Procedia Computer Science

Volume 80, 2016, Pages 988–998

ICCS 2016. The International Conference on Computational Science

Generating a 3D Normative Infant Cranial Model

Binhang Yuan[1], Ronald N. Goldman[2], Eric Wang[3], Olushola Olorunnipa[4], and David Y. Khechoyan[5]

[1] Rice University, Houston, Texas, U.S.A
by8@rice.edu
[2] Rice University, Houston, Texas, U.S.A
rng@rice.edu
[3] Baylor College of Medicine, Houston, Texas, U.S.A.
eric.wang@bcm.edu
[4] Texas Children's Hospital, Houston, Texas, U.S.A.
sxolorun@texaschildrens.org
[5] Texas Children's Hospital, Houston, Texas, U.S.A.
David.Khechoyan@bcm.edu

Abstract

We describe an algorithm to generate a normative infant cranial model from the input of 3D meshes that are extracted from CT scans of normal infant skulls. We generate a correspondence map between meshes based on a registration algorithm. Then we apply our averaging algorithm to construct the normative model. The goal of this normal model is to assist an objective evaluating system to analyze the efficacy of plastic surgeries.

Keywords: 3D Normative Model, Infant Plastic Surgery, Objective Evaluation, Cranial Vault Remodeling, Point Set Registration, Point Set Averaging Algorithm

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 physician and are confounded by high inter- and intra-observer error and lack of concordance. Previously, we have proposed a framework that can comprehensively provide regional and global cranial three-dimensional geometric features with corresponding visualizations to aid the surgeons in evaluating the severity of the deformity (as it compares to normative cranial forms) and the changes that had occurred with the surgical intervention [16].

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

 Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2016

doi:10.1016/j.procs.2016.05.394

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 [1] 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, some medical centers have accumulated some 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.

The automatic registration algorithm provides a 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.

The main contribution of our work is that our method 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 [16] can be extended to a comparison between abnormal and normal head shapes, so the extended framework naturally inherits the merits of the original framework.

This paper is organized as follows: we first overview related work about normative head shape modeling in section 2. We introduce our method for generating a normative model in Section 3. Additionally, in Section 4, we show our pilot experimental results. Finally, we close in Section 5 by summarizing the main contributions of our method.

2 Related Work

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 [4] 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 [8] 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 [9] 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 [15] 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 craniosynostosis [10, 7, 5, 13]. Burge et al [5] 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 [7] 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 [1] meshes of the subjects' head.

McComb [11] 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.

3 Normative Model Generation

In this section, 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. We will first give an overview and then introduce each module separately.

3.1 Algorithm Overview

The framework of our normative model for normal infant cranial shape is summarized in Figure 1.

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 with 3D images from stereophotography of the subject's head for diagnosis and objective evaluation. To accomplish this goal, we use the tools of DeVIDE [2] and MeshLab [3]. The details of this part will be covered in Section 3.2.

The registration module plays an important role in this system. Registration provides the correspondence between pairs of 3D meshes. We will give a brief review of this method in Section 3.3.

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 3.4.

3.2 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 [2] to extract a rough surface mesh with noisy fragments and then use MeshLab [3] to refine the mesh for further analysis.

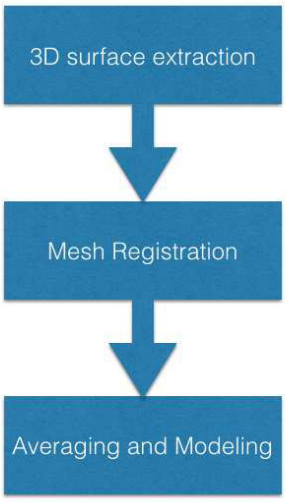


Figure 1: Framework for generating the normative model.

DeVIDE Operating Pipeline. DeVIDE [2], 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 purposes. Figure 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 3 shows the extracted result of a subject's head shape.
- StlWRT module writes the mesh in STL format back to the disk.

