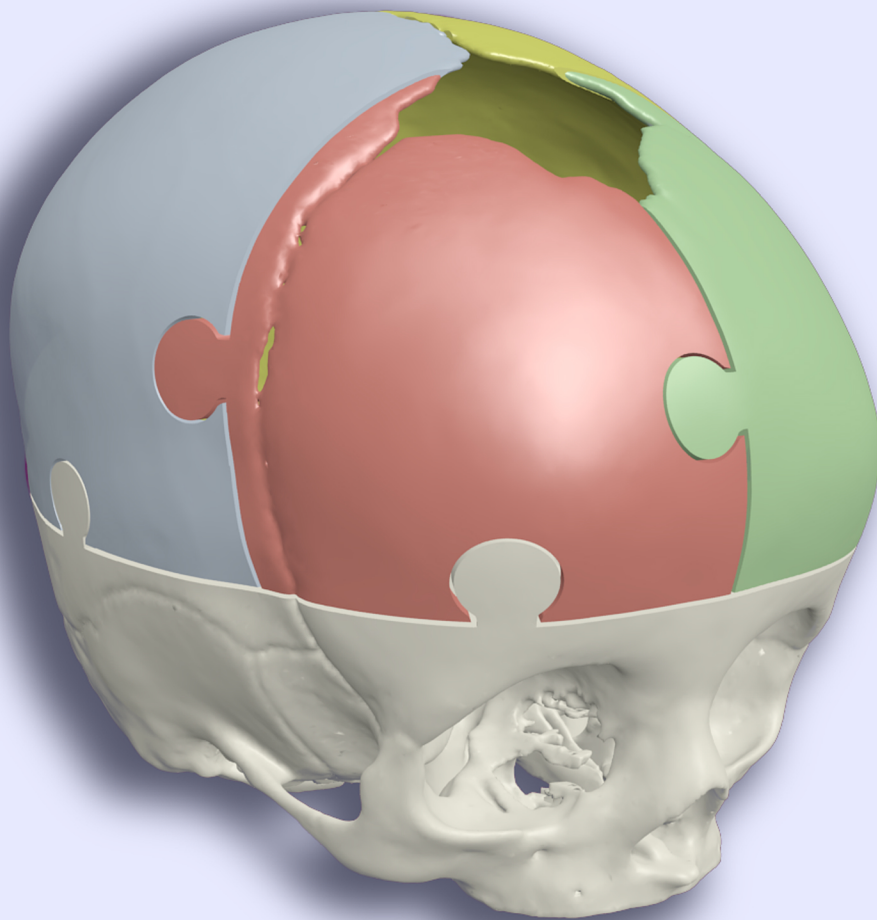


# Towards an objective automated pre-operative 3D planning technique for open cranial vault reconstruction in craniosynostosis patients

Author: M.L. Tolhuisen



**UNIVERSITY  
OF TWENTE.**

**Radboudumc**



## GENERAL INFORMATION

### **Master' s thesis**

Period: October 2015-September 2016  
Department: Neurosurgery, Radboudumc, Nijmegen  
Course: Medical Imaging and Intervention, Technical Medicine,  
University of Twente, Enschede

### **Graduation committee**

Chairman: Prof. I.A.M.J. Broeders  
*Department of Surgery, Meander Medical Centre, Amersfoort*  
&  
*Faculty of Science and Technology Institute Technical Medicine,  
University of Twente, Enschede*

Clinical supervisor: Dr. H.H.K. Delye  
*Department of Neurosurgery, Radboud University Medical  
Centre, Nijmegen, The Netherlands*

Technical supervisor: Dr. F. van der Heijden  
*Department of Robotics and Mechatronics, University of  
Twente, Enschede*

Process supervisor: Dr. M. Groenier  
*Department of Technical Medicine, University of Twente,  
Enschede*

Additional member: Dr. R.M.J. van Damme  
*Department of Applied Mathematics, University of Twente,  
Enschede, The Netherlands*



## ABSTRACT

**[Background]** 3D virtual planning of open cranial vault reconstruction is used to simulate and define an operative plan for craniostygnosis surgery pre-operatively. However, virtual planning techniques are subjective and dependent on the experience and preferences of the surgical team. In order to enable further development of a truly objective automated 3D pre-operative planning technique for open cranial vault reconstructions, curvature maps were used for the local shape comparison of the patient's skull with an age-specific normative skull model. **[Methods]** Mesh data of skulls were created using marching cubes. All skulls were oriented similarly within the used coordinate system and ray casting was used to obtain sampled 3D metrical data of the inner and outer layer of the skulls. Normalised skulls were created for the age groups of 0-1, 2-4, 5-7, 8-10, 11-14, 15-18, 19-24, 25-36, 37-48 months. Also, a test object was created and a cranial CT-scan of a 11 months old trigonocephaly patient was selected. Gaussian curvature was estimated with quadric surface fitting and curvature maps were computed. The shape comparison was tested for the test object and within the normalised skull of the 11-14 age group. Finally, shape comparison was performed for the trigonocephalic skull with the age-appropriate normalised skull. **[Results]** Normalised skulls and thickness maps were created for the defined age groups. Similar shapes were correctly identified and the identification of the region on the patient's skull, that maximally corresponded in shape with the reference shape, was feasible. **[Conclusion]** It is shown that curvature maps allow the shape comparison of craniostygnosis skulls with age-appropriate normative skulls. This study showed the first step towards an objective user-independent pre-operative planning technique for open cranial vault reconstructions.



# 1 GENERAL INTRODUCTION

Craniosynostosis is the result of the premature fusion of one or more sutures within the new-born's skull. The prevalence of craniosynostosis in the Netherlands is between 1:2100 and 1:2500 births and is therefore a rare disease[1], [2]. Several forms of craniosynostosis are known (Figure 1) [3]. The existence of each form is dependent on which suture is fused. Growth of the skull manifests perpendicular to the suture. When fusion of a suture occurs, growth in perpendicular direction of that suture will be hindered. Compensatory growth at the location of the other sutures will cause a deformed skull shape. This can lead to several complications, such as increased intracranial hypertension with subsequently mental retardation and malfunction of the eyes, ears and respiratory system. [1], [4]–[7]

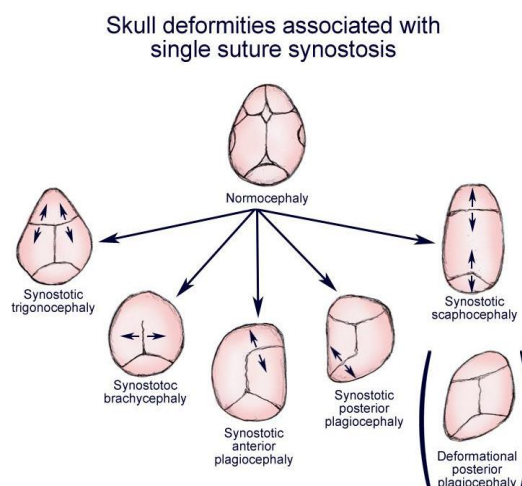


Figure 1 Craniosynostosis forms, compensatory growth is represented by the black arrows, source: Raj et al. [3]

Patients of the Radboud University Medical Centre (Radboudumc) with craniosynostosis that are older than 6 months are subjected to open cranial vault reconstruction. The goal of this treatment is to prevent or treat increased intracranial pressure by increasing the intracranial volume, but also to restore the cranial shape[6], [8]. During the procedure, the cranial vault is divided into several osseous panels which are relocated and fixed to form a new cranial vault shape. Many different reconstruction techniques have been described for different forms of craniosynostosis and no consensus on the best technique exists[6], [9]–[17]. Therefore, the location of cutting lines and relocation of the panels is dependent on the experience and preference of the surgical team and the results will vary between surgical teams [18], [19].

Current research focuses on the development of pre-operative planning techniques. Mardini et al.[20] and Soleman et al.[21] pre-operatively perform a virtual reconstruction to determine the cutting lines and reallocate the created osseous panels pre-operatively. Also, Clijmans et al[22] present a semi-automated technique that is used for the determination of the cutting lines. Within their algorithm several reconstruction protocols are known and the most suitable protocol is automatically detected. At the Radboudumc a pre-operative plan is created as well. Following standard clinical practice, a cranial CT-scan of the patient's skull is made prior to the surgery. The volumetric data of the skull is obtained from the CT-data and represented in a 3D mesh. With help of the surgeon, all surgical options are virtually evaluated and the optimal cutting lines are determined. With the use of a pre-operative plan for open cranial vault reconstruction, the reproducibility will increase and the operation time, blood loss and infection rates will decrease [20], [21], [23].

To our knowledge still no fully automated objective planning technique is available. Created pre-operative planning techniques are yet based on the interpretation of the surgical team. The aim of this study is to set the first steps towards a user-independent automated pre-operative planning technique that fully eliminates subjectivity. This planning technique will merely consist of an objective evidence-based algorithm allowing for the reconstruction to establish a shape of the patient's cranial vault that maximally corresponds with an age-appropriate reference skull. This study developed an algorithm that enables local shape comparison of the patient's skull with an 3D age-appropriate reference skull. Chapter 2 describes the method for the creation of comparative mesh data that were used in both methods described the subsequent chapters. Chapter 3 and 4 elaborate on the creation of normalised skull models and the algorithm for the local shape comparison of skulls, respectively.





