation for the sphere.

3.4.3 Surface Area

Given a 3D triangle mesh, to compute a surface area of a selected region can be simply implemented by summing the area of each triangle in the region.

Additionally, the *area around a vertex* can be defined by the following formula:

$$S_{around}(x_i) = \frac{1}{3} \sum_j S_{\Delta_j} \quad (3.9)$$

where Δ_j is any triangle that contains the vertex x_i , S_{Δ} is the area of triangle Δ_j and $S_{around}(x_i)$ is the surface area around vertex x_i . We choose the constant $\frac{1}{3}$ because if all three vertices in a triangle are selected there will not be any redundancy or missing area in the area sum.

3.5 Visualizing local features

In order to illustrate our local feature comparison and visualization techniques, we first need to introduce our false color function. Then we discuss our methods for

visualizing curvature, computing normal vector distribution and calculating relative changed surface area in this section.

3.5.1 False Color Function

To visualize the local features, especially curvature, on a 3D triangle mesh, one appropriate method is using false color to represent the magnitude of the local features. With sophisticated preprocessing, the local features can be mapped into the range $[0,1]$ by a reasonable method that we will introduce in Section 3.5.2. Then for each value c in the range $[0,1]$, our false color function maps c to a false color represented by an RGB value. The false color is defined by the following function:

$$\left\{ \begin{array}{l} R = 0 \\ G = 2c \quad \text{if } 0 \leq c < 0.5 \\ B = 1 - 2c \end{array} \right. \quad (3.10)$$

$$\left\{ \begin{array}{l} R = 2c - 1 \\ G = 2 - 2c \quad \text{if } 0.5 < c \leq 1 \\ B = 0 \end{array} \right.$$

where R, G, B represent the color dimensions of red, green and blue.

This function will map 0 to blue, 0.5 to green and 1 to red. Values between 0 and 0.5 will be mapped by a linear combination of blue and green, while values between 0.5 and 1 will be mapped by a linear combination of green and red.

With the help of this color function, we are able to visualize each local feature.

3.5.2 Curvature Visualization

Curvature visualization can be divided into two parts: (i) visualizing curvature on the source mesh and the target mesh separately and (ii) visualizing curvature comparisons based on the correspondence map.

To visualize the result of the curvature computation, we first need to normalize the value of curvature to the range $[0,1]$ in order to use our false color function. We use the following formula to accomplish this goal:

$$c(K) = 1 - e^{-\alpha \cdot K} \quad (3.11)$$

where K can be maximal principal curvature, minimal principle curvature, mean curvature or Gaussian curvature, and α is a constant set by experiment. Note α should be set to make the distribution of $c(K)$ well-proportioned in the range $[0,1]$.

The second part is to visualize the difference in the curvature between the source mesh and the target mesh. With the correspondence map described in Section 3.3, we can compare local features between the source mesh and the target mesh. For the curvature, we can compare the difference $DC(x_i)$ between corresponding vertices in the source mesh and in the target mesh by the following formula:

$$DC(x_i) = 1 - e^{-\beta \left| \frac{K(x_i) - K(y_j)}{K(x_i) + K(y_j)} \right|} \quad (3.12)$$

where $K(x_i)$ is the curvature at vertex x_i in the source mesh, and $K(y_j)$ is the curvature at the corresponding vertex y_j in the target mesh; β is a constant set by experiment. The purpose of this function is to map the difference for the vertex pair into the range of $[0,1]$ in order to prepare the way for visualization using false color. Note that the curvature here can be any of the four types of curvature: maximal principal curvature, minimal principal curvature, mean curvature or Gaussian curva-

ture, and β should be set to make the distribution of $DC(x_i)$ well-proportioned in the range $[0,1]$.

3.5.3 Normal Vector Distribution

Similar to [16], we compute the azimuth angle θ and the elevation angle ϕ for each unit vertex normal vector $n(x_n, y_n, z_n)$ by the following formulas:

$$\theta = \begin{cases} \arccos\left(\frac{x_n}{\sqrt{x_n^2+z_n^2}}\right) & \text{if } z_n \geq 0 \\ -\arccos\left(\frac{x_n}{\sqrt{x_n^2+z_n^2}}\right) & \text{if } z_n < 0 \end{cases} \quad (3.13)$$

$$\phi = \arctan\left(\frac{y_n}{\sqrt{x_n^2+z_n^2}}\right)$$

where θ is in the range $[-\pi, \pi]$ and ϕ is in the range $[-\frac{\pi}{2}, \frac{\pi}{2}]$. Then we group the computed angles θ and ϕ of the vertex normal vectors into 12×12 bins (12 parts for elevation and 12 parts for azimuth). Instead of counting each vertex as 1 or 0, we use the normalized mean curvature given by Formula (11) as the weight for each vertex. By summing these weights, we generate a histogram to represent the distribution of the weighted normal vectors. We use normalized mean curvature as the weight for two reasons: (i) mean curvature is typically considered to represent the magnitude of the normal vector; (ii) normalized mean curvature will be in the range $[0,1]$, this will prevent some extremely large mean curvatures from influencing the distribution.

Different from [16], we focus on the anterior head shape instead of posterior head shape. With the revised histogram, we define the left anterior flatness score (LAFS), right anterior flatness score (RAFS) and asymmetry score (AS) in our own way. LAFS is the sum of the histogram bins that correspond to combinations of azimuth angles ranging from 30° to 90° and elevation angles ranging from -15° to -45° , while RPFs is the sum of the histogram bins that correspond to combinations of azimuth angles

ranging from 90° to 150° and elevation angles ranging from -15° to -45° . The AS is defined by the following formula:

$$AS = \frac{LAFS - RAFS}{LAFS + RAFS} \quad (3.14)$$

Note negative AS indicates that the left side is flatter while positive AS indicates that the right side is flatter.

3.5.4 Relative Changed Surface Area

We want to compute the local surface area at locations on the head where there are obvious differences in curvature for the mesh pairs. For this purpose, we need to filter the difference function $DC(x_i)$ (see Formula (12)) by the following formula to select such areas:

$$D^*(x_i) = \begin{cases} 0 & DC(x_i) < t \\ 1 & DC(x_i) \geq t \end{cases} \quad (3.15)$$

where t is a threshold parameter to filter the curvature difference. Using this function, we can select the vertices with a particular level of curvature variation.

Based on the function D^* , we can compute the relative changed surface area at locations on the head where there are relatively large changes in curvature by the following formula:

$$DS = \frac{\sum_{i=1}^M D^*(x_i) \cdot S_{around}(x_i)}{\sum_k S_k} \quad (3.16)$$

where the term $\sum_k S_k$ means the area sum of all the triangles in the source mesh, $D^*(x_i)$ is defined by Formula (15) and $S_{around}(x_i)$ is defined by Formula (9). Divided by the total surface area of the source mesh, the value of the relative change in area DS will also lie in the range of $[0,1]$ and it will be reasonable to compare the difference in this feature between different mesh pairs.

3.6 Application of Computational Framework

The 3D triangle meshes of each pilot subject’s head were extracted from the 3dMD image files. A typical patient with diagnosis of craniosynostosis would routinely have 3dMD scans performed at multiple time points (e.g., pre-operative, and short-term and long-term post-operative time points). As part of the pilot application of the computational framework, anonymized meshes from five subjects with varied head shape deformity were utilized for analysis. Each subject had a complete set of 3dMD images at two time points. Informed consent was acquired from each subject for use of the anonymized 3dMD mesh in this pilot analysis.

In the following subsection, we will introduce the experimental implementation of each step in our evaluation and visualization system.

