# A New Objective Automatic Computational Framework for Evaluating and Visualizing the Results of Infant Cranial Surgery

B. Yuan[1], D. Khechoyan[2], R. Goldman[3]
Binhang Yuan, Department of Computer Science, Rice University
David Y Khechoyan, Texas Children's Hospital, Department of Surgery, Baylor College of Medicine
Ron Goldman, Department of Computer Science, Rice University
Binhang.Yuan@rice.edu, david.khechoyan@gmail.com, rng@rice.edu

## Abstract

We describe a novel automatic computational framework for evaluating and visualizing the results of infant cranial surgeries. We begin by capturing a 3D triangle mesh of the subject's head using a 3dMD camera. Our framework includes a mesh decimation algorithm to simplify these 3D meshes. Then we register mesh pairs and compare local features at each related vertex pair on pre-surgery and post-surgery meshes. Finally using false color, we visualize the difference between local geometric features before and after surgery. The goal of this visualization is to assist surgeons in evaluating the efficacy of their surgical techniques.

## 1 Introduction

Cranial vault remodeling operations are performed on infants born with craniosynostosis and head shape dysmorphosis. 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 and visualization frameworks that focus on three-dimensional aspects of normal and abnormal morphology (anatomy) are required to evaluate and visualize outcomes accurately. The new 3D image capturing system gives good raw data for this objective analysis. 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 [1].

Previous work has focused on rudimentary linear and angular measurements in the setting of direct or indirect anthropometry or craniometric analysis. This research tries to use simple mathematical metrics to evaluate and visualize the results of Plastic Surgery. This related work will be surveyed in Section 2.

We describe a method that can comprehensively provide regional cranial three-dimensional surface features 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.

The main contribution of our work is to provide a 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. More details of this framework will be covered in Section 3.1.

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. We will give more details concerning mesh simplification in Section 3.2.

To compare local surface features, we need to find a correspondence between mesh pairs. We will introduce the registration method in Section 3.3.

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. We will describe these features in Section 3.4.

Additionally, in Section 4, we will show how to visualize these features and describe some experimental results generated by our framework. Finally, we will summarize the contribution of our framework in Section 5.

## 2 Related Work

With the recent advances of 3D imaging, several researchers have tried to use mathematical metrics to evaluate the surgical outcome in Plastic Surgery.

[12] considers curvature for quantitative assessment of head shape. Curvature is an important local feature for shape analysis. The limitation of [12] 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 Section 3.4.1.

[4] uses normal vectors for head shape analysis. 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. See Section 3.4.2.

[13] 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 opinion as the ground truth for the training set. As a result, the accuracy of their computational assessment will be determined by the doctors' subjective evaluation of cleft lip severity. By 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.

Finally, normative head shape models are an important tool for surgeons to evaluate a subject's condition.Various researchers have constructed such models based on different data sets. For example, [5] generates a fronto-orbital bandeau template from the CT scans of 103 children with normal skulls and uses this template to optimize the outcomes of the surgery. [9] applies three-dimensional vector analysis to generate a set of point clouds and uses the average and standard deviations for the age and gender bins of point clouds to create a normative 3D model.

# 3 Framework

## 3.1 Framework Overview

The outline of our evaluation and visualization system for infant cranial surgeries is summarized in Figure 1.

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 $2 \times 10^6$ 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. In our evaluation system, we can compare 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 all 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.

Finally, we will discuss how to combine these local features to evaluate and visualize the surgery in Section 3.5.

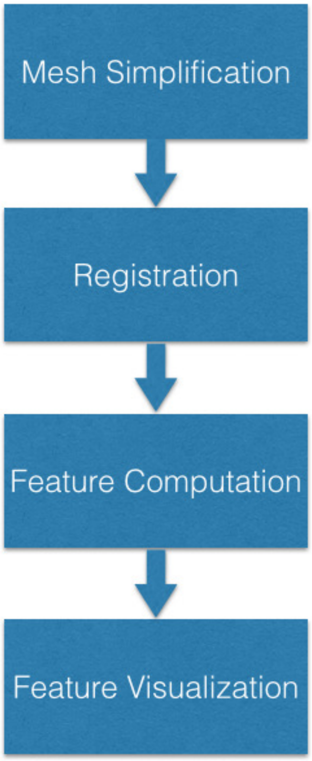


Figure 1: Outline of the evaluation and visualization framework.