## 2 COMPARATIVE MESH DATA

### 2.1 INTRODUCTION

To enable the comparison of skulls, all skulls should be similarly represented within the coordinate system. Not only should the skulls be similarly oriented within the coordinate system, the skulls should also be similarly sampled. This provides knowledge about the location of data points within the coordinate system and enables comparison of similar data points of different skulls. In this study CT-scans were used to obtain information of skulls. Due to the high contrast of bone with surrounding tissue, skulls can accurately be segmented from the voxel data and represented by mesh data. For this study the use of mesh data was superior to intensity-based data due to the low processing time of mesh data. This chapter focusses on a method for the similar representation of all skulls within a reference coordinate system.

### 2.2 MATERIALS AND METHODS

#### 2.2.1 Materials

The algorithm for the creation of comparative mesh data was implemented in MATLAB 2016a®. GPU integration was used in order to decrease computation time and memory usage. The icosphere was created in 3ds Max 2016®. CT-scans were obtained from the hospital database and were provided as DICOM files.

#### 2.2.2 Methods

For each cranial CT-scan the DICOM data were loaded and segmentation of the skull was applied, based on a 167 Hounsfield unit (HU) threshold. Small isolated high intensity pixels were removed and a Gaussian filter was applied to reduce noise in all slices. Marching cubes, based on the same 167 HU threshold, were used to create 3D mesh data (for the elaboration on marching cubes see supplement 7.1) [24].

A graphical user interface was implemented enabling the user to select the centre of the sella turcica, nasion and both anterior clinoid processes (Figure 2). The registration algorithm places the centre of the sella turcica at the origin of the coordinate system and the sella-nasion line onto the y-axis. Based on both anterior clinoid processes the skull is rotated to place the skull in a straight up symmetrical position, with the  $x, y$ -plane being horizontal.

To obtain evenly distributed data points of the inner and outer surface of the skull the mesh data was resampled using raycasting. A triangular unit hemi-icosphere with 20641 vertices and 43284 faces was created within 3ds Max 2016® (for more information on icospheres see supplement 7.2). The location of the vertices were used to determine the direction of the rays. For each vertex of the hemi icosphere, a ray was casted from the origin through this vertex. The Möller-Trumbore technique was used to find all intersections of the rays with the original mesh of the skull (for the elaboration on Möller-Trumbore see supplement 7.3). The outer and second outer intersection were selected because these intersections form the intersections with the inner and the outer layer of the mesh respectively. Both the hit distance and position of the hit were stored for each ray. Due to the presence of fontanelles or holes within the skull it was possible that no

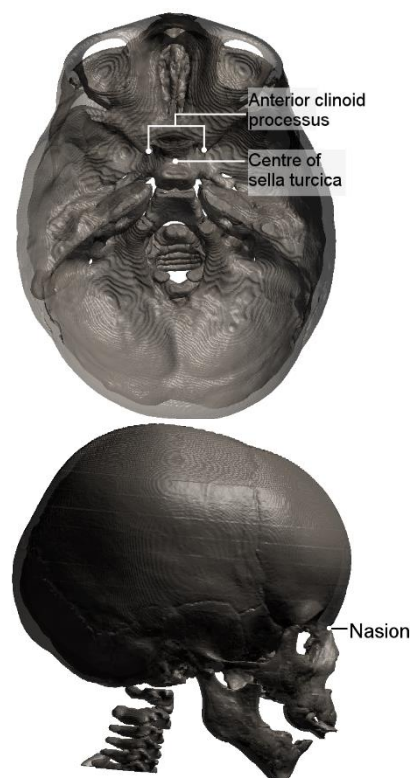


Figure 2 Registration landmarks; (Top) Centre of sella turcica and anterior clinoid processes, (Bottom) Nasion

intersections with the inner or outer layer were found. For these rays the intersections with the inner and outer layer were stored as *NaN* values. Only intersections with a minimal distance from the origin were selected to avoid that the outer and second outer intersection were found within another structure.

Because this study only focusses on the shape comparison with no discrimination of fontanelles or holes within the skull, extrapolation was used to fill in all the holes within the mesh data. This extrapolation followed an iterative process (see *Pseudocode 1*). New positions for missing vertices were based on the hit distance of neighbouring vertices. For each iteration, only vertices at the borders of the holes were filled because only for these vertices the hit distance of their neighbours were known. With every iteration, the border of the holes shift to the centre of the holes till all positions of the vertices were estimated and the holes were filled. The number of iterations for this study was 200.

### *Pseudocode 1*

```

Create list with vertexIds of vertices with NaN values
Repeat n iterations:
  For each vertexId compute the mean position of neighbouring
  vertices:
    Find faceIds in FaceList with vertexId
    Get vertexIds from FaceList for faceIds
    For vertexIds:
      If position vertex with vertexId is not NaN
        Place position in array
    Compute mean hit distance r
    Get direction ray  $(\theta, \varphi)$  of vertexId
    Compute new position for vertexId:
       $x = r \cos \theta \sin \varphi$ 
       $y = r \sin \theta \sin \varphi$ 
       $z = r \cos \varphi$ 

```

The inner and outer layer intersections were stored separately. For each separate layer, the found intersections with that layer and the faces of the icosphere formed the sampled mesh data of the respective layer. The sampled mesh data of both layers for each skull were stored within the research database.

## 2.3 RESULTS

Figure 3 shows the results of the segmentation, registration and sampling of a skull, obtained from a single CT-scan. The inner and outer layer of the skull are shown in Figure 3(*Left*). This figure shows that the skull is oriented in a straight up position with the  $x, y$  -plane being horizontal. The sella-nasion line is located at the  $y$  -axis with the centre of the sella turcica located at the origin. Figure 3(*Right*) shows the result of the sampling. The data points are distributed over the surface of the skull according to the intersection of the rays with the original mesh. The results show an almost even distribution of the data points on the surface of the skull.

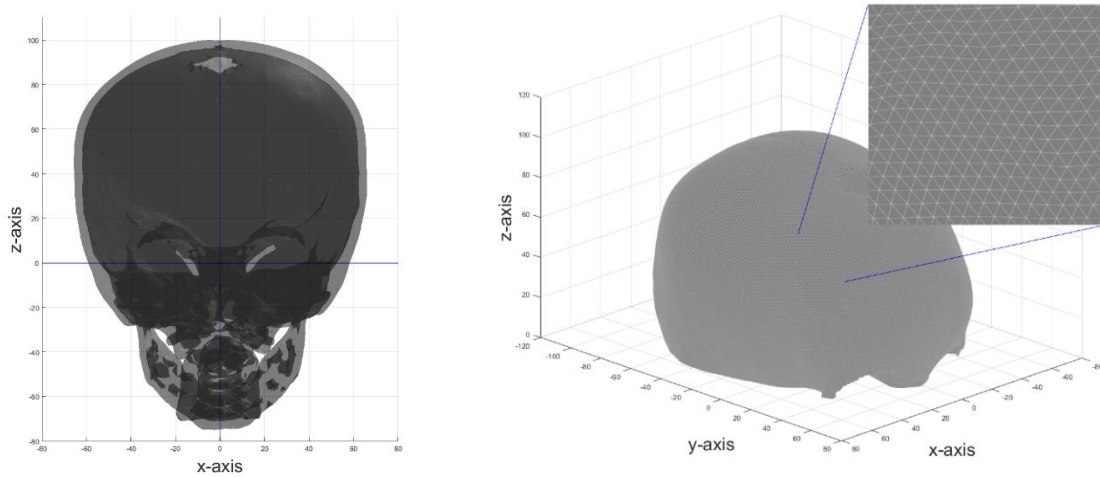


Figure 3 (Left) Inner and outer layer of a skull on which registration is applied (Right) Sampled skull with a small area of the mesh data in more detail

## 2.4 DISCUSSION

For all sampled skulls, the vertex numbering is similar to the numbering of the icosphere. This enables comparison of vertices of different skulls with corresponding vertex numbers. However, for the implementation of algorithms that process the mesh data of skulls, variance in the mesh structure should be taken into account. First, the valence is not equal for all vertices. The connectivity of the vertices of the sampled skulls is similar to the vertices of the icosphere used within the raycasting method. Figure 4 shows the connectivity of the vertices of a simple icosphere. For most vertices the valence is 6, but exceptions exist for which the valence is only 5. This is important knowledge for the implementation of algorithms that are based on the connectivity of the vertices. The used icosphere contained 6 vertices with valence 5. Also, Figure 5 shows that if rays are casted from a single point in different directions and the intersections are found at different distances, variance in the resolution of the mesh data exists. Because skulls are not perfect spherical objects and the intersections will be found at different distances, algorithms should account for variance in the resolution of the mesh data.