3.6.1 Simplification Results

The quadric mesh simplification method [26] has been implemented by using the open source software MeshLab [32].

This algorithm is quite time-efficient. We run our dataset in a 64-bit Windows operating system workstation with the Intel Core i5-4590 processor and 8 GB RAM. For an input mesh containing 82782 vertices and 164842 faces, decimating this mesh to a simplified mesh with 8033 vertices and 16000 faces takes only about 2 seconds. Figure 3.3 illustrates this decimation process: the left figure is a mesh with 82782 vertices and 164842 faces, while the right figure is the simplified mesh with 8033 vertices and 16000 faces.

From Figure 3.3, we can see that the mesh simplification maintains the overall shape of the subject’s head. No significant features are lost by decimation.

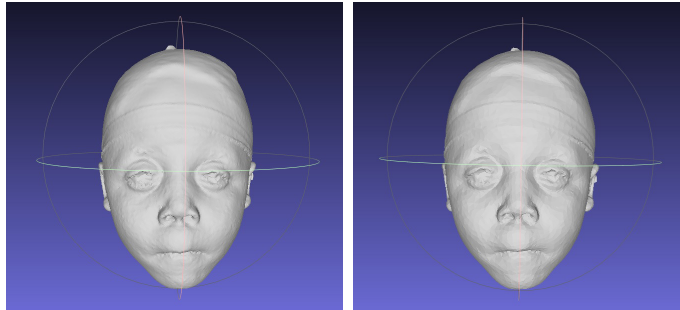


Figure 3.3 : Mesh simplification, left: input mesh, right: simplified mesh

3.6.2 Registration Results

We use the coherent point drift [27] method to register mesh pairs. Specifically, we use the rigid mode of this algorithm and set the maximal number of iterations to 200 and take the tolerance of the error as e^{-8} .

We select 5 mesh pairs in Figure 3.4 to visualize the registration results. The orange meshes on the left represent the source meshes and the blue meshes on the right represent the target meshes. We randomly select some vertices on the source mesh and use the correspondence map to get their related vertices on the target mesh. Then we draw lines to connect these vertex pairs to illustrate the result of registration.

The mesh simplification method reduces the time of registration dramatically. We run the code of [27] in MATLAB R2013a with the same system environment. The running time for the registration is summarized in Table 3.1.

3.6.3 Local Features: Comparison and Visualization

In this part of our experiments, we implement the methods described in Section 3.5 to compare and visualize the local features. The comparison and visualization results were considered helpful to the treating physicians in assessing the outcomes of infant

Subject ID	Non Simplified		
	Source Vertices	Target Vertices	Time
Subject 1	82782	87602	13:18'41"
Subject 2	60661	41150	3:52'57"
Subject 3	40317	57868	2:56'45"
Subject 4	36877	47863	1:44'55"
Subject 5	38365	35984	1:15:46
Subject ID	Simplified		
	Source Vertices	Target Vertices	Time
Subject 1	8274	8745	6'21"
Subject 2	8057	8146	4'22"
Subject 3	8048	7969	4'10"
Subject 4	8221	8070	4'10"
Subject 5	8038	8171	5'50"

Table 3.1 : Registration time

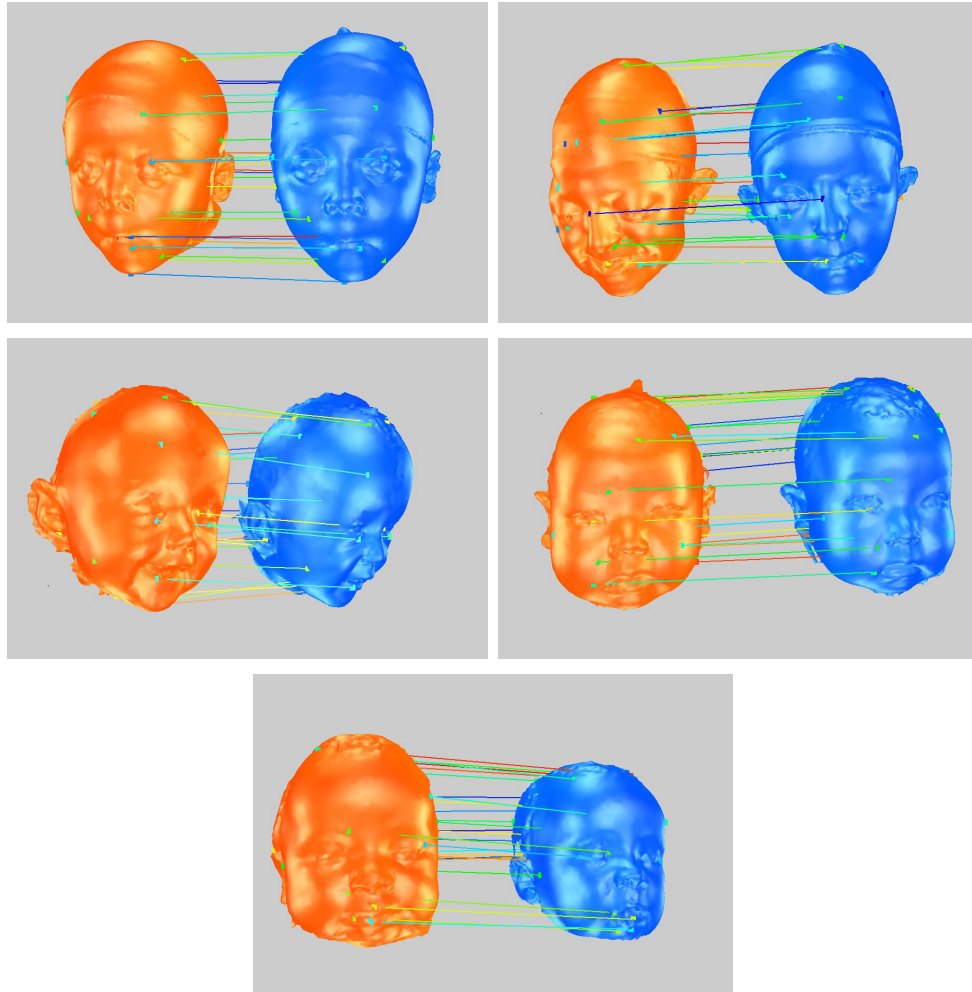


Figure 3.4 : Registration. The orange meshes on the left represent the source meshes and the blue meshes on the right represent the target meshes.

cranial surgery.

Curvature Visualization

Using the method described in Section 3.5.2, we generate Figure 3.5 to visualize curvature and Figure 3.6 to display curvature comparisons. Based on our observation experiments, we find that the mean curvature and the maximal principal curvature give more relevant information to assist in evaluating the shape of the subject's head.

To get the best visual effect when visualizing curvature, for the first three types of curvature, we set α in Formula (11) to 40, while for the Gaussian curvature, we set α in Formula (11) to 2000. From Figure 3.5, we can see that the flat part of the head shows low values of curvature represented in blue, while the curved part such as the ears and the tip of the nose, shows high values of curvature represented in red. These colors are consistent with the geometric meaning of curvature.