## 3.2 Mesh Simplification

[6] 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 [11] 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 [11], 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 ([11]).

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 $X_{N\times 3} = \begin{pmatrix} x_1, & ..., & x_N \end{pmatrix}^T$, and the target point set $Y_{M\times 3} = \begin{pmatrix} y_1, & ..., & y_M \end{pmatrix}^T$, the transformation can be represented as:

$$\mathrm{T}\left(y_j; R, t, s\right) = sRy_j + t \tag{1}$$

where $R$ is a $3 \times 3$ rotation matrix, $t$ is a $3 \times 1$ 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:

$$y_j = f\left(x_i\right) \tag{2}$$

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 underling 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 2 illustrates the geometric meaning of two-dimensional curvature.

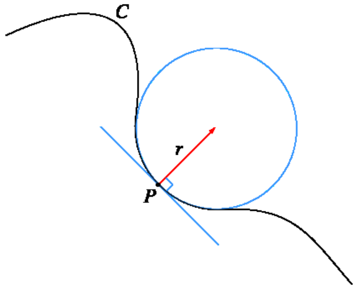


Figure 2: Geometric illustration of two-dimensional curvature.

Given a plane parametric curve represented as:

$$\begin{cases} x = x\left(t\right) \\ y = y\left(t\right) \end{cases} \tag{3}$$

the formula for the curvature $K$ is given by:

$$K = \frac{\left| \frac{dx}{dt} \cdot \frac{d^2y}{dt^2} - \frac{dy}{dt} \cdot \frac{d^2x}{dt^2} \right|}{\left( \left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2 \right)^{\frac{3}{2}}} \tag{4}$$

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 $e_\theta$ in the tangent plane of $P$, we can use the normal vector $n_p$ and the tangent vector $e_\theta$ 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).

$$K_G = K_1 \cdot K_2 \tag{5}$$

$$K_M = \frac{K_1 + K_2}{2} \tag{6}$$

[10] gives an algorithm to compute curvatures for 3D triangle meshes. Many other researchers have improved the accuracy and efficiency for this algorithm. See [7] and [8].

We use the popular computer graphics API, the Point Cloud Library (PCL) [3], 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:

$$N_{\Delta ABC} = \frac{(A-C)\times(B-C)}{|(A-C)\times(B-C)|} \tag{7}$$

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:

$$n_A = \frac{\sum\limits_{\Delta k} N_{\Delta k}}{\left|\sum\limits_{\Delta k} N_{\Delta k}\right|} \tag{8}$$

where $\Delta k$ represents any triangle face that contains vertex $A$, $N_{\Delta k}$ represents the unit normal vector of face $\Delta k$ 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. [4] 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:

$$S_{around}(x_i) = \frac{1}{3}\sum_j S_{\Delta_j} \tag{9}$$

where $\Delta_j$ is any triangle that contains the vertex $x_i$, $S_\Delta$ is the area of triangle $\Delta_j$ 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:

$$\begin{cases} R = 0 \\ G = 2c \quad if\ 0 \le c < 0.5 \\ B = 1 - 2c \end{cases}$$
$$\begin{cases} R = 2c - 1 \\ G = 2 - 2c \ \ if\ 0.5 < c \le 1 \\ B = 0 \end{cases} \tag{10}$$

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:

$$c(K) = 1 - e^{-\alpha \cdot K} \tag{11}$$

where $K$ can be maximal principal curvature, minimal principle curvature, mean curvature or Gaussian curvature, and $\alpha$ is a constant set by experiment. Note $\alpha$ 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:

$$DC(x_i) = 1 - e^{-\beta\left|\frac{K(x_i)-K(y_j)}{K(x_i)+K(y_j)}\right|} \tag{12}$$

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; $\beta$ 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 curvature, and $\beta$ should be set to make the distribution of $DC\left(x_i\right)$ well-proportioned in the range [0,1].