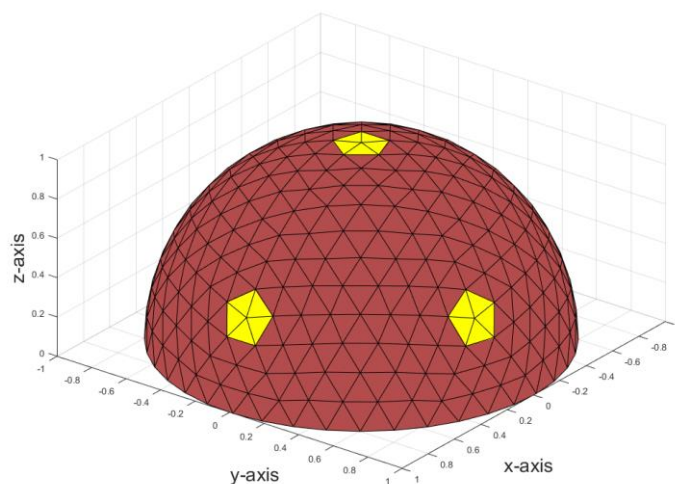
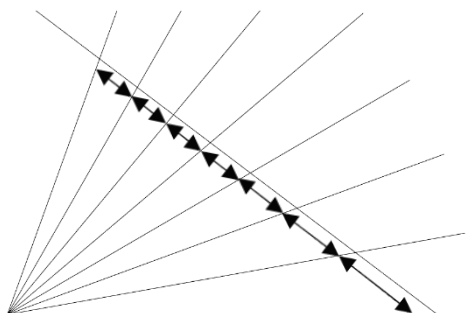


Figure 4 Icosphere (4 iterations) with the neighbouring faces of vertices with valence 5 highlighted



*Figure 5 Ray cast sampling: when all rays are casted from a central point, even though evenly distributed, an irregular spatial resolution of sampling points will be found on the target*

This study did not focus on the improvement of the registration method. Remarks were already made in [25] concerning the accuracy of the method. The use of the anterior clinoid processes as registration landmarks is questionable. The close relative location of the clinoid processes to the sella turcica make the registration algorithm prone to errors. Also, literature describes variation of the anterior clinoid processes between healthy individuals and the influence of craniosynostosis on the cranial base anatomy has been described[26]–[28]. This was confirmed by the researchers who performed the registration in this study. In some cases the clinoid processes were indistinguishable and discrepancy existed in the best selection of the landmarks. Therefore, a large intra- and inter-observer variability is suspected due to difficulties in the selection of the landmarks.

The development of an automated registration algorithm for skulls is challenging. A recent study at the department of Neurosurgery of the Radboudumc already focussed on the development of an automated registration algorithm for skulls, using an iterative closest point algorithm and intrinsic shape signatures. This algorithm places an object in the reference coordinate system based on a target object in which reference landmarks are selected. The study showed the feasibility of the algorithm for objects in which stable landmarks can be identified. However, no sufficient stable landmarks on the skull are known. Also, the algorithm was only tested for objects with a similar shape as the target object. Because the shape of skulls differs, especially in case of craniosynostosis, the algorithm should be robust for shape deformities. Further research should focus on the identification of stable landmarks on the skull and the development of a robust automatic registration algorithm that is feasible for both healthy skulls and skulls with craniosynostosis.

## 2.5 CONCLUSION

This study implemented a method for the similar representation of skulls within a reference coordinate system to enable the comparison of skulls.

## 3 NORMALISED SKULLS

### 3.1 INTRODUCTION

The goal of open cranial vault reconstructions in craniosynostosis patients is to restore the cranial shape and intracranial volume. However, no age-appropriate 3D reference shape and volume are available. The only objective parameters that are used are 2D measurements, such as the cephalic index for scaphocephaly and the frontal angle for trigonocephaly[29], [30]. Delye et al.[31] have set the initial step in the creation of an age-specific 3D database, containing normative data of geometrical measurements, bone thickness, and bone density of the paediatric skull. However, the geometrical measurements consist only of geodesic length measurements. As shape and volume are 3D properties, they cannot be sufficiently described by 2D measurements. Consequently, reconstructions are mainly performed according to the interpretation of the surgical team on what is a normal head shape, i.e. what is beauty. This is affirmed by literature, whereas the surgical results are often evaluated based on the satisfaction of the surgical team and parents with the shape of the head[9]–[15]. In order to create a fully objective pre-operative plan, the reconstruction should be based on objective 3D reference data. This study creates normalised paediatric skull models that can be used as age-specific reference in the creation of pre-operative plans.

### 3.2 MATERIALS AND METHODS

#### 3.2.1 Materials

The method that establishes the normative skull models was implemented in MATLAB 2016a®. Due to the large amount of memory use and computation time, OpenCL was used for GPU integration. CT-scans were obtained from the hospital database and were provided as DICOM files.

#### 3.2.2 Methods

For the establishment of normative skull models, paediatric CT-scans were collected from the medical database. All cranial CT-scans of patients with the age between 0 and 48 months were selected and screened. Only CT-scans with a maximal slice thickness of 3 mm and in which the nasion and the entire cranium from the level of the sella turcica were present, were included. All scans of patients with disease that could have had effect on the shape of the cranium, such as space occupying lesions, metabolic bone disease, fractures and malformations were excluded from the dataset. Due to the limited amount of included scans, ages were clustered. Age groups were based on the ages of patients at the Radboudumc during surgery and follow-up, and the growth curves of the skull described by Delye et al.[31] and Waitzman et al.[32]. The age groups were 0-1, 2-4, 5-7, 8-10, 11-14, 15-18, 19-24, 25-36, 37-48 months.

All scans were processed as described in chapter 2. Because the vertex numbering for the inner and outer layer was similar for all skulls, normalisation per vertex number was possible. For each vertex number, the coordinates of the vertices for the separate layers of the skulls within the same age group were normalised according to the following equation:

$$v(x, y, z)_{norm} = \frac{\sum_{i=1}^{N_v} v_i(x, y, z)}{N_v}$$

$N_v$  is the number of vertices stored for the respective vertex number, excluding *NaN* coordinates. The normalisation was only applied for the vertex numbers for which  $N_v > 0$ , i.e. only vertex numbers for which at least one skull contained an intersection coordinate with the respective layer were normalised. The maximum value of  $N_v$  was the number of skulls within the age group. The normalised vertices of the vertex numbers for which no skulls contained an intersection coordinate were stored as *NaN*. The normalised vertices and the corresponding faces remained numbered similarly to the hemi icosphere. Also, the thickness of the normalised skulls was obtained by computing the distance of the inner and

outer layer. For each vertex number that contained an intersection with both the inner and outer layer of the skull, the distance was computed:

$$distance = \sqrt{(x_{out} - x_{in})^2 + (y_{out} - y_{in})^2 + (z_{out} - z_{in})^2}$$

$(x_{in}, y_{in}, z_{in})$  and  $(x_{out}, y_{out}, z_{out})$  are the coordinates of the inner layer vertex and outer layer vertex, respectively.

### 3.3 RESULTS

Table 1 shows the number of CT-scans used per age group to obtain the normalised skulls. All outer layers of the normalised skulls were plotted in a single figure to show the growth of the normalised skulls over time. Figure 6 shows the cross-sections of this overlay. The results show that most growth of the skull occurs in the first 7 months, which agrees with the results shown by Delye et al.[31]. Figure 7, Figure 8 and Figure 9 show the thickness maps for all age groups. The scale of the colour maps of the first three age groups were chosen differently from the other groups to show the best variation in thickness. Notable is that the thickness map of second age group deviates from the thickness growth pattern. No abnormalities in the scans were found and this variance was assigned to influence of variation within the population due to small population groups.

*Table 1 Number of scans available per age groups*

Age group (months)	Number of scans
<b>0-1</b>	6
<b>2-4</b>	8
<b>5-7</b>	3
<b>8-10</b>	8
<b>11-14</b>	16
<b>15-18</b>	11
<b>19-24</b>	13
<b>25-36</b>	9
<b>37-48</b>	11

### 3.4 DISCUSSION

Because no appropriate age-specific 3D normative data was available till now, reconstruction techniques were originated from experiences, subjective decision making (what is beauty?) and common practice. This study provides objective 3D normative data that can be used in the creation of objective pre-operative reconstruction plans. The planning technique will now be able to create a plan that will reconstruct the patient's skull in such that its shape maximally corresponds with the age-specific reference skull. Hereby, the plan will not be based on the opinion of the surgical team, but on the normalised shape and volume of the healthy population of the same age group as the patient.

This study has also set a step in the development of objective evaluation tools for open cranial vault reconstructions. At this moment the surgical results are only evaluated based on 2D measurements or the opinions of the surgical team and the parents. These evaluation parameters are not sufficient and as long as the surgical team and parents are content with the surgical results, there will be no reason to reconsider the used surgical techniques. This study provides objective age-specific 3D normative data. Suggestions for evaluation parameters are intracranial volume and shape correspondence between the patient's skull and the age-appropriate normalised skull. By objective evaluation, superior reconstruction techniques can be selected and standardisation in the treatment of craniosynostosis can be accomplished. A drawback of the use of the normalised skulls created in this study is that post-operative CT-scans of the patient are required, which increases the received radiation dosage of the

patient. Therefore, research at the Radboudumc also focusses on the development of post-operative evaluation tools based on 3D stereophotogrammetry.

The database with CT-scans of healthy skulls still needs expanding. The goal is to create a database with at least 20 scans per non-clustered age-group. At this moment all cranial CT-scans that are present in the Radboudumc medical database and were obtained within the period of January 2001 till December 2015 were screened. Because the Radboudumc medical database appears not to be sufficient, negotiations with other medical centres are ongoing. The aim is to collaborate with medical centres from several West-European countries in order to create reliable 3D normative data that represent the West-European population.

The thickness maps were created by computing the difference between the hit with the inner layer and with the outer layer of the skull for each separate ray. Figure 5 suggests that the rays are not casted perpendicular to the surface of the skull. As result a bias in the thickness computation exists because the rays do not follow the shortest route through the skulls. This bias increases with the thickness of the skull. Because the skull thickness in infants is only a few millimetres, it is expected that this bias was minimal.

### 3.5 CONCLUSION

This study provided 3D normalised skull models for different age-groups, based on a small CT-database of the Dutch population.