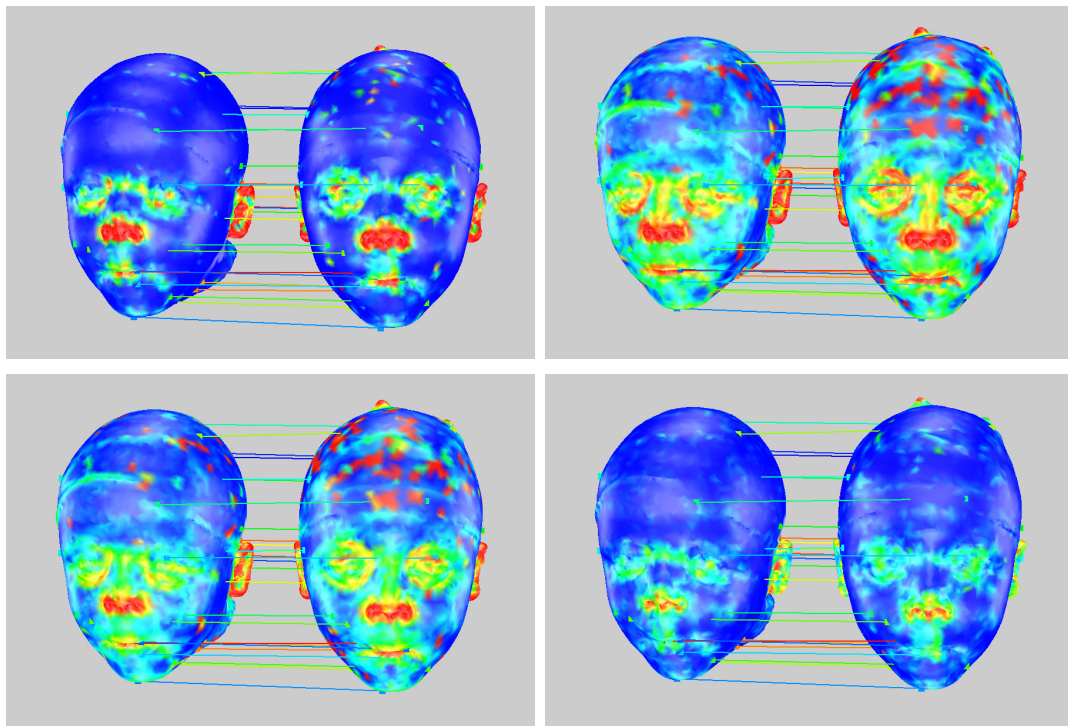


Figure 3.5 : Curvature visualization. From top to bottom: Gaussian curvature, maximal principal curvature, mean curvature and minimal principle curvature. Blue represents low curvature, while red represents high curvature.

When visualizing curvature comparisons, we set β in Formula (12) to 2.5 to get the best visual effect. A visualization result can be viewed in Figure 3.6. Note that this subject has a head shape deformity characterized by a tower-like and narrow (from front to back) appearance (i.e., turribrachycephaly). The two comparison meshes are

shown at distinct treatment time points. The changes in head shape are localized to the anterior 2/3rds of the cranial vault. At the second time point, the forehead has a more normal curved appearance which correlates to the three-dimension false color in Figure 3.6.

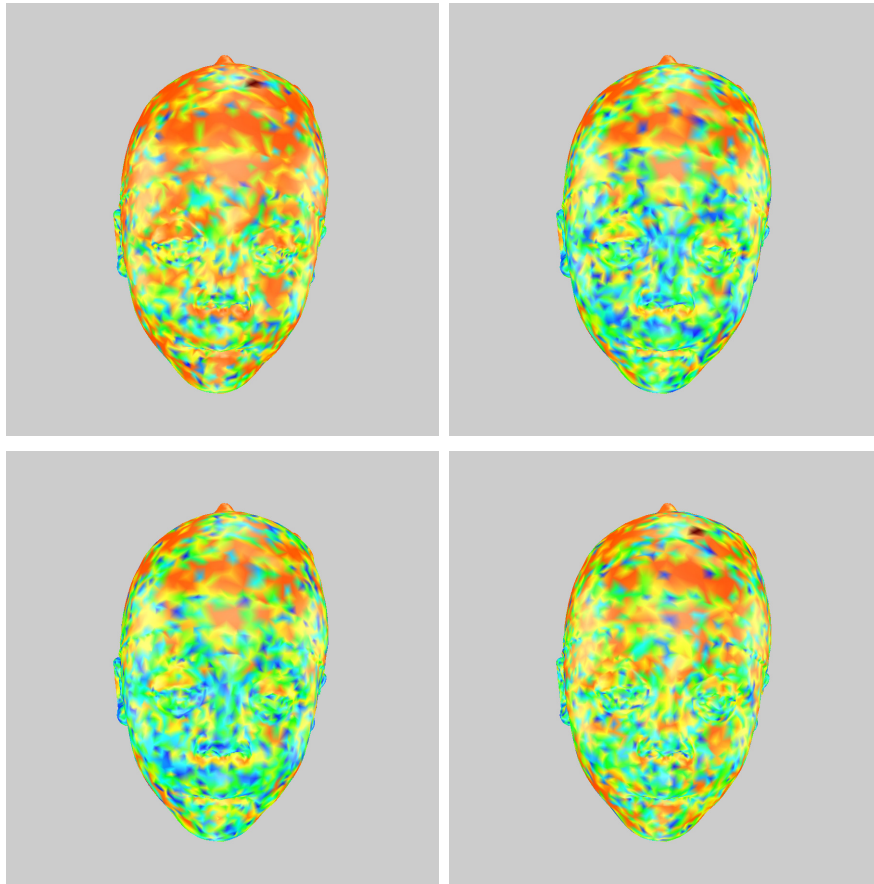


Figure 3.6 : Curvature comparison. From top to bottom: Gaussian curvature, maximal principal curvature, mean curvature, minimal principal curvature. Blue represents low $DC(x_i)$, while red represents high $DC(x_i)$.

Normal Vector Distribution

According to the method in Section 3.5.3, we compute the vertex normal vector distributions and calculate LAFS, RAFS and AS among our data set.

Figure 3.7 illustrates a pair of the vertex normal vector distributions. The figures are the normalized mean curvature weighted distributions at two treatment time points of test subject 1.

We also compute LAFS, RAFS, and AS for two distinct treatment time points within our pilot test subject data set.

The results are summarized in Table 3.2.

Note the absolute value of AS quantified the degree of asymmetry. High AS implies a more asymmetric head shape. The results in Table 3.2 are consistent with specific head shape deformity of each subject. This measurement allows for more accurate quantification of severity of head shape deformity, which may, in turn, improve diagnosis and clinical care.

Relative Changed Surface Area

Table 3.3, Table 3.4 and Table 3.5 show the relative changed surface area filtered by four different kinds of curvature. The computational method is given by Formula (15) and Formula (16) in Section 3.5.3. The value of β in Formula (12) is set to the same value as in the previous Curvature Visualization section and t is set to 0.25, 0.5 and 0.75 respectively.

This relative changed surface area information is very helpful to the clinician in evaluating the regional changes that occur with surgery. It also allows for more meaningful subject counseling.