#### 3.5.3 Normal Vector Distribution

Similar to [4], we compute the azimuth angle $\theta$ and the elevation angle $\phi$ for each unit vertex normal vector $n(x_n, y_n, z_n)$ by the following formulas:

$$\theta = \begin{cases} \arccos(\frac{x_n}{\sqrt{x_n^2+z_n^2}}) & if\ z_n \geq 0 \\ -\arccos(\frac{x_n}{\sqrt{x_n^2+z_n^2}}) & if\ z_n < 0 \end{cases} \qquad (13)$$
$$\phi = \arctan\left(\frac{y_n}{\sqrt{x_n^2+z_n^2}}\right)$$

where $\theta$ is in the range $[-\pi, \pi]$ and $\phi$ is in the range $[-\frac{\pi}{2}, \frac{\pi}{2}]$. Then we group the computed angles $\theta$ and $\phi$ of the vertex normal vectors into $12 \times 12$ 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 [4], 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:

$$AS = \frac{LAFS - RAFS}{LAFS + RAFS} \qquad (14)$$

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 $DC\left(x_i\right)$ (see Formula (12)) by the following formula to select such areas:

$$D^*\left(x_i\right) = \begin{cases} 0 & DC\left(x_i\right) < t \\ 1 & DC\left(x_i\right) \geq t \end{cases} \qquad (15)$$

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:

$$DS = \frac{\sum\limits_{i=1}^{M} D^*\left(x_i\right) \cdot S_{around}\left(x_i\right)}{\sum\limits_{k} S_k} \qquad (16)$$

where the term $\sum\limits_{k} S_k$ 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.

## 4 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.

### 4.1 Simplification Results

The quadric mesh simplification method [6] has been implemented by using the open source software MeshLab [2].

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

### 4.2 Registration Results

We use the coherent point drift [11] method to register mesh pairs. Specifically, we use the rigid mode of this algorithm

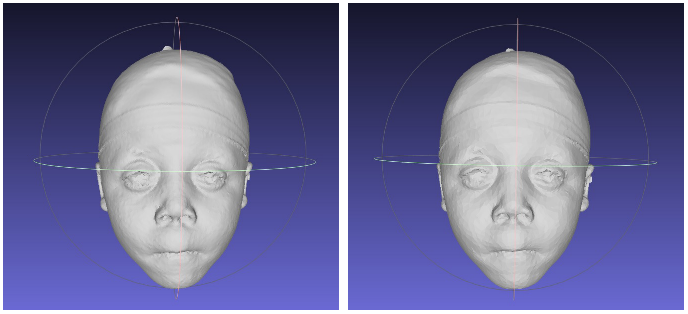


Figure 3: Mesh simplification, left: input mesh, right: simplified mesh

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 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 [11] in MATLAB R2013a with the same system environment. The running time for the registration is summarized in Table 1.

| Suject 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 1: Registration time

## 4.3 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 cranial surgery.

### 4.3.1 Curvature visualization

Using the method described in Section 3.5.2, we generate Figure 5 to visualize curvature and Figure 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 $\alpha$ in Formula (11) to 40, while for the Gaussian curvature, we set $\alpha$ in Formula (11) to 2000. From Figure 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.

When visualizing curvature comparisons, we set $\beta$ in Formula (12) to 2.5 to get the best visual effect. A visualization result can be viewed in Figure 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 6.

### 4.3.2 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 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 2.

| Subject ID | Pre-surgery | | |
|---|---|---|---|
| | LAFS | RAFS | AS |
| Subject 1 | 49.0923 | 52.4496 | -0.03306 |
| Subject 2 | 43.0490 | 38.6312 | 0.05409 |
| Subject 3 | 31.7818 | 39.6151 | -0.10971 |
| Subject 4 | 33.1238 | 26.0269 | 0.11998 |
| Subject 5 | 29.9552 | 22.2420 | 0.14777 |
| **Subject ID** | **Post-surgery** | | |
| | LAFS | RAFS | AS |
| Subject 1 | 57.3033 | 54.2026 | 0.02781 |
| Subject 2 | 67.6544 | 60.4020 | 0.05663 |
| Subject 3 | 60.9226 | 54.5493 | 0.05519 |
| Subject 4 | 51.4286 | 42.5658 | 0.09429 |
| Subject 5 | 35.9186 | 40.1260 | 0.0553 |

Table 2: LAFS, RPFS and AS

Note the absolute value of AS quantified the degree of asymmetry. High AS implies a more asymmetric head shape. The results in Table 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.