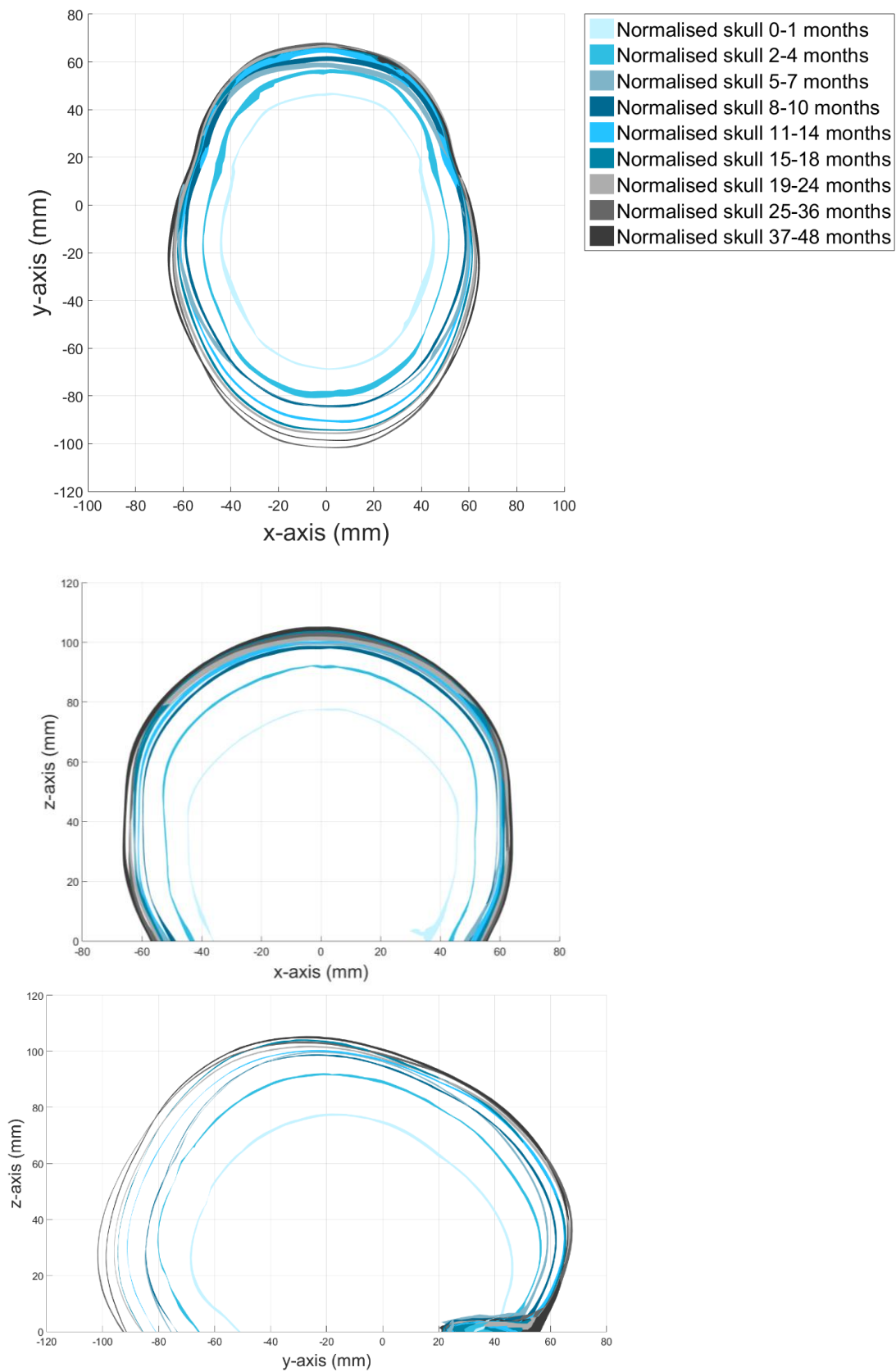


Figure 6 Normalised skulls: (Top) cross-section in z-direction, (Middle) cross section in y-direction, (Bottom) cross-section in x-direction



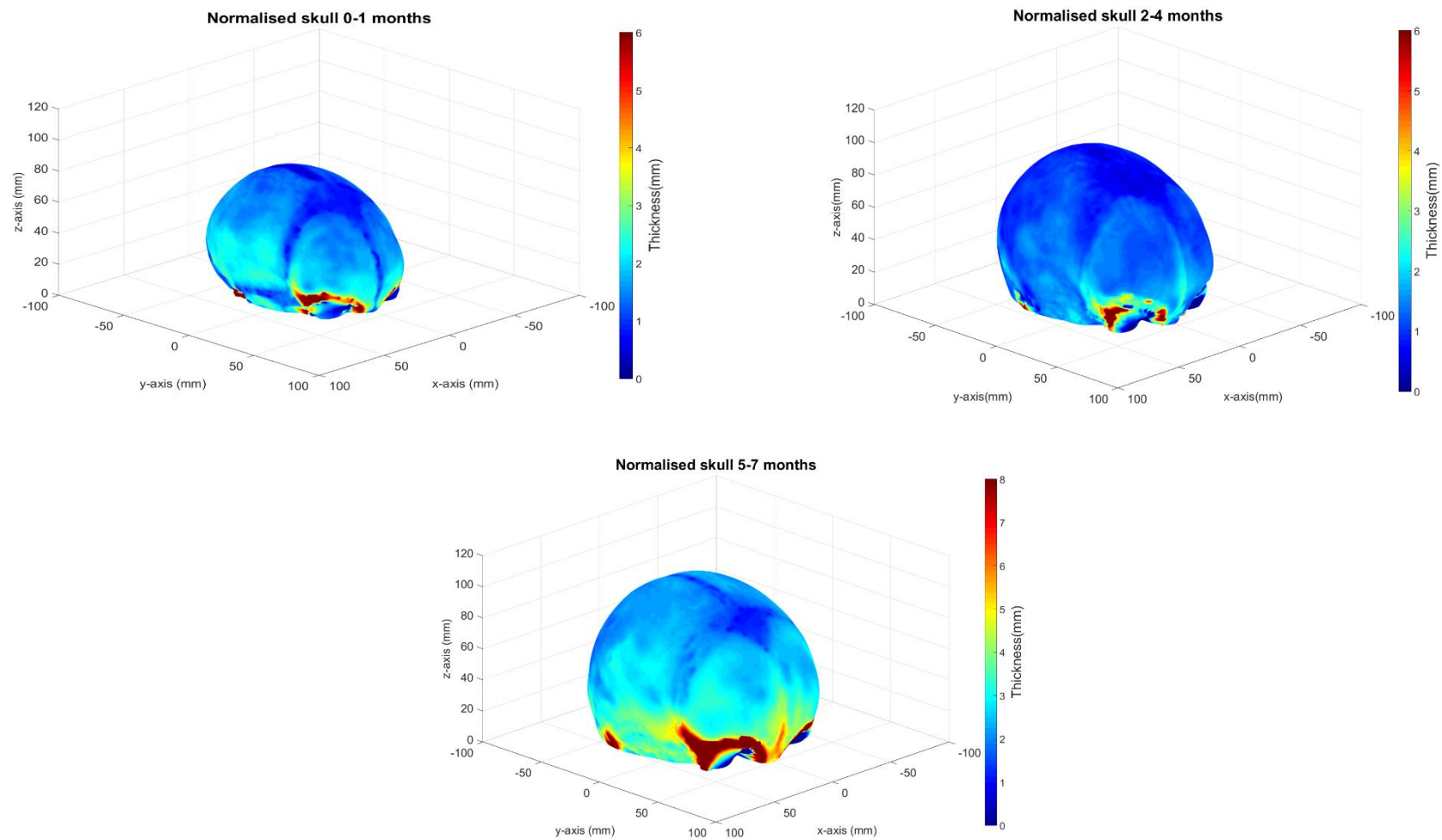


Figure 7 Thickness maps of normalised skulls; age groups 0-1, 2-4, and 5-7 months