Subject ID	Pre-surgery			Post-surgery		
	LAFS	RAFS	AS	LAFS	RAFS	AS
Subject 1	49.0923	52.4496	-0.03306	57.3033	54.2026	0.02781
Subject 2	43.0490	38.6312	0.05409	67.6544	60.4020	0.05663
Subject 3	31.7818	39.6151	-0.10971	60.9226	54.5493	0.05519
Subject 4	33.1238	26.0269	0.11998	51.4286	42.5658	0.09429
Subject 5	29.9552	22.2420	0.14777	35.9186	40.1260	0.05530
Subject 6	48.2420	53.3174	0.04997	56.9955	51.9252	0.04655
Subject 7	48.3503	44.1282	0.04565	53.2970	46.2095	0.07123
Subject 8	42.4052	55.9537	0.13775	52.3954	57.9485	0.05033
Subject 9	66.0616	65.3683	0.00528	64.6218	67.9659	0.02522
Subject 10	54.2664	47.7836	0.06356	51.0006	39.3414	0.12906
Subject 11	41.0100	57.7453	0.16946	53.3046	45.7331	0.07645
Subject 12	20.1010	23.5369	0.07874	34.4817	30.6050	0.05956
Subject 13	29.3217	18.4092	0.22863	29.2531	40.0900	0.15628
Subject 14	22.4992	29.6454	0.13705	28.8485	39.9900	0.16185
Subject 15	17.2269	15.9391	0.03883	38.0420	44.3377	0.07642
Subject 16	19.4698	18.2094	0.03345	43.1527	50.7385	0.08079
Subject 17	37.2002	44.1074	0.08495	48.3973	44.2544	0.04471
Subject 18	24.2604	27.2204	0.05750	28.9321	24.8459	0.07598
Subject 19	21.6221	41.2580	0.31228	28.4197	23.9606	0.08513

Table 3.2 : LAFS, RPFs and AS

Threshold $t = 0.25$				
Subject ID	Max	Min	Mean	Gaussian
Subject 1	0.5952	0.6596	0.5910	0.7307
Subject 2	0.5548	0.6130	0.5505	0.7067
Subject 3	0.6134	0.6641	0.6095	0.7521
Subject 4	0.5448	0.5997	0.5377	0.7007
Subject 5	0.6022	0.6330	0.5948	0.7319
Subject 6	0.6917	0.6981	0.6883	0.7586
Subject 7	0.6084	0.6571	0.6036	0.7394
Subject 8	0.6641	0.7037	0.6617	0.7859
Subject 9	0.6665	0.6733	0.6597	0.7343
Subject 10	0.7448	0.7528	0.7431	0.8327
Subject 11	0.7024	0.7145	0.6979	0.7701
Subject 12	0.6833	0.6564	0.6786	0.7109
Subject 13	0.5696	0.5874	0.5650	0.6470
Subject 14	0.6645	0.7029	0.6599	0.7519
Subject 15	0.7439	0.7761	0.7390	0.8165
Subject 16	0.6908	0.7279	0.6864	0.7850
Subject 17	0.7305	0.7663	0.7264	0.8114
Subject 18	0.6316	0.6803	0.6234	0.7466
Subject 19	0.6502	0.6957	0.6429	0.7601

Table 3.3 : Relative changed surface area, threshold $t = 0.25$

Threshold $t = 0.5$				
Subject ID	Max	Min	Mean	Gaussian
Subject 1	0.5264	0.6063	0.5204	0.6077
Subject 2	0.4746	0.5486	0.4638	0.6624
Subject 3	0.5507	0.6123	0.5461	0.7205
Subject 4	0.4661	0.5299	0.4564	0.6577
Subject 5	0.5350	0.5757	0.5248	0.6976
Subject 6	0.6484	0.6653	0.6426	0.7400
Subject 7	0.5422	0.6075	0.5339	0.7036
Subject 8	0.6109	0.6629	0.6101	0.7582
Subject 9	0.6159	0.6354	0.6084	0.7120
Subject 10	0.7152	0.7217	0.7141	0.8171
Subject 11	0.6600	0.6835	0.6541	0.7513
Subject 12	0.6452	0.6286	0.6405	0.6945
Subject 13	0.5206	0.5540	0.5152	0.6264
Subject 14	0.6142	0.6701	0.6068	0.7280
Subject 15	0.7115	0.7529	0.7063	0.7996
Subject 16	0.6463	0.6960	0.6411	0.7624
Subject 17	0.6965	0.7382	0.6905	0.7929
Subject 18	0.5722	0.6384	0.5579	0.7181
Subject 19	0.5950	0.6517	0.5861	0.7342

Table 3.4 : Relative changed surface area, threshold $t = 0.50$

Threshold $t = 0.75$				
Subject ID	Max	Min	Mean	Gaussian
Subject 1	0.3745	0.4693	0.3674	0.5923
Subject 2	0.2885	0.3852	0.2770	0.5378
Subject 3	0.3744	0.4670	0.3740	0.6194
Subject 4	0.2921	0.3531	0.2830	0.5242
Subject 5	0.3518	0.4032	0.3425	0.5867
Subject 6	0.5339	0.5739	0.5339	0.6826
Subject 7	0.3949	0.4635	0.3846	0.6039
Subject 8	0.4731	0.5293	0.4628	0.6769
Subject 9	0.4981	0.5290	0.4873	0.6438
Subject 10	0.6043	0.6162	0.6056	0.7660
Subject 11	0.5555	0.5873	0.5448	0.6900
Subject 12	0.5364	0.5460	0.5301	0.6417
Subject 13	0.3898	0.4535	0.3796	0.5594
Subject 14	0.4772	0.5616	0.4713	0.6595
Subject 15	0.6101	0.6807	0.5993	0.7491
Subject 16	0.5257	0.5918	0.5188	0.6958
Subject 17	0.5961	0.6599	0.5837	0.7412
Subject 18	0.4286	0.5125	0.4217	0.6258
Subject 19	0.4437	0.5273	0.4366	0.6482

Table 3.5 : Relative changed surface area, threshold $t = 0.75$

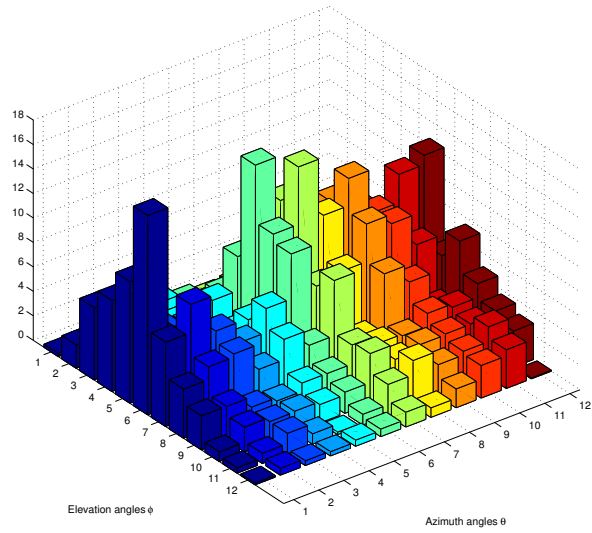
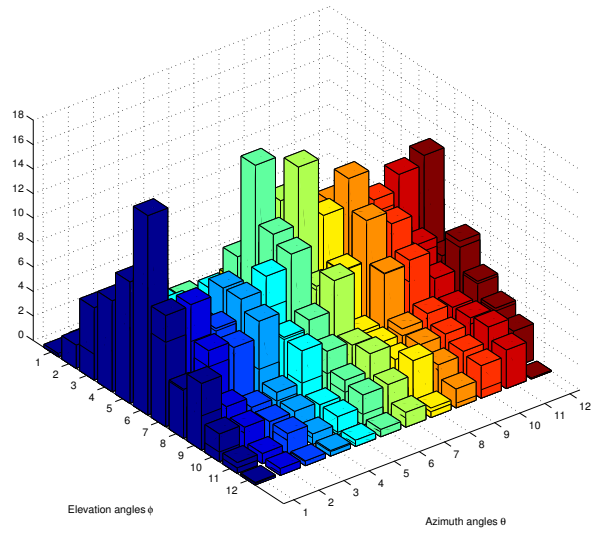


Figure 3.7 : Weighted vertex normal vector distributions. The first figure is the normalized mean curvature weighted distribution for the head shape of subject 1 at time point 1. The second figure is the normalized mean curvature weighted distribution for the same subject at time point 2.