### 4.3.3 Relative Changed Surface Area

Table 3 shows 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 $\beta$ in Formula (12) is set to the same value as in the Section 4.3.1 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.

| 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 |
| 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 |
| 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 |

Table 3: Relative changed surface area

## 5 Conclusion and Future work

We described 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 expect that these visualization methods will help the physician to assess the quality of their techniques for infant cranial surgery.

In this paper, we focus on comparisons of head shapes between two distinct time point pairs. A similar methodology can be used to compare a subject's head shape with age-matched normative controls. Our future aim is to generate a robust series of normal head shape models from which age-, gender-, and race-matched controls may be extracted. The comparison 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. The comparison of a post-treatment subject's head shape and an age-matched control would allow for evaluation of a specific treatment approach or surgical technique.

## 6 Acknowledgement

This research is supported by a grant from the Department of Surgery, Texas Children's Hospital, Houston, Texas.

## References

[1] 3dmd. http://www.3dmd.com.

[2] Meshlab. http://meshlab.sourceforge.net.

[3] Pcl. http://pointclouds.org.

[4] I Atmosukarto, LG Shapiro, JR Starr, CL Heike, B Collett, ML Cunningham, and ML Speltz. Three-dimensional head shape quantification for infants with and without deformational plagiocephaly. *The Cleft Palate-Craniofacial Journal*, 47(4):368–377, 2010.

[5] Jonathan Burge, Nikoo R Saber, Thomas Looi, Brooke French, Zoha Usmani, Niloofar Anooshiravani, Peter Kim, Christopher Forrest, and John Phillips. Application of cad/cam prefabricated age-matched templates in cranio-orbital remodeling in craniosynostosis. *Journal of Craniofacial Surgery*, 22(5):1810–1813, 2011.

[6] Michael Garland and Paul S Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 209–216. ACM Press/Addison-Wesley Publishing Co., 1997.

[7] Gaël Guennebaud, Marcel Germann, and Markus Gross. Dynamic sampling and rendering of algebraic point set surfaces. In *Computer Graphics Forum*, volume 27, pages 653–662. The Eurographics Association and Blackwell Publishing Ltd., 2008.

[8] Gaël Guennebaud and Markus Gross. Algebraic point set surfaces. In *ACM Transactions on Graphics (TOG)*, volume 26, page 23. ACM, 2007.

[9] Jeffrey R Marcus, Leahthan F Domeshek, Andre M Loyd, John M Schoenleber, Rajesh R Das, Roger W Nightingale, and Srinivasan Mukundan Jr. Use of a three-dimensional, normative database of pediatric craniofacial morphology for modern anthropometric analysis. *Plastic and reconstructive surgery*, 124(6):2076–2084, 2009.

[10] Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III*, pages 35–57. Springer, 2003.

[11] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(12):2262–2275, 2010.

[12] Chia-Chi Teng, Aaron Hart, Richard Hopper, David Khechoyan, and Jennifer 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*, pages 1–4. IEEE, 2010.

[13] Jia Wu, Raymond Tse, and Linda 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.*