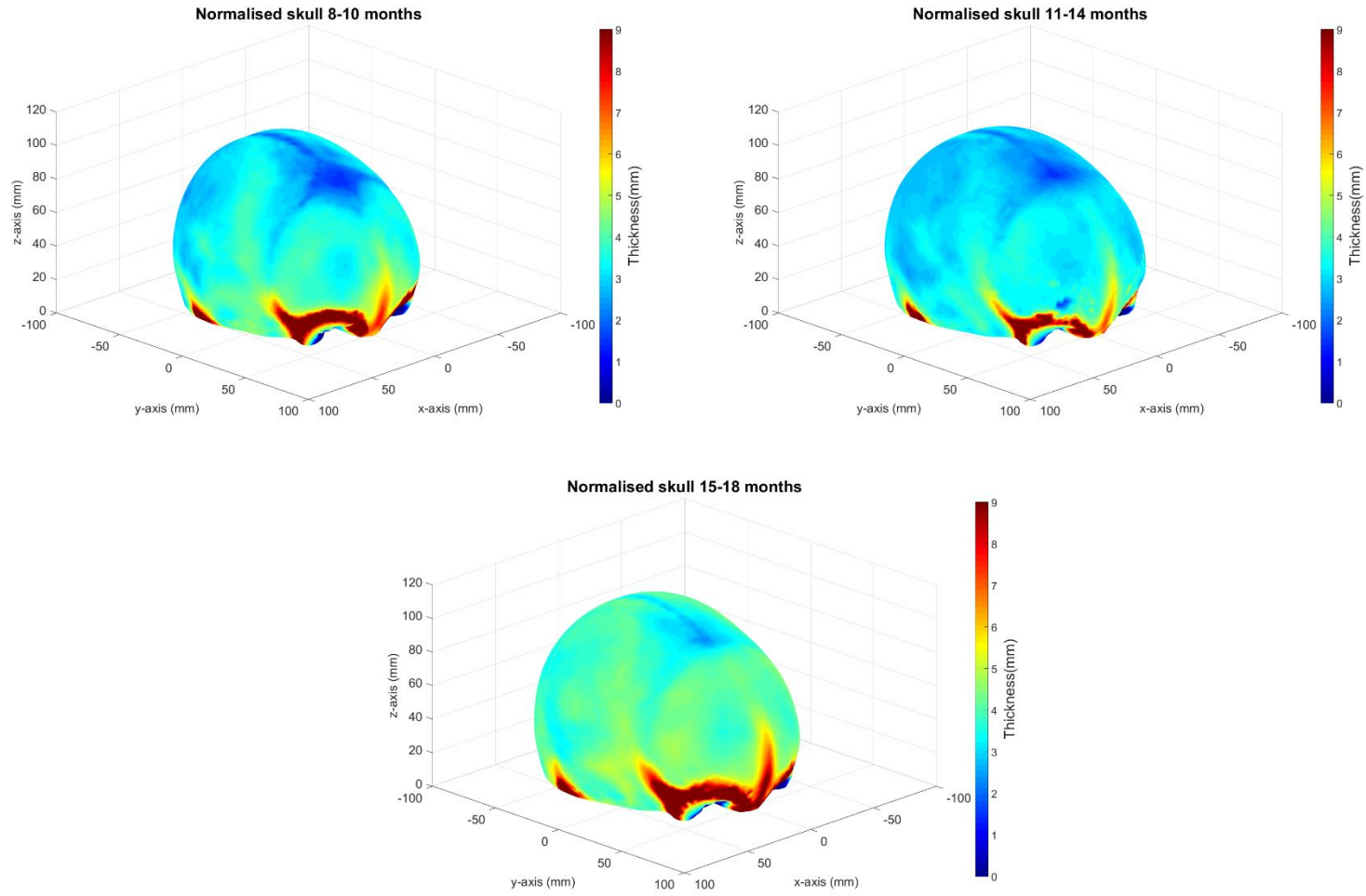


Figure 8 Thickness maps of normalised skulls; age groups 8-10, 11-14, and 15-18 months

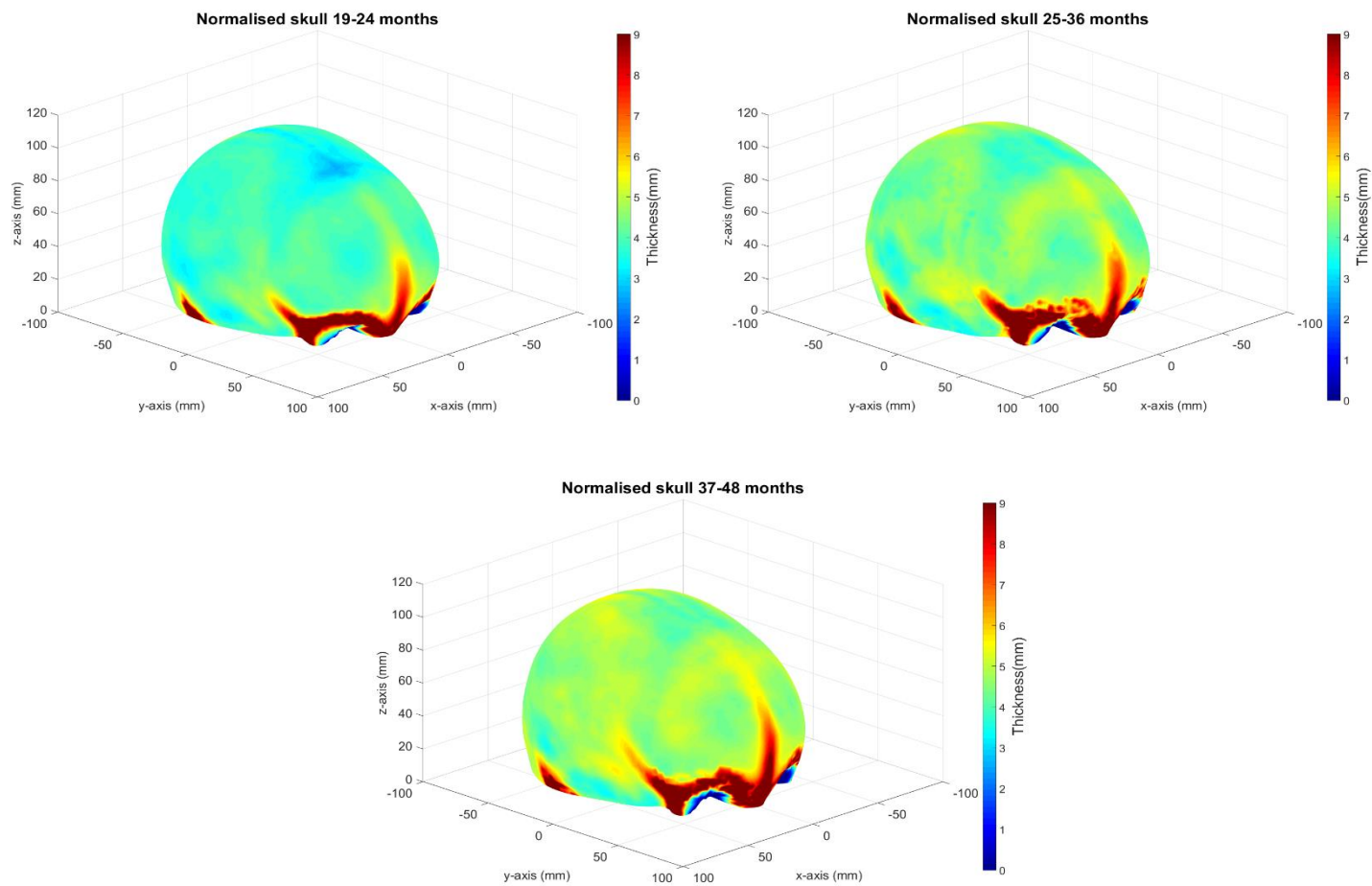


Figure 9 Thickness maps of normalised skulls; age groups 19-24, 25-36, and 37-48 months



## 4 SHAPE COMPARISON

### 4.1 INTRODUCTION

Current research focuses on the development of pre-operative planning techniques for open cranial vault reconstructions[20], [21], [22]. However, to our knowledge no solution has been found for the variance in results caused by subjective decision making yet. The developed techniques for the creation of a pre-operative plan still require interactive user-input, remain time consuming, complicated and labour-intensive, and follow an iterative process. Often several sessions are needed in which the medical technician discusses the feasibility of the created plan with the surgeon. Hereby, the pre-operative plans are based on the opinion of the surgeon of what is beauty [20]–[23].

In order to create a fully objective planning technique, an algorithm will be developed that is able to compare the patient’s skull with an age-appropriate reference model. The new cranial shape is obtained by relocating osseous panels and is therefore dependent on the shape of these panels. In other words, when the created panels do not have the optimal shape it is impossible to obtain an optimal cranial vault shape. Preliminary work already created normalised skulls that can be used as reference models. This chapter focusses on an algorithm for the local shape comparison of the patients’ skulls with a reference model, based on a method described by Gatzke et al.[33]. The algorithm enables the selection of a region on a reference skull and finds the region on the patient’s skull that maximally corresponds in shape with the selected reference shape. This chapter will first elaborate on the surface fitting method and the curvature computation for mesh data, then describe the shape comparison algorithm and finally describe the analysis in which the feasibility of the shape comparison was demonstrated.

### 4.2 MATERIALS AND METHODS

#### 4.2.1 Materials

The shape comparison algorithm was implemented in MATLAB 2016a®. Due to the large amount of memory use and computation time, OpenCL was used for GPU integration. The 3D test objects were created with 3ds Max 2016®. CT-scans were obtained from the hospital database and were provided as DICOM files.

#### 4.2.2 Methods

##### 4.2.2.1 Quadric surface fitting

Because curvature is an intrinsic property of a surface and therefore gives direct information about the shape of a surface, the shape comparison was based on curvature (for the definition of curvature, see supplement 0). All skulls and test objects were represented as mesh data, based on the method described in chapter 2. The curvature was computed for each vertex of the mesh data, based on the quadric surface fitting method described by Hoppe et al[34] and Garland et al.[35]. The quadric surface fitting method obtained a quadric surface that approximated the surface of the mesh around a specific vertex (from now on called vertex quadric). A quadric surface can be defined by a  $4 \times 4$  symmetrical matrix that stores 10 coefficients:

$$Q_v(\mathbf{v}) = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ q_{14} & q_{24} & q_{34} & q_{44} \end{bmatrix} \quad (1)$$

The goal of the surface fitting method was to find the 10 coefficients that solve the least square fitting problem for vertex  $\mathbf{v}$ :

$$\mathbf{v}^T Q_v \mathbf{v} = 0 \quad (2)$$

According to Hoppe et al[34] and Garland et al.[35], to find the 10 coefficients of this quadric surface, the quadrics of the faces within a specified radius  $R_N$  from the vertex, weighted by their face area, are summed:

$$Q_v(\mathbf{v}) = \sum_{i=1}^{nrF} \text{area}(F_i) \cdot Q^{F_i} \quad (3)$$

$Q_v$  is the vertex quadric,  $F_i$  is a neighbouring triangular face build up from three vertices  $\mathbf{v}_1$ ,  $\mathbf{v}_2$  and  $\mathbf{v}_3$ , and  $Q^{F_i}$  the quadric surface that approximates the surface of  $F_i$  (from now on called face quadric).  $nrF$  is the number of faces that are present within a specified neighbourhood on which the vertex quadric is based.