Chapter 4

Normative Model Generation

In this chapter, we will describe our method for generating a normative model. Our system contains the following three parts: (i) mesh extraction to abstract the mesh representing the head shape with the complete soft tissue from CT scans; (ii) mesh registration to find correspondences between mesh pairs; (iii) an averaging algorithm to compute the normative head shape.

The advantage of stereophotography has been discussed in Chapter 1 and Chapter 2. However, because 3dMD image system was applied for a short period, it is difficult to collect enough 3D meshes of normal infants' head shapes. On the other hand, historically Texas Children's Hospital has accumulated plenty of CT skull images, which provide a good alternative data. A CT scan allows doctors to see inside the organ without surgery. Since we want to extract the surfaces of subjects' head with the complete soft tissue, we will apply some tools for this purpose.

The automatic registration algorithm provides correspondence between mesh pairs, which eliminates the requirement for manual landmarking and is necessary for the averaging computation.

The averaging algorithm is the core procedure for generating normative models. Our averaging computation is a scalable stream algorithm. Whenever new data is acquired, the data can be used to update the normative model with low overhead.

Finally, we show some pilot experimental results generated by our normative model.

4.1 Algorithm Overview

The framework of our normative model for normal infant cranial shape is summarized in Figure 4.1.

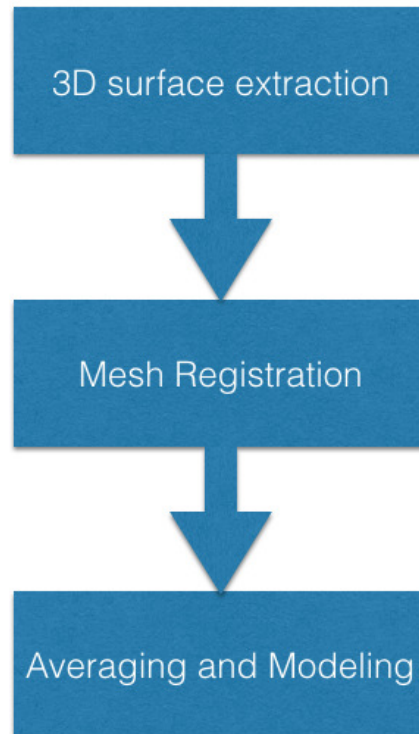


Figure 4.1 : Framework of the normative model.

The first module in our framework extracts from CT scans the surface of the subjects' head shapes with the complete soft tissue covering the skull. The purpose of this step is to make it possible to compare the normative model we generate and the 3D images from stereophotography of the subject's head for diagnosis and objective evaluation. To accomplish this goal, we use the tools of DeVIDE [33] and MeshLab [32]. The details of this part will be covered in Section 4.2.

The registration module plays an important role in this system. Registration provides the correspondence between pairs of 3D meshes. We apply the same registration discussed in Chapter 3. We will give a brief review of this method in Section 4.3.

The averaging computation module is the core of our system. Our algorithm takes advantage of scalability and good spacial coherence. The detailed method will be introduced in Section 4.4.

Finally, the pilot experimental results of the normative model will be discussed in Section 4.5.

4.2 3D Surface Extraction

In this section, we will introduce how we extract the three dimensional surface of the subject's head shape. To accomplish this goal, we first use DeVIDE [33] to extract a rough surface mesh with noisy fragments and then use MeshLab [32] to refine the mesh for further analysis.

4.2.1 DeVIDE Operating Pipeline

DeVIDE [33], short for Delft Visualization and Image processing Development Environment, is a Python-based data flow application builder that enables rapid prototyping of medical visualization and image processing applications via visual programming. By visually connecting functional blocks, the platform allows users to create a pipeline for different purpose. Figure 4.2 illustrates our pipeline. The function of each component is summarized below:

- DICOMReader module loads the DICOM series from the disk.
- DoubleThreshold module filters the CT scan series by the density threshold. To

maintain the complete soft tissue of the head, we set the lower threshold to 1.0 and upper threshold to 2500.0.

- Closing module performs a greyscale morphological closing on the input image. To clean the input data, the closing module first runs a dilation algorithm by replacing a pixel with the maximal greyscale value over an ellipsoidal neighborhood, then an erosion algorithm by replacing a pixel with the minimal greyscale value over an ellipsoidal neighborhood.
- Contour module extracts iso-surfaces from volume data, and is the crucial part in this pipeline.
- WsMeshSmooth module smoothes the input mesh.
- VtkQuadricDecimation module reduces the number of triangles in the mesh.
- Slice3dVWR module displays the output of the pipeline. The left part of Figure 4.3 shows the extracted result of a subject’s head shape.
- StIWRT module writes the mesh in STL format back to the disk.

4.2.2 Refinement by MeshLab

After we extract the mesh in STL format from the DICOM data by the DeVIDE pipeline, there are still two issues: (i) some noisy fragments are in the mesh; (ii) the shell of the head is constructed by a few close connected layers, whereas we want a single layer surface for further analysis.

To solve the first issue, we use the tool of “Select Connected Components in a region” in MeshLab to select the main part of the mesh and eliminate the noisy fragments.

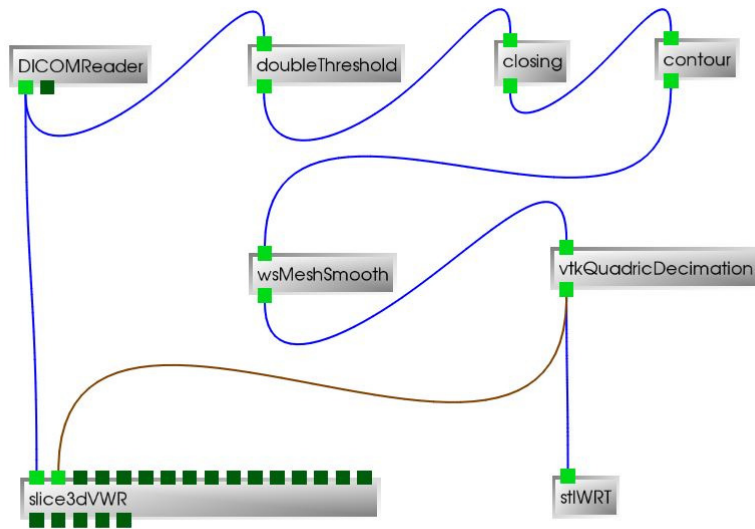


Figure 4.2 : Pipeline of the DeVIDE software.

To overcome the second challenge, we compute the ambient occlusion value for each triangular face of the mesh. The ambient occlusion value computation generates a number of well distributed view directions and computes how many times this face is visible from these directions, The result is mapped to a grayscale. We select the outside layer of our extracted mesh by filtering the triangular faces with low grayscale.

The right part of Figure 4.3 illustrates the refined result of a subject’s head shape.

4.3 Point Set Registration

For this module, we still use the coherent point drift method described in [27] for registration. We will briefly review the input and output of this algorithm in this section.