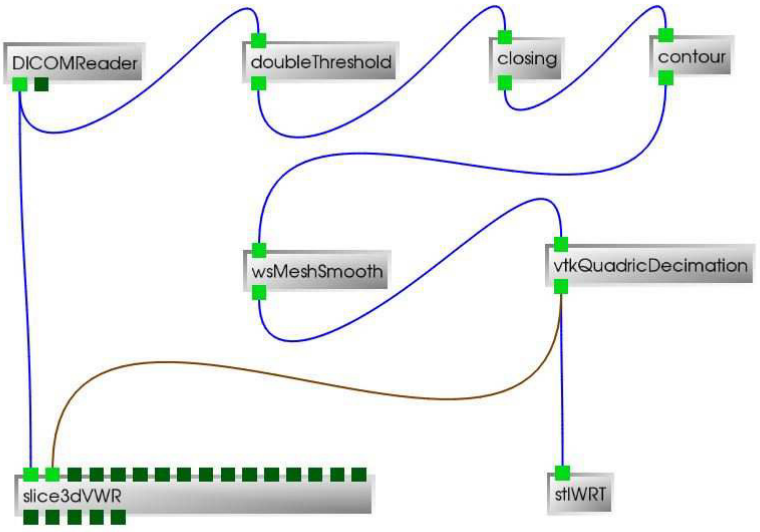


Figure 2: Pipeline of the DeVIDE software.

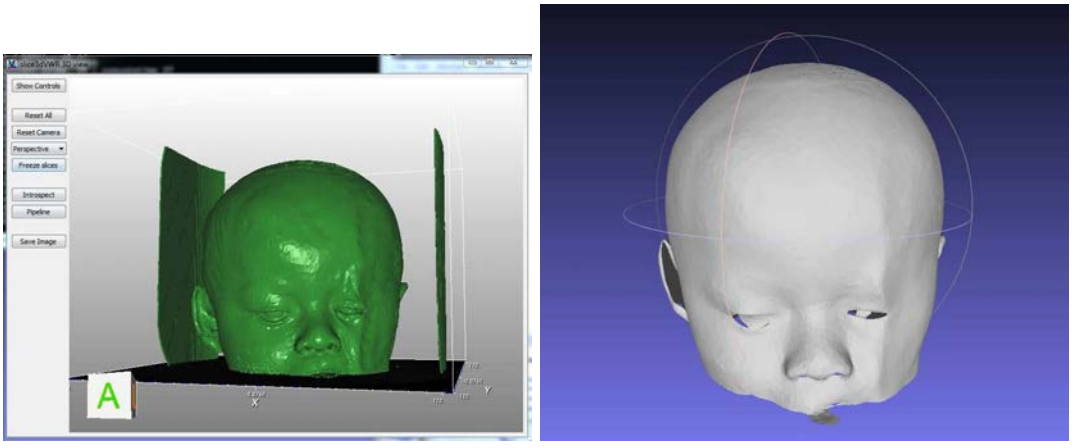


Figure 3: Surface extraction. Left: a surface we extracted after applying DeVIDE. The surface contains some noisy fragments and is constructed by a few closely connected layers. Right: a surface we extracted after applying MeshLab. The surface is constructed by a single layer without noisy fragments.

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 closely connected layers, whereas we want a single layer surface for further analysis.

To resolve the first issue, we use the tool "Select Connected Components in a region" in

MeshLab to select the main part of the mesh and eliminate the noisy fragments.

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 3 illustrates the refined result of a subject's head shape.

3.3 Point Set Registration

For this module, we use the coherent point drift method described in [12] to implement registration. This method takes the input of two point sets. The registration assigns correspondences between two point sets and computes the transformation that maps one point set to the other.

In [12], the alignment of two point sets is viewed as a probability density estimation problem, where the first point set (*source point set*) represents the Gaussian mixture model cluster centroids and the second point set (*target point set*) represents the data points. The correspondence probability between two point sets is considered as the posterior probability of the Gaussian mixture model cluster centroid given the data points. This method uses the 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 by 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 ([12]).