Based on equation (3), the surface fitting method proceeded according to *Pseudocode 2*. The individual steps are explained in the following section of this chapter.

### *Pseudocode 2*

```

For each face:
    Compute face quadric
    Compute face area
    Multiply face quadric by face area
For each vertex:
    Find faceIds within search radius  $R_N$  from vertex
    Obtain face quadrics of faceIds
    Sum face quadrics
  
```

### *Face quadric*

A face quadric was defined by the coefficients of the general plane equation in which the face was present (for mathematical explanation see supplement 7.5):

$$ax + by + cz + d = 0 \quad (4)$$

The plane coefficients  $a$ ,  $b$  and  $c$  were given by the unit face normal:  $\mathbf{n}_{F_i} = (a, b, c)$ . The face normal and coefficient  $d$  were computed with the following equations:

$$\mathbf{n}_{F_i} = \frac{(\mathbf{v}_1 - \mathbf{v}_2) \times (\mathbf{v}_1 - \mathbf{v}_3)}{\|(\mathbf{v}_1 - \mathbf{v}_2) \times (\mathbf{v}_1 - \mathbf{v}_3)\|}$$

$$d = -\mathbf{n}^T \cdot \mathbf{p}$$

A face quadric was then defined as:

$$Q^{F_i} = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix}$$

### Face area

The area of a face was defined as half of the length of the face normal vector (note: no unit vector):

$$area(F_i) = \frac{1}{2}((\mathbf{v}_1 - \mathbf{v}_2) \times (\mathbf{v}_1 - \mathbf{v}_3))$$

### Multiplying quadrics by a scalar

The face quadrics were multiplied by the face areas according to the following rule:

$$\gamma Q = \gamma \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ q_{14} & q_{24} & q_{34} & q_{44} \end{bmatrix}$$

### Neighbourhood search

A neighbourhood search algorithm was implemented that allowed the search for faces within a specified radius  $R_N$  from a vertex. To enhance the process time of this algorithm, the neighbourhood search was based on clustering. Figure 10 illustrates the clustering for an icosphere mesh. First, a bounding box was created including all the vertices (Figure 10 (Left)). This box was divided in a grid based on  $R_N$  and each grid element was given a cluster identity. Each vertex was allocated to the cluster identity of the grid in which the vertex was present. This obtained a list with information about which vertex is present in which cluster. Figure 10 (Right) shows the mesh in which the vertices are colour coded based on the allocated cluster identities. This list was used to find all faces within  $R_N$  from a specific vertex. Hereby, it was only necessary to check whether vertices were within  $R_N$  distance from the original vertex, for neighbours within adjacent clusters and the cluster in which the respective vertex was present. The pseudocode of the clustering and neighbourhood search are given in *Pseudocode 3* and *Pseudocode 4*, respectively. Because the  $R_N$  has important effect on scaling, the appropriate  $R_N$  was not known beforehand. Chapter 4.2.2.4 will further elaborate on the selection of the appropriate  $R_N$ .

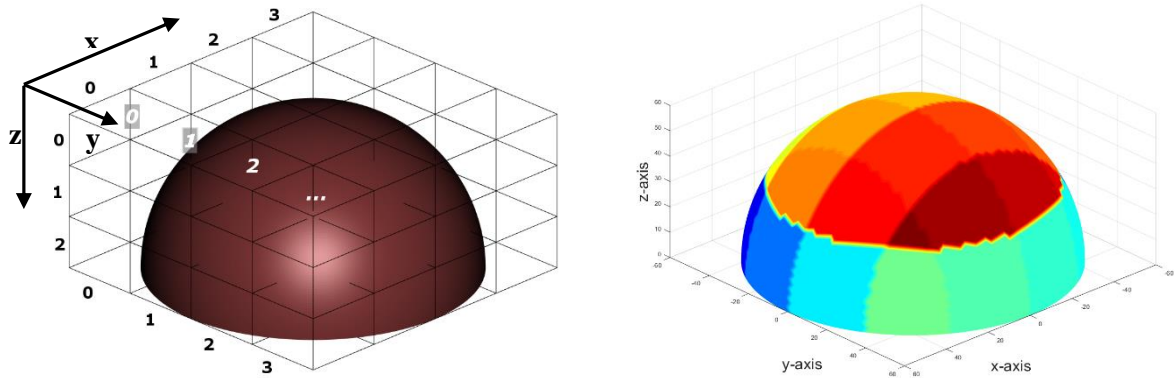


Figure 10 (Left) Bounding box and grid with indexation (Right) Vertices colour coded by cluster identity with  $R_N=30$  mm

**Pseudocode 3**

**Determine** boundaries of bounding box:  
**Find** minimal  $x, y, z$  - values of vertices  
 $(minGrid_x, minGrid_y, minGrid_z)$   
**Enlarge** bounding box by 1 % to make sure that all vertices are included  
**Save**  $x, y, z$  - size bounding box  $(Size_x, Size_y, Size_z)$   
**Divide** bounding box in grid with size  $\frac{Size_x}{R} \times \frac{Size_y}{R} \times \frac{Size_z}{R}$   
**Compute** number of boxes in grid in  $x, y, z$  - direction  
 $(nrbox_x, nrbox_y, nrbox_z)$   
**Create** vertex list with the corresponding clusterId (*VertexClusterList*):  
**Compute** box number in grid in  $x, y, z$  - direction

$$Blocknr_x = \text{floor} \left( nrbox_x \cdot \left( \frac{Vx - minGrid_x}{Size_x} \right) \right)$$

$$Blocknr_y = \text{floor} \left( nrbox_y \cdot \left( \frac{Vy - minGrid_y}{Size_y} \right) \right)$$

$$Blocknr_z = \text{floor} \left( nrbox_z \cdot \left( \frac{Vz - minGrid_z}{Size_z} \right) \right)$$

**Assign** clusterId to vertex  
 $ClusterId = Blocknr_x \cdot nrbox_x \cdot nrbox_z + Blocknr_y + Blocknr_z \cdot nrbox_z$

**Pseudocode 4**

**Obtain** *clusterId* from *VertexClusterList*  
**Determine** clusters adjacent to *clusterId*  
**Get** all *vertexIds* that are present within *clusterId* and adjacent clusters  
**Find** all *faceIds* that contribute to *vertexIds* within the clusters  
**For** each *faceId*:  
**Get** *vertexIds* of *faceId*  
**Walk** through *vertexIds*  
**Check** if *vertexId* is present within the clusters  
**If** all *vertexIds* of *faceId* are present within clusters, add face to *neighbourhoodlist*

**Quadric summation**

To solve equation (1), the face quadrics within a specified neighbourhood radius from the respective vertex were summed. The methods of the previous sections enabled to compute the face quadrics and find the faces (and therefore the face quadrics) contributing to a vertex neighbourhood. The summation of the face quadrics for each vertex was performed according to the following rule:

$$\sum_{i=1}^k Q = Q_i + Q_{i+1} + \dots + Q_k$$



#### 4.2.2.2 Quadric curvature

For each vertex the vertex quadric was obtained and the curvature was computed. Next to the  $4 \times 4$  matrix given in equation (1), quadrics can be represented by the implicit function. This implicit function was deduced from the least square problem given in equation (2):

$$F(x, y, z) = q_{11}x^2 + 2q_{12}xy + 2q_{13}xz + 2q_{14}x + q_{22}y^2 + 2q_{23}yz + 2q_{24}y + q_{33}z^2 + 2q_{34}z + q_{44} \quad (5)$$

Hereby,  $x, y$  and  $z$  are the coordinates of the vertex for which the vertex quadric was obtained. Goldman[36] describes the curvature formulas for implicit functions. Supplement 7.6 summarises these formulas. According to these formulas, the curvature formulas for quadrics were defined as followed:

$$\begin{aligned} \frac{\partial^2 F}{\partial x^2} &= 2q_{11} \\ \frac{\partial F}{\partial x} &= 2(q_{11}x + q_{12}y + q_{13}z + q_{14}) & \frac{\partial^2 F}{\partial y \partial x} &= \frac{\partial^2 F}{\partial x \partial y} = 2q_{12} \\ \frac{\partial F}{\partial y} &= 2(q_{12}x + q_{22}y + q_{23}z + q_{24}) & \frac{\partial^2 F}{\partial z \partial x} &= \frac{\partial^2 F}{\partial x \partial z} = 2q_{13} \\ \frac{\partial F}{\partial z} &= 2(q_{13}x + q_{23}y + q_{33}z + q_{34}) & \frac{\partial^2 F}{\partial z \partial y} &= \frac{\partial^2 F}{\partial y \partial z} = 2q_{23} \\ & & \frac{\partial^2 F}{\partial z^2} &= 2q_{33} \end{aligned}$$

$$\nabla F = 2([q_{11}x + q_{12}y + q_{13}z + q_{14}, \quad q_{12}x + q_{22}y + q_{23}z + q_{24}, \quad q_{13}x + q_{23}y + q_{33}z + q_{34}])$$

$$H(F) = \begin{bmatrix} 2q_{11} & 2q_{12} & 2q_{13} \\ 2q_{12} & 2q_{22} & 2q_{23} \\ 2q_{13} & 2q_{23} & 2q_{33} \end{bmatrix}$$

$$H^*(F) = \begin{bmatrix} 4q_{22}q_{33} - 4(q_{23})^2 & 4q_{23}q_{13} - 4q_{12}q_{33} & 4q_{12}q_{23} - 4q_{22}q_{13} \\ 4q_{13}q_{23} - 4q_{12}q_{33} & 4q_{11}q_{33} - 4(q_{13})^2 & 4q_{12}q_{13} - 4q_{11}q_{23} \\ 4q_{12}q_{23} - 4q_{13}q_{22} & 4q_{12}q_{13} - 4q_{11}q_{23} & 4q_{11}q_{22} - 4(q_{12})^2 \end{bmatrix}$$

$$\text{Trace}(H) = 2q_{11} + 2q_{22} + 2q_{33}$$

$$K_G = \frac{\nabla F \cdot H^*(F) \cdot \nabla F^T}{2\|\nabla F\|^4}$$

$$K_M = \frac{\nabla F \cdot H(F) \cdot \nabla F^T - |\nabla F|^2 \text{Trace}(H)}{2\|\nabla F\|^3}$$

$$k_1, k_2 = K_M \pm \sqrt{K_M^2 - K_G}$$

$K_G$  is the Gaussian curvature,  $K_M$  the mean curvature, and  $k_1$  and  $k_2$  are the principal curvatures.