The input to this method is two point sets. The registration assigns correspon-

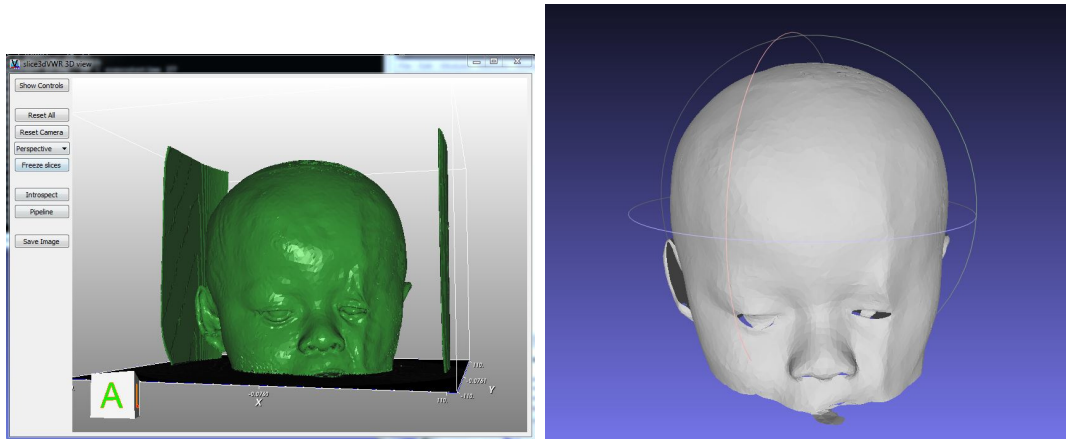


Figure 4.3 : Surface extraction. Left: a surface we extracted after applying DeVIDE. Right: a surface we extracted after applying MeshLab.

dences between two sets of points and recovers the transformation that maps one point set to the other. The output of this registration process includes a transformation on the target and the correspondence map between the source and the target. The transformation relocates the target to the same position and resizes the target to the same scale as the source to prepare the way for the correspondence map.

Formally, given the source point set $X_{N \times 3} = \left(x_1, \dots, x_N \right)^T$, and the target point set $Y_{M \times 3} = \left(y_1, \dots, y_M \right)^T$, the transformation can be represented as:

$$T(y_j; R, t, s) = sRy_j + t \quad (4.1)$$

where R is a 3×3 rotation matrix, t is a 3×1 translation vector, s is a scaling parameter and y_j is any point of the target point set.

The correspondence map can be represented as a function:

$$y_j = f(x_i) \quad (4.2)$$

where x_i is any point in the source, and y_j is the corresponding point in the target.

4.4 Averaging Algorithm

In this section, we will introduce our averaging algorithm. We design the averaging computation as a stream algorithm, which has good scalability.

We first select one mesh from the input mesh set as the temporary average result, and then merge one mesh at a time from the input set by a two-step computation:

i. Update the current vertices. After applying the registration algorithm between the temporary average R (as source) and the incoming input mesh M_i (as target), each vertex on R has a corresponding vertex on M_i and a transformation is performed on the coordinates of all vertices in M_i . For each vertex on R , the current coordinates of this vertex, represented by $(\bar{x}_{i-1}, \bar{y}_{i-1}, \bar{z}_{i-1})$, will be updated to $(\bar{x}_i, \bar{y}_i, \bar{z}_i)$ by the following formula:

$$\begin{cases} \bar{x}_i = (1 - \theta) \bar{x}_{i-1} + \theta x_i \\ \bar{y}_i = (1 - \theta) \bar{y}_{i-1} + \theta y_i \\ \bar{z}_i = (1 - \theta) \bar{z}_{i-1} + \theta z_i \end{cases} \quad (4.3)$$

where (x_i, y_i, z_i) are the coordinates of the vertex in M_i that is mapped to $(\bar{x}_{i-1}, \bar{y}_{i-1}, \bar{z}_{i-1})$, and θ is the weight determined by:

$$\theta = \frac{1}{i} \cdot \frac{1}{e^{\alpha d}} \quad (4.4)$$

where i counts the number of meshes that have been averaged, α is a constant determined by experiments, $d = \sqrt{(\bar{x}_{i-1} - x_i)^2 + (\bar{y}_{i-1} - y_i)^2 + (\bar{z}_{i-1} - z_i)^2}$ is the Euclidean distance between the previous averaged vertex and the new incoming vertex. By Formula 4.4, the weight of the new incoming vertex decreases with the number of meshes in the average model and with the Euclidean distance to the current coordinates. The general ideas behind this approach are that: (i) each mesh should carry the same weight in our stream computation; (ii) the Euclidean distance to the

current coordinates can be used to estimate the confidence of the registration result, we penalize a vertex with large Euclidean distance by decreasing the weight of this vertex.

ii. Laplacian smoothing. Laplacian smoothing [34, 35] is a widely used algorithm to smooth a polygonal mesh. For each vertex in a mesh, a new coordinate is updated based on local information (in our approach, the coordinates of its nearest neighbors). Initially after updating the current vertices, the resulting mesh appears bumpy. So we use the Laplacian smoothing algorithm to improve the spacial coherence of the results. Formally, the smoothing operation can be described as:

$$\begin{cases} x_i = \frac{1}{n} \sum_{j=1}^n x_j \\ y_i = \frac{1}{n} \sum_{j=1}^n y_j \\ z_i = \frac{1}{n} \sum_{j=1}^n z_j \end{cases} \quad (4.5)$$

where n is the number of adjacent vertices to node i , (x_j, y_j, z_j) is the position of the j th adjacent vertex and (x_i, y_i, z_i) is the new position for node i .

In short, the overall process of the averaging computation is summarized in Algorithm 1.

4.5 Pilot Experimental Results

In this section, we will describe our pilot experimental results. We collected 8 series of CT scans from Texas Children’s Hospital. We set the α in Formula 4.4 to 0.1, because this parameter value leads to good experimental results. We applied the above pipeline to generate a normative model. Our pilot experiments includes two parts. The first part demonstrates the good spacial coherence based on Laplacian smoothing. The second part discusses the convergence of our model.

Algorithm 1 Average Mesh Set

Select one mesh M_0 as base from the input mesh set S

Delete M_0 from S

$R \leftarrow M_0$

$i \leftarrow 1$

for all $M_i \in S$ **do**

Register M_i (as source) to R (as target) by [27]

Merge M_i into R by Formula 4.3

Apply Laplacian smoothing on R

$i \leftarrow i + 1$

end for

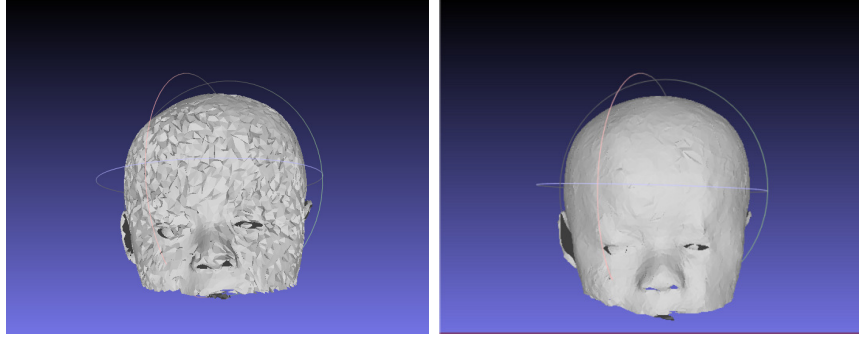
return R

4.5.1 Spacial Coherence

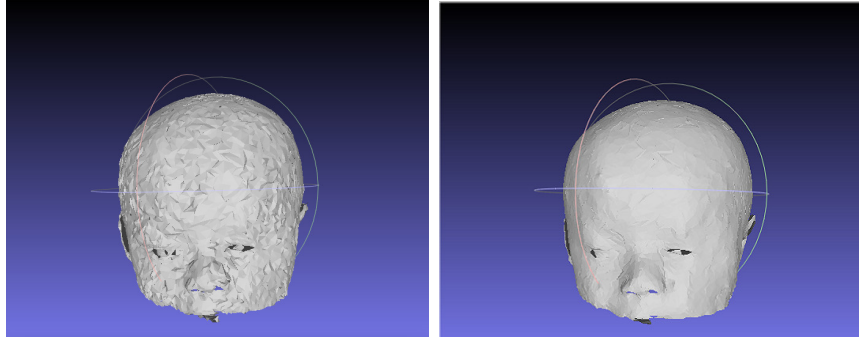
The Laplacian smoothing method [6] has been implemented by using the open source software MeshLab[32]. This algorithm is quite time-efficient. We perform our experiments in a 64-bit Windows operating system workstation with the Intel Core i5-4590 processor and 8 GB RAM. For an input triangle mesh containing 10830 vertices and 20000 faces, Laplacian smoothing takes less than 1 second. Figure 4.4 illustrates the effect of Laplacian smoothing. From Figure 4.4, we observe that Laplacian smoothing leads to good spacial coherence in our model.