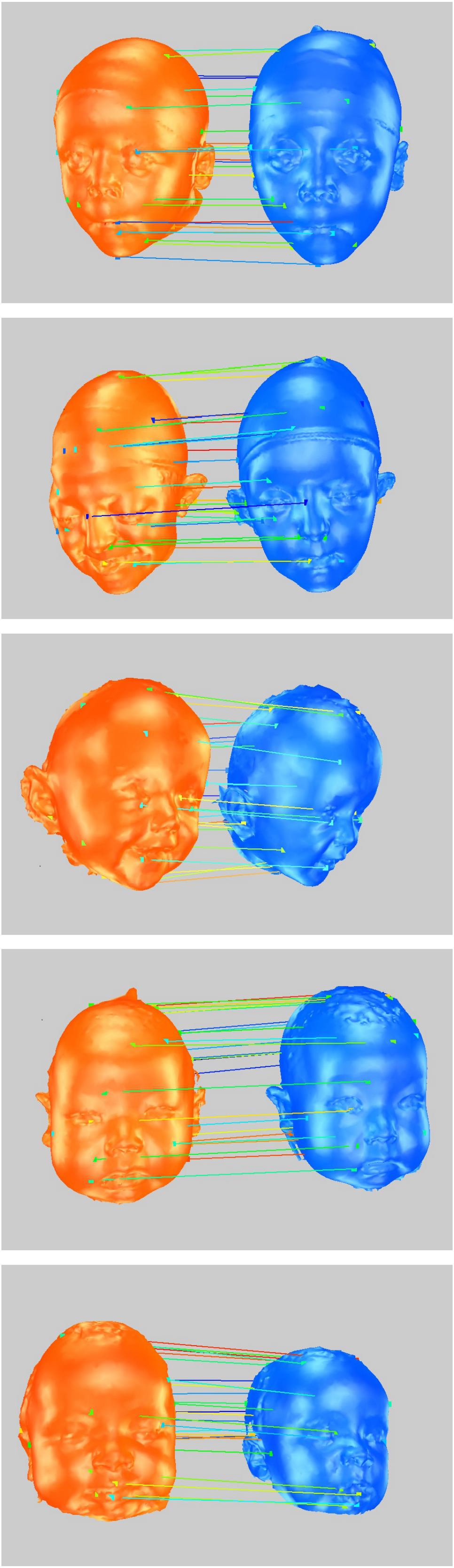


Figure 4: Registration. The orange meshes on the left represent the source meshes and the blue meshes on the right represent the target meshes.

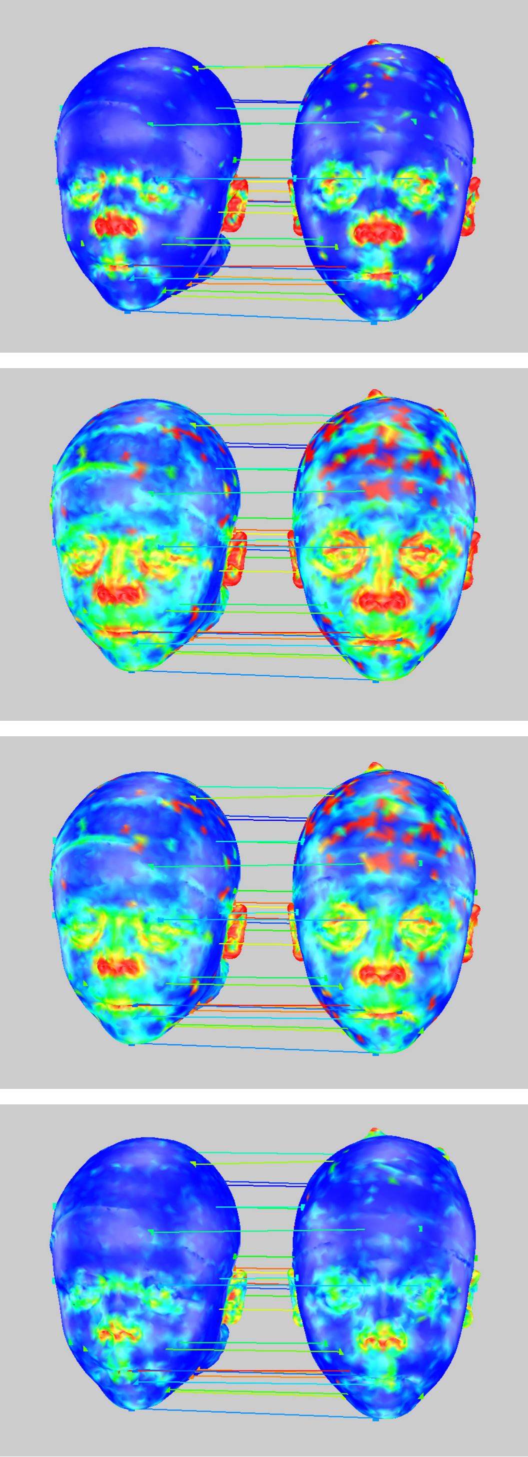


Figure 5: Curvature visualization. From top to bottom: Gaussian curvature, maximal principal curvature, mean curvature and minimal principle curvature. Blue represents low curvature, while red represents high curvature.