### 4.2.2.3 Shape comparison

The shape comparison algorithm creates curvature maps that can be used to quantify the shape correspondence of two regions. Because the Gaussian curvature has the most geometrical meaning, the curvature maps were based on the Gaussian curvature values of the vertices.

The curvature maps were built up from regions defined by  $i$ -ring neighbourhoods. An 1-ring neighbourhood of a vertex was defined as the set of vertices that are connected to the vertex with only a single edge. The valence of the mesh structures was in most cases 6. Some exceptions existed where the valence was 5. Figure 11 shows a 3-ring neighbourhood of a vertex with valence 6. Exploring Table 2 gave the sequences for the total number of vertices and faces and per  $i$ -ring neighbourhood for a valence of 6. These sequences were used for the ring-based neighbourhood search given in *Pseudocode 5* on page 28:

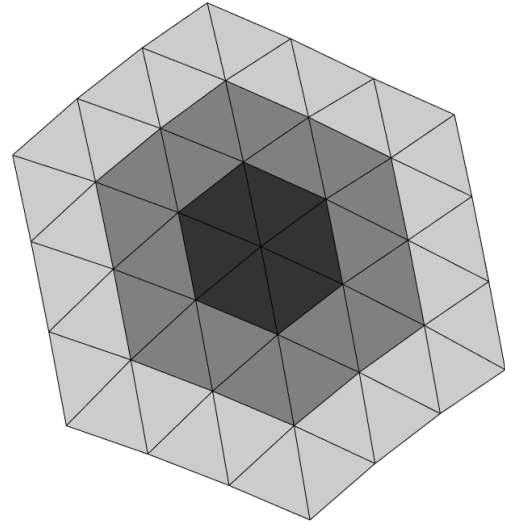


Figure 11 3-Ring neighbourhood

$$nrVperRing = n \cdot 6$$

$$totnrofV = 3 \cdot n \cdot (n + 1)$$

$$nrFperRing = 6 + (n - 1) * 12$$

$$totnrofF = 6 \cdot n^2$$

Table 2 Total number of vertices and faces and per ring for a  $N$ -neighbourhood

$i$	Number of vertices per ring ( $nrVperRing$ )	Total number of vertices ( $totnrofV$ )	Number of faces per ring ( $nrFperRing$ )	Total number of faces ( $totnrofF$ )
1	6	6	6	6
2	12	18	18	24
3	18	30	30	54
4	24	54	42	96
...	...	...	...	...

A curvature map was defined as a piecewise linear function that describes the change of the curvature over the different neighbouring rings  $R_i$ . For each vertex of the mesh data, a curvature map was computed based on an 8-ring neighbourhood. Per ring, the mean of the Gaussian curvature values was computed:

$$g(i) = \frac{1}{N_i} \sum_{j \in R_i} \kappa(\mathbf{v}_j)$$

$N_i$  is the number of vertices in ring  $R_i$ , and  $\kappa(\mathbf{v}_j)$  are the curvature values of the vertices within the ring. A logarithmic filter was used to suppress high curvature variations:

$$h(x) = \sin(x) \log(1 + |x|) \quad (6)$$

$$f^i = h_1 \circ g(\kappa)$$

$f^i$  was scaled by the distance of the ring to the original vertex, which resulted in the  $i^{th}$  value of the curvature map  $\kappa_{map}$ :

$$r_i = \sqrt{\frac{A_i}{\pi}}$$

$$\kappa_{map_i} = f^i \cdot r_i$$

$A_i$  is the area of the  $i$ -ring, which was computed by summing the areas of all faces within the ring. The areas of all faces were already computed in chapter 4.2.2.1.

For the shape comparison of two regions, the  $L_1$  difference between the curvature maps of both regions ( $\kappa_{mapA}$  and  $\kappa_{mapB}$ ) was computed:

$$S = \sum_{i=1}^{N_i} \left( \left| \kappa_{mapA_i} - \kappa_{mapB_i} \right| \right) \quad (7)$$

$S$  is the signature that represents the difference in shape. When  $\kappa_{mapA}$  and  $\kappa_{mapB}$  are equal,  $S$  will be zero and with increasing difference between the shape of the regions,  $S$  will increase as well. In conclusion, to find the region on a mesh that maximally corresponded with a selected reference region, the vertex for which the curvature map results in the minimal value of  $S$  was found. The ring neighbourhood of this vertex was then the best fit for the reference region.

#### 4.2.2.4 Analysis

To show the feasibility of the shape analysis algorithm, several analyses were performed. First, to verify the formulas for quadric curvature given in chapter 4.2.2.2, the Gaussian, mean and principal curvature were computed for quadrics describing a plane, sphere and saddle. The quadric for the plane was obtained with the method described in chapter 4.2.2.1. The quadrics for the sphere and saddle were defined by standard quadric forms. Because the plane is a flat object, all curvature values were expected to be zero. Also, according to the definition of the Gaussian and mean curvature given in supplement 0,  $\kappa_G = 0$  and  $\kappa_G < 0$  for a symmetrical saddle shape. Due to the opposite symmetrical shape of the saddle, the principal curvatures were expected to be:  $\kappa_1 = -\kappa_2$ . Moreover, Goldman [36] shows that for a sphere:  $\kappa_M = \frac{1}{R_s}$ ,  $\kappa_G = \frac{1}{R_s^2}$  and  $\kappa_1 = \kappa_2$ . Because the curvatures for the different forms were known, it was possible to verify the results.

Curvature based on quadric surface fitting was computed for all vertices of a hemi-icosphere, full icosphere and an 11 months old skull. The radius of both icospheres was 60 mm, which was based on the width of the normative skull from the 11-14 months age group. Because the curvatures for the spheres were known ( $\kappa_G = \frac{1}{R_s^2} = \frac{1}{60^2} = 2.8 \cdot 10^{-4} m^{-2}$ ), it was again possible to verify the results. For this, the mean values and standard deviations for the different neighbourhood sizes were computed. Furthermore, it was expected that if the quadric surface fitting was based on a small neighbourhood size, high variety of curvature, for example small ridges, would be detected on the skull surface. In contrast, the curvature estimation based on a greater neighbourhood size would only detect small variety in curvature of the surface and small ridges would be missed. To be able to choose the most appropriate neighbourhood size, curvature was computed based on  $R_N = 10, 15, 20, 25, 30$  and  $35 mm$ . The appropriate radius was selected according to the detail discrimination, which was used in further analysis.

To verify the shape comparison algorithm, the method was performed for three test cases. First, shape comparison was applied on a test object. The test object consisted of a sphere with 5 spherical bulges

with different radii, a torus shaped bulge and a spherical hole (Figure 18 (*Top Left*)). The reference region was selected at the top of the bulge with the second last radius. The hole had the same radius as this bulge. Therefore, the shape of the inside surface of the hole was similar to the outside shape of the bulge. The torus and hole were included in the test object in order to show that the curvature maps discriminate different shapes. The shape of the test object was compared with the shape of second test object that was a 45° anti-clockwise rotated version of the original. The region on the rotated test object that showed the maximal shape correspondence with the reference region was determined. Because the compared objects were equivalent, it possible to verify whether the right region was identified by the algorithm. Secondly, the shape comparison was applied within the normalised skull from the 11-14 months age group. Because a healthy skull is (almost) symmetrical, it was expected that a shape on one side of the skull could also be found on the other side of the skull. Therefore, a distinguishable region, the right temple, was selected as reference region and the algorithm was tested for the ability to identify the similar shape on the left temple. Finally, the algorithm was tested for an 11 month old trigonocephaly case. Because for trigonocephaly patients the forehead is reconstructed, a reference region on the right side of the forehead of the normalised skull was selected. The region on the trigonocephaly skull for which the shape maximally corresponded with the shape of the reference region was identified. Because only the shape of the outer surface of the skull is relevant for the aesthetics of the patient, the shape comparison was applied on only the outer layer of the skulls.

#### 4.2.2.5 Extended icosphere

Because the results showed inaccuracies in the curvature estimation for vertices near borders, the hemi-icosphere in the ray casting method described in chapter 2, was extended in the  $z$  -direction (see Figure 12). As a result, the ray casting obtained extra geometrical information of at least  $-\frac{1}{2}R_N < z < 0$  for all objects and skulls. Hereby, for vertices with  $0 < v_z < \frac{1}{2}R_N$  the curvature estimation was now based on full neighbourhoods. Because not all skulls contained data for  $z < 0$ , the extrapolation method described in chapter 2 was used to fill in missing data points. The extended icosphere was preferred over the full icosphere, because of less memory storage. The extra data was only used for the goal to increase the accuracy of the curvature computation for vertices near borders. The shape comparison algorithm was only applied on regions of vertices with  $z > 0$ .