4.5.2 Model Convergence

To investigate the convergence of our model, we compare the difference of some geometric features for the averaging models between each iteration. The results are



(a) Average Result of 2 subjects without Laplacian smoothing. (b) Average Result of 2 subjects with Laplacian smoothing.



(c) Average Result of 3 subjects without Laplacian smoothing. (d) Average Result of 3 subjects with Laplacian smoothing.

Figure 4.4 : Laplacian smoothing. Laplacian smoothing leads to good spacial coherence.

listed in Table 4.1. The computation of each measurement is introduced below:

Coordinate difference is the average of the Euclidean distance between each related vertex pair:

$$D_{coordinate} = \frac{1}{N} \sum_{k=1}^N \sqrt{\left(x_k^{(i)} - x_k^{(i-1)}\right)^2 + \left(y_k^{(i)} - y_k^{(i-1)}\right)^2 + \left(z_k^{(i)} - z_k^{(i-1)}\right)^2} \quad (4.6)$$

where $\left(x_k^{(i)}, y_k^{(i)}, z_k^{(i)}\right)$ are the coordinates of a vertex in the average model (iteration i), $\left(x_k^{(i-1)}, y_k^{(i-1)}, z_k^{(i-1)}\right)$ are the coordinates of a vertex in the average model (iteration i-1), and N is the number of vertices in the model.

Normal vector difference is the average of the difference in radian between each related unit normal vector pair:

$$D_{normal} = \frac{1}{N} \sum_{k=1}^N \arccos \left(\frac{v_k^{(i)} \cdot v_k^{(i-1)}}{|v_k^{(i)}| |v_k^{(i-1)}|} \right) \quad (4.7)$$

where $v_k^{(i)}$ is the normal vector of a vertex in the average model (iteration i), $v_k^{(i-1)}$ is the normal vector of a vertex in the average model (iteration i-1), and N is the number of vertices in the model.

Mean curvature and Gaussian curvature differences are the average of the difference of mean curvature and Gaussian curvature between each vertex pair:

$$D_{mean} = \frac{1}{N} \sum_{k=1}^N \left| K_{M_k}^{(i)} - K_{M_k}^{(i-1)} \right| \quad (4.8)$$

$$D_{Gaussian} = \frac{1}{N} \sum_{k=1}^N \left| K_{G_k}^{(i)} - K_{G_k}^{(i-1)} \right| \quad (4.9)$$

where $K_{M_k}^{(i)}$ ($K_{G_k}^{(i)}$) is the mean (Gaussian) curvature of a vertex in average model (iteration i), $K_{M_k}^{(i-1)}$ ($K_{G_k}^{(i-1)}$) is the mean (Gaussian) curvature of a vertex in average model (iteration i-1), and N is the number of vertices in the model.

From Table 4.1, it appears that our averaging algorithm converges slowly as the number of iterations increases. We plan to perform further experiments and analysis when more CT series are provided by Texas Children's Hospital.

Iteration	Changed Features			
	Coordinate	Normal Vector	Mean Curvature	Gaussian Curvature
1	0.67550	0.15387	2.7404×10^{-3}	1.5183×10^{-4}
2	0.37089	0.14542	1.9535×10^{-3}	9.6765×10^{-5}
3	0.27989	0.12629	1.4463×10^{-3}	7.4839×10^{-5}
4	0.19795	0.13806	1.0457×10^{-3}	5.1666×10^{-5}
5	0.17924	0.13366	9.1896×10^{-4}	5.0025×10^{-5}
6	0.17322	0.12825	9.7602×10^{-4}	6.3305×10^{-5}
7	0.12775	0.15102	7.0388×10^{-4}	4.9135×10^{-5}

Table 4.1 : Geometric feature differences for the averaging models between each iteration.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

This thesis describes the effort we make to help surgeons objectively assess and guide their infant cranial surgical operations. Our work mainly focuses on two parts:

We first describe an objective automatic visualization framework, using local surface features from differential geometry to represent particular characteristics of the shape of a subject's head before and after surgery. We have verified that these visualization methods help the physician to assess the quality of their techniques for infant cranial surgery.

We also propose an algorithm to generate normative infant cranial models from CT scans of normal infant skulls. The algorithm has good scalability and produces good results including the information of all input subjects' head shapes with good spacial coherence.

5.2 Future Work

Based on these two components of our work, we recommend exploring two projects in the future:

5.2.1 Geometric Features

In our comparison framework, we employ 3D curvatures, normal vectors and surface areas as the local geometric features. However, there are still plenty of other geometric features, such as curvature flow, lines of curvature, and geodesics worth exploring. It would be beneficial if we can test the potential effectiveness of these geometric features.

5.2.2 Implicit Normative Model

In this thesis, we generate an explicit 3D surface to represent the normative infant head shape by methods adapted from geometric modeling and computer graphics. It would also be interesting to try a statistic machine learning approach for this task. Instead of constructing an explicit 3D surface based on some geometric features, we could try to build a classification model to discriminate if a subject's head is normal or abnormal and a regression model to reveal the severity of the difference between a subject's head shape and the normal head shape. However, it will be necessary to accumulate more data to develop such a machine learning approach for building reliable statistical models.

Bibliography

- [1] J. Y. Wong, A. K. Oh, E. Ohta, A. T. Hunt, G. F. Rogers, J. B. Mulliken, and C. K. Deutsch, “Validity and reliability of craniofacial anthropometric measurement of 3d digital photogrammetric images,” *The Cleft Palate-Craniofacial Journal*, vol. 45, no. 3, pp. 232–239, 2008.
- [2] H.-T. Lübbers, L. Medinger, A. Kruse, K. W. Grätz, and F. Matthews, “Precision and accuracy of the 3dmd photogrammetric system in craniomaxillofacial application,” *Journal of Craniofacial Surgery*, vol. 21, no. 3, pp. 763–767, 2010.
- [3] C. L. Heike, M. L. Cunningham, A. V. Hing, E. Stuhaug, and J. R. Starr, “Picture perfect? reliability of craniofacial anthropometry using three-dimensional digital stereophotogrammetry,” *Plastic and reconstructive surgery*, vol. 124, no. 4, pp. 1261–1272, 2009.
- [4] D. R. McKay, K. M. Davidge, S. K. Williams, L. A. Ellis, D. K. Chong, R. P. Teixeira, A. L. Greensmith, and A. D. Holmes, “Measuring cranial vault volume with three-dimensional photography: a method of measurement comparable to the gold standard,” *Journal of Craniofacial Surgery*, vol. 21, no. 5, pp. 1419–1422, 2010.
- [5] “3dmd.” <http://www.3dmd.com>.
- [6] B. Yuan, D. Y Khechoyan, and R. Goldman, “A new objective automatic computational framework for evaluating and visualizing the results of infant cranial