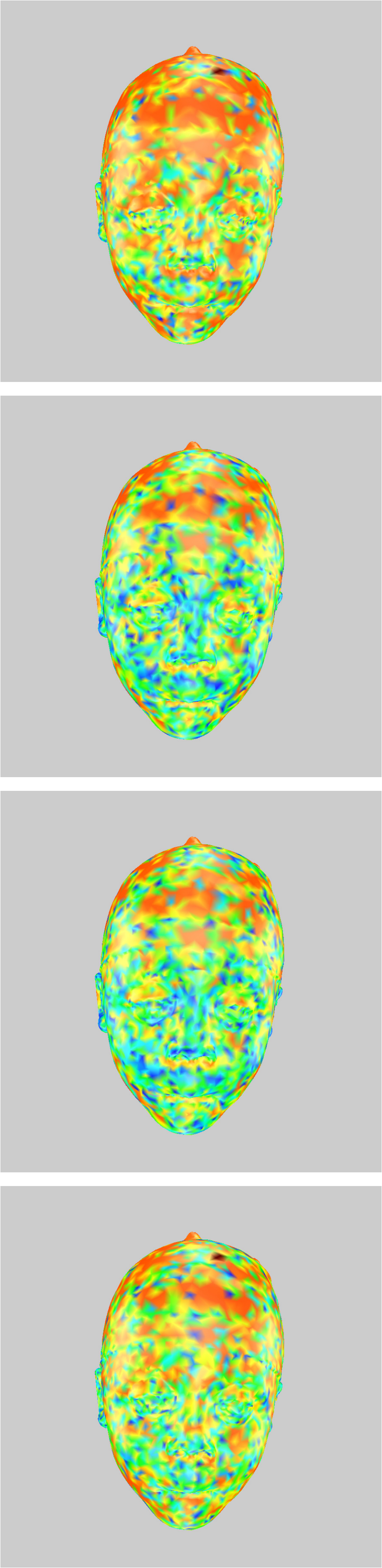
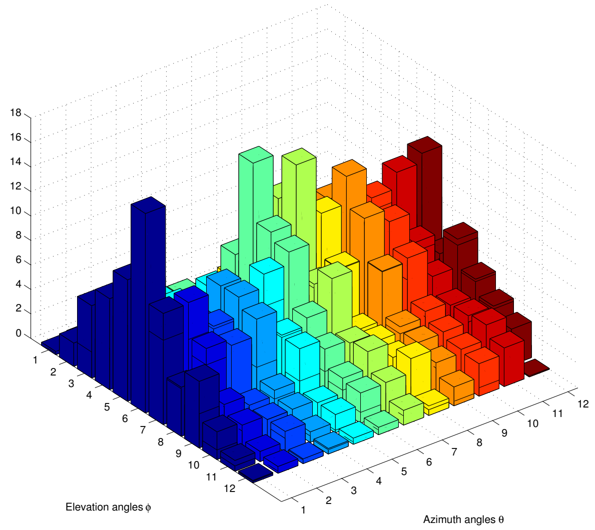


Figure 6: Curvature comparison. From top to bottom: Gaussian curvature, maximal principal curvature, mean curvature, minimal principal curvature. Blue represents low $DC\left(x_i\right)$, while red represents high $DC\left(x_i\right)$.



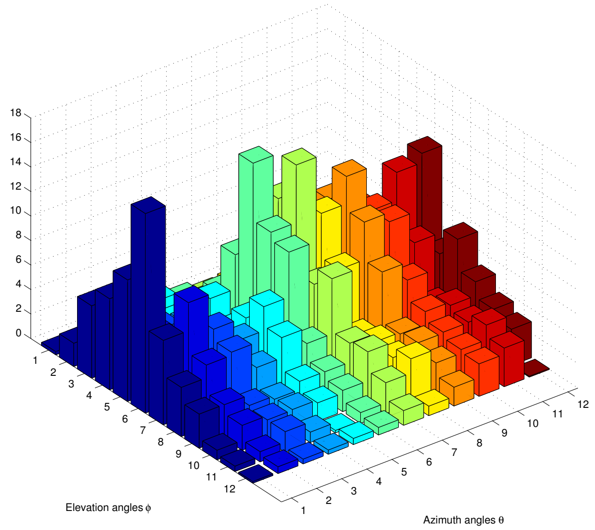


Figure 7: Weighted vertex normal vector distributions. The first figure is the normalized mean curvature weighted distribution for the head shape of subject 1 at time point 1. The second figure is the normalized mean curvature weighted distribution for the same subject at time point 2.

View publication stats