The output of this registration method 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 3×3 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:

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

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 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, 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 the 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:

$$\begin{cases} \bar{x}_i = (1-\theta)\,\bar{x}_{i-1} + \theta x_i \\ \bar{y}_i = (1-\theta)\,\bar{y}_{i-1} + \theta y_i \\ \bar{z}_i = (1-\theta)\,\bar{z}_{i-1} + \theta z_i \end{cases} \tag{3}$$

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

$$\theta = \frac{1}{i} \cdot \frac{1}{e^{\alpha d}} \tag{4}$$

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, 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 [14, 6] 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:

$$\begin{cases} x_i = \frac{1}{n} \sum_{j=1}^{n} x_j \\ y_i = \frac{1}{n} \sum_{j=1}^{n} y_j \\ z_i = \frac{1}{n} \sum_{j=1}^{n} z_j \end{cases} \tag{5}$$

where n is the number of adjacent vertices to node i, (x_j, y_j, z_j) is the position of the jth 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 Pilot Experimental Results

In this section, we will describe our pilot experimental results. Based on our survey, in order to protect each subject's privacy, no public data resource of normal infant cranial CT scans is available. With the help of our collaborators, we collected 8 series of CT scans from Texas Children's Hospital with permission for research. We set the α in Formula 4 to 0.1, because this parameter value leads to good experimental results. We applied the above pipeline to generate

Algorithm 1 Average Mesh Set

```
Select one mesh M_0 as a base from the input mesh set S
Delete M_0 from S
R ← M_0
i ← 1
for all M_i ∈ S do
    Register R (as source) to M_i (as target) by [12].
    Merge M_i into R by Formula 3
    Apply Laplacian smoothing on R.
    i ← i + 1
end for
return R
```

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.

4.1 Spacial Coherence

The Laplacian smoothing method [6] has been implemented by using the open source software MeshLab[3] . 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 illustrates the effect of Laplacian smoothing. From Figure 4, we observe that Laplacian smoothing leads to good spacial coherence in our model.

4.2 Model Convergence Discussion

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 listed in Table 1. The computation of each measurement is introduced below:

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

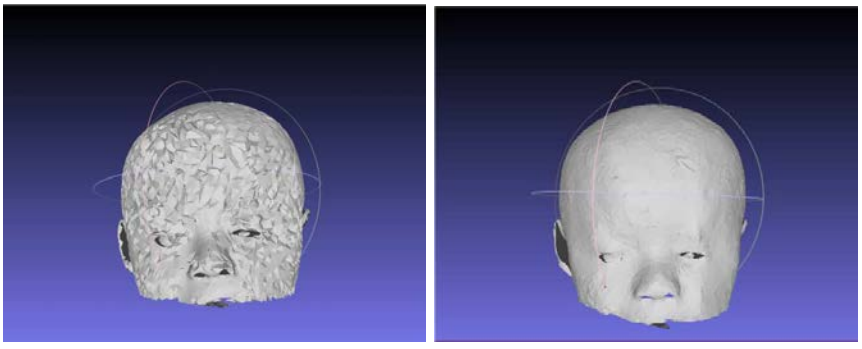
$$D_{coordinate} = \frac{1}{N}\sum_{k=1}^{N}\sqrt{\left(x_k^{(i)} - x_k^{(i-1)}\right)^2 + \left(y_k^{(i)} - y_k^{(i-1)}\right)^2 + \left(z_k^{(i)} - z_k^{(i-1)}\right)^2} \tag{6}$$

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:

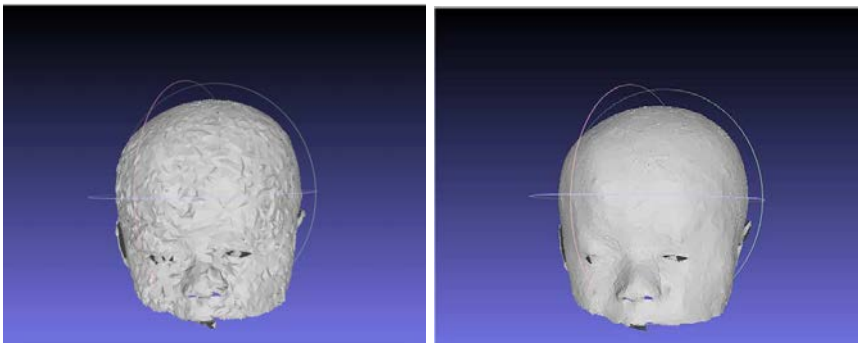
$$D_{normal} = \frac{1}{N}\sum_{k=1}^{N}\arccos\left(\frac{v_k^{(i)} \cdot v_k^{(i-1)}}{\left|v_k^{(i)}\right|\left|v_k^{(i-1)}\right|}\right) \tag{7}$$

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



(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: Laplacian smoothing. Laplacian smoothing leads to good spacial coherence.

model.

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

$$D_{mean} = \frac{1}{N}\sum_{k=1}^{N}\left|K_{M_k}^{(i)} - K_{M_k}^{(i-1)}\right| \tag{8}$$

$$D_{Gaussian} = \frac{1}{N}\sum_{k=1}^{N}\left|K_{G_k}^{(i)} - K_{G_k}^{(i-1)}\right| \tag{9}$$

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 1, it appears that our averaging algorithm converges slowly as the number of iterations increases. This situation is expected since in Formula 4, the updating weight decreases as the number of averaged meshes increases. We plan to perform further experiments and analysis when more CT series are provided by Texas Children's Hospital.

5 Conclusions

We 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. The normal cranial shape generation method plays a critical role in an objective surgery evaluation system. The degree of divergence from normal heads provides surgeons with significant information for plastic surgery evaluations.

In this paper, 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.

References

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

[2] Devide. https://code.google.com/p/devide/.

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

[4] J Brief, St 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*, 5(5):353–358, 2000.

[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] Glen A Hansen, Rod W Douglass, and Andrew Zardecki. *Mesh enhancement: selected elliptic methods, foundations and applications*. Imperial College Press, 2005.

[7] David Y Khechoyan, Nikoo R Saber, Jonathan Burge, Adel Fattah, James Drake, Christopher R Forrest, and John H Phillips. Surgical outcomes in craniosynostosis reconstruction: The use of prefabricated templates in cranial vault remodelling. *Journal of Plastic, Reconstructive & Aesthetic Surgery*, 67(1):9–16, 2014.

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

[8] Jeffrey R Marcus, Leahthan F Domeshek, Rajesh Das, Sean Marshall, Roger Nightingale, Tracey H Stokes, and Srinivasan Mukundan Jr. Objective three-dimensional analysis of cranial morphology. *Eplasty*, 8, 2008.

[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] Samir Mardini, Saad Alsubaie, Cenk Cayci, Harvey Chim, and Nicholas Wetjen. Three-dimensional preoperative virtual planning and template use for surgical correction of craniosynostosis. *Journal of Plastic, Reconstructive & Aesthetic Surgery*, 67(3):336–343, 2014.

[11] Ryan Walsh McComb. An exploratory approach for mapping the surface of the human skull in three dimensions: Technical methods and clinical application. *UCLA: Oral Biology 0615*, 2013.

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

[13] Jehuda Soleman, Florian Thieringer, Joerg Beinemann, Christoph Kunz, and Raphael Guzman. Computer-assisted virtual planning and surgical template fabrication for frontoorbital advancement. *Neurosurgical focus*, 38(5):E5, 2015.

[14] Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and H-P Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 175–184. ACM, 2004.

[15] Krishna Subramanyan and David Dean. A procedure to average 3d anatomical structures. *Medical image analysis*, 4(4):317–334, 2000.

[16] Binhang Yuan, David Y Khechoyan, and Ron Goldman. A new objective automatic computational framework for evaluating and visualizing the results of infant cranial surgery. *ASE International Conference on Biomedical Computing*, 2015.