- surgery,” 2015.
- [7] C.-H. J. Tzou, N. M. Artner, I. Pona, A. Hold, E. Placheta, W. G. Kropatsch, and M. Frey, “Comparison of three-dimensional surface-imaging systems,” *Journal of Plastic, Reconstructive & Aesthetic Surgery*, vol. 67, no. 4, pp. 489–497, 2014.
- [8] L. A. Whitaker, S. P. Bartlett, L. Schut, and D. Bruce, “Craniosynostosis: an analysis of the timing, treatment, and complications in 164 consecutive patients.,” *Plastic and reconstructive surgery*, vol. 80, no. 2, pp. 195–206, 1987.
- [9] B. L. Hutchison, L. A. Hutchison, J. M. Thompson, and E. A. Mitchell, “Quantification of plagiocephaly and brachycephaly in infants using a digital photographic technique,” 2009.
- [10] M. B. Moghaddam, T. M. Brown, A. Clausen, T. DaSilva, E. Ho, and C. R. Forrest, “Outcome analysis after helmet therapy using 3d photogrammetry in patients with deformational plagiocephaly: the role of root mean square,” *Journal of Plastic, Reconstructive & Aesthetic Surgery*, vol. 67, no. 2, pp. 159–165, 2014.
- [11] D. Khechoyan, C. Schook, C. B. Birgfeld, R. K. Khosla, B. Saltzman, C. C. Teng, R. Ettinger, J. S. Gruss, R. Ellenbogen, and R. A. Hopper, “Changes in frontal morphology after single-stage open posterior–middle vault expansion for sagittal craniosynostosis,” *Plastic and reconstructive surgery*, vol. 129, no. 2, pp. 504–516, 2012.
- [12] J. R. Marcus, T. H. Stokes, S. Mukundan, and C. R. Forrest, “Quantitative and qualitative assessment of morphology in sagittal synostosis: mid-sagittal vector analysis,” *Journal of Craniofacial Surgery*, vol. 17, no. 4, pp. 680–686, 2006.

- [13] H. Schaaf, C. Y. Malik, P. Streckbein, J. Pons-Kuehnemann, H.-P. Howaldt, and J.-F. Wilbrand, “Three-dimensional photographic analysis of outcome after helmet treatment of a nonsynostotic cranial deformity,” *Journal of Craniofacial Surgery*, vol. 21, no. 6, pp. 1677–1682, 2010.
- [14] J. R. Marcus, L. F. Domeshek, A. M. Loyd, J. M. Schoenleber, R. R. Das, R. W. Nightingale, and S. Mukundan Jr, “Use of a three-dimensional, normative database of pediatric craniofacial morphology for modern anthropometric analysis,” *Plastic and reconstructive surgery*, vol. 124, no. 6, pp. 2076–2084, 2009.
- [15] C.-C. Teng, A. Hart, R. Hopper, D. Khechoyan, and J. Brown, “Quantitative assessment of supraorbital osseous bar stability and symmetry after frontal orbital advancement for unilateral coronal craniosynostosis,” in *Bioinformatics and Biomedical Engineering (iCBBE), 2010 4th International Conference on*, pp. 1–4, IEEE, 2010.
- [16] I. Atmosukarto, L. Shapiro, J. Starr, C. Heike, B. Collett, M. Cunningham, and M. Speltz, “Three-dimensional head shape quantification for infants with and without deformational plagiocephaly,” *The Cleft Palate-Craniofacial Journal*, vol. 47, no. 4, pp. 368–377, 2010.
- [17] J. Wu, R. Tse, and L. G. Shapiro, “Learning to rank the severity of unrepaired cleft lip nasal deformity on 3d mesh data,” in *Pattern Recognition (ICPR), 2014 22nd International Conference on*.
- [18] J. Brief, S. Hassfeld, S. Däuber, A. Pernozzoli, J. Münchenberg, T. Redlich, M. Walz, R. Krempien, H. Weisser, C. Poeckler, *et al.*, “3d norm data: the first step towards semiautomatic virtual craniofacial surgery,” *Computer aided*

- surgery*, vol. 5, no. 5, pp. 353–358, 2000.
- [19] J. R. Marcus, L. F. Domeshek, R. Das, S. Marshall, R. Nightingale, T. H. Stokes, and S. Mukundan Jr, “Objective three-dimensional analysis of cranial morphology,” *Eplasty*, vol. 8, 2008.
- [20] K. Subramanyan and D. Dean, “A procedure to average 3d anatomical structures,” *Medical image analysis*, vol. 4, no. 4, pp. 317–334, 2000.
- [21] S. Mardini, S. Alsubaie, C. Cayci, H. Chim, and N. Wetjen, “Three-dimensional preoperative virtual planning and template use for surgical correction of craniosynostosis,” *Journal of Plastic, Reconstructive & Aesthetic Surgery*, vol. 67, no. 3, pp. 336–343, 2014.
- [22] D. Y. Khechoyan, N. R. Saber, J. Burge, A. Fattah, J. Drake, C. R. Forrest, and J. H. Phillips, “Surgical outcomes in craniosynostosis reconstruction: The use of prefabricated templates in cranial vault remodelling,” *Journal of Plastic, Reconstructive & Aesthetic Surgery*, vol. 67, no. 1, pp. 9–16, 2014.
- [23] J. Burge, N. R. Saber, T. Looi, B. French, Z. Usmani, N. Anooshiravani, P. Kim, C. Forrest, and J. Phillips, “Application of cad/cam prefabricated age-matched templates in cranio-orbital remodeling in craniosynostosis,” *Journal of Craniofacial Surgery*, vol. 22, no. 5, pp. 1810–1813, 2011.
- [24] J. Soleman, F. Thieringer, J. Beinemann, C. Kunz, and R. Guzman, “Computer-assisted virtual planning and surgical template fabrication for frontoorbital advancement,” *Neurosurgical focus*, vol. 38, no. 5, p. E5, 2015.
- [25] R. W. McComb, “An exploratory approach for mapping the surface of the human skull in three dimensions: Technical methods and clinical application,” 2013.

- [26] M. Garland and P. S. Heckbert, “Surface simplification using quadric error metrics,” in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pp. 209–216, ACM Press/Addison-Wesley Publishing Co., 1997.
- [27] A. Myronenko and X. Song, “Point set registration: Coherent point drift,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 12, pp. 2262–2275, 2010.
- [28] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr, “Discrete differential-geometry operators for triangulated 2-manifolds,” in *Visualization and mathematics III*, pp. 35–57, Springer, 2003.
- [29] G. Guennebaud, M. Germann, and M. Gross, “Dynamic sampling and rendering of algebraic point set surfaces,” in *Computer Graphics Forum*, vol. 27, pp. 653–662, The Eurographics Association and Blackwell Publishing Ltd., 2008.
- [30] G. Guennebaud and M. Gross, “Algebraic point set surfaces,” in *ACM Transactions on Graphics (TOG)*, vol. 26, p. 23, ACM, 2007.
- [31] “Pcl.” <http://pointclouds.org>.
- [32] “Meshlab.” <http://meshlab.sourceforge.net>.
- [33] “Devide.” <https://code.google.com/p/devide/>.
- [34] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel, “Laplacian surface editing,” in *Proceedings of the 2004 Eurographics/ACM SIG-GRAPH symposium on Geometry processing*, pp. 175–184, ACM, 2004.

- [35] G. A. Hansen, R. W. Douglass, and A. Zardecki, *Mesh enhancement: selected elliptic methods, foundations and applications*. Imperial College Press, 2005.