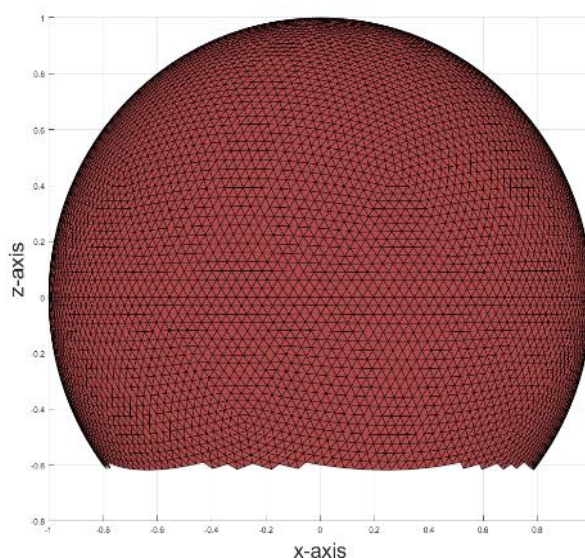


Figure 12 Extended unit hemi-icosphere

*Pseudocode 5*

```

Allocate vertex neighbour list (VNlist)
Allocate face neighbour list (FNlist)

Walk through all rings
  For the first ring
    Walk through all vertices
      Find adjacent faces (Find faceId containing VertexId)
      For each adjacent faceId
        Walk through the vertices of this face
          Place vertex ids in a temporary array
        Remove repetitive vertexids and original
          vertexId
        Place adjacent vertex ids in the VNList (1-6)
          valence 5 vertices contain a 'zero'
          neighbour
        Place adjacent face ids in FNList (1 to 6),
          valence 5 vertices contain a zero
          neighbour
    For the 2nd till number of rings
      Walk through all vertices
        Get neighbours from VNList of the previous found ring:
          VNList(VertexId, (totNrV - prevFoundNrV)+1)
        For each neighbouring vertex, if not zero
          Get the first ring neighbourhood vertices and
            place in temp vertex neighbourhoodlist
          Get the first ring neighbourhood faces and place
            in temp face neighbourhoodlist
          Remove repetitive vertexIds and faceIds, and
            zeros
          Remove all neighbouring vertices and faces that
            were already found in previous rings
          Remove original vertex
          Add found neighbours to VNList and FNList

```

## 4.3 RESULTS

### 4.3.1 Quadric curvature

#### 4.3.1.1 Plane

The plane was defined by points  $v_1 = (0,1,0)$ ,  $v_2 = (0,0,1)$  and  $v_3 = (1,0,0)$ . According to the method described in chapter 4.2.2.1, the parameters of the quadric that approximates the surface of the plane are computed as followed:

$$\begin{aligned} \mathbf{w}_1 &= \mathbf{v}_1 - \mathbf{v}_2 = (0,1,-1) \\ \mathbf{w}_2 &= \mathbf{v}_1 - \mathbf{v}_3 = (-1,1,0) \\ \mathbf{n}_{Fi} &= \frac{\mathbf{w}_1 \times \mathbf{w}_2}{\|\mathbf{w}_1 \times \mathbf{w}_2\|} = \frac{(1,1,1)}{1.73} = (0.58 \quad 0.58 \quad 0.58) \\ a &= 0.58, b = 0.58, c = 0.58 \\ d &= -(a \cdot v_{1x} + b \cdot v_{1y} + c \cdot v_{1z}) = -0.58 \\ \text{area}(f) &= \frac{\|\mathbf{w}_1 \times \mathbf{w}_2\|}{2} = 0.50 \end{aligned}$$

And thus, the plane and quadric are defined as:

$$ax + by + cz + d = 0.58x + 0.58y + 0.58z - 0.58 = 0$$

$$\begin{aligned} Q_v(\mathbf{v}) &= \sum_{f \ni v} \text{area}(f) \cdot Q^f(\mathbf{p}) = 0.50 \cdot \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix} \\ &= 0.50 \cdot \begin{bmatrix} 0.33 & 0.33 & 0.33 & -0.33 \\ 0.33 & 0.33 & 0.33 & -0.33 \\ 0.33 & 0.33 & 0.33 & -0.33 \\ -0.33 & -0.33 & -0.33 & 0.33 \end{bmatrix} = \begin{bmatrix} 0.17 & 0.17 & 0.17 & -0.17 \\ 0.17 & 0.17 & 0.17 & -0.17 \\ 0.17 & 0.17 & 0.17 & -0.17 \\ -0.17 & -0.17 & -0.17 & 0.17 \end{bmatrix} \end{aligned}$$

To visualise the quadric surface, all solutions for equation (2) were plotted with interval  $[-5,5]$  and a step size of 0.1 for  $x, y$  and  $z$ . Figure 13 shows that the quadric surface is congruent to the original plane.

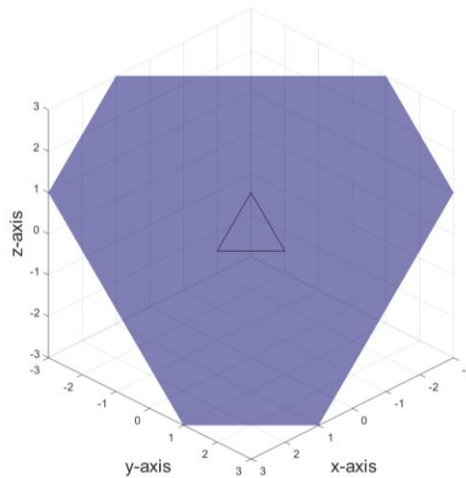


Figure 13 Quadric approximating the surface of the plane defined by  $P1 = (0,1,0)$ ,  $P2 = (0,0,1)$  and  $P3 = (1,0,0)$

Given equation (5), the implicit function of the quadric is:

$$F(x, y, z) = 0.17x^2 + 0.33xy + 0.33xz - 0.33x + 0.17y^2 + \\ + 0.33yz - 0.33y + 0.17z^2 - 0.17z + 0.17 = 0$$

The curvature was computed for  $P = (0.5, 0.5, 0.5)$ , which is a point on the surface of the plane. The first and second derivatives are:

$$\frac{\partial F}{\partial x} = \frac{\partial F}{\partial y} = \frac{\partial F}{\partial z} = 2(0.17x + 0.17y + 0.17z - 0.17) = 0.17$$

$$\frac{\partial^2 F}{\partial x^2} = \frac{\partial^2 F}{\partial y \partial x} = \frac{\partial^2 F}{\partial z \partial x} = \frac{\partial^2 F}{\partial x \partial y} = \frac{\partial^2 F}{\partial y^2} = \frac{\partial^2 F}{\partial z \partial y} = \frac{\partial^2 F}{\partial x \partial z} = \frac{\partial^2 F}{\partial y \partial z} = \frac{\partial^2 F}{\partial z^2} = 2 \cdot 0.17 = 0.33$$

$$\nabla F = (0.17, 0.17, 0.17)$$

$$H(F) = \begin{bmatrix} 0.33 & 0.33 & 0.33 \\ 0.33 & 0.33 & 0.33 \\ 0.33 & 0.33 & 0.33 \end{bmatrix}$$

$$H^*(F) = \begin{bmatrix} 4(0.17)^2 - 4(0.17)^2 & 4(0.17)^2 - 4(0.17)^2 & 4(0.17)^2 - 4(0.17)^2 \\ 4(0.17)^2 - 4(0.17)^2 & 4(0.17)^2 - 4(0.17)^2 & 4(0.17)^2 - 4(0.17)^2 \\ 4(0.17)^2 - 4(0.17)^2 & 4(0.17)^2 - 4(0.17)^2 & 4(0.17)^2 - 4(0.17)^2 \end{bmatrix}$$

$$\text{Trace}(H) = 0.33 + 0.33 + 0.33 = 1.0$$

$$K_G = \frac{(0.17, 0.17, 0.17) \cdot \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0.17 \\ 0.17 \\ 0.17 \end{bmatrix}}{\sqrt{0.17^2 + 0.17^2 + 0.17^2}^4} = 0 \text{ m}^{-2}$$

$$K_M = \frac{(0.17, 0.17, 0.17) \cdot \begin{bmatrix} 0.33 & 0.33 & 0.33 \\ 0.33 & 0.33 & 0.33 \\ 0.33 & 0.33 & 0.33 \end{bmatrix} \cdot \begin{bmatrix} 0.17 \\ 0.17 \\ 0.17 \end{bmatrix} - 0.29^2 \cdot 1.0}{0.048} = 0 \text{ m}^{-1}$$

$$k_1, k_2 = 0 \pm \sqrt{0^2 - 0^2} = 0 \text{ m}^{-1}$$

In conclusion, the results confirm the expectations for the curvature values of a plane.

### 4.3.1.2 Sphere and saddle

Standard quadric forms were used to define a symmetrical saddle shaped and spherical surface. The quadric forms and results are shown in Figure 14. For the saddle, the parameters were  $k = 3, l = 3$  and  $m = 3$ . The radius  $R_s$  of the sphere was 3. The results in Figure 14(Right) meet the expectations discussed in chapter 4.2.2.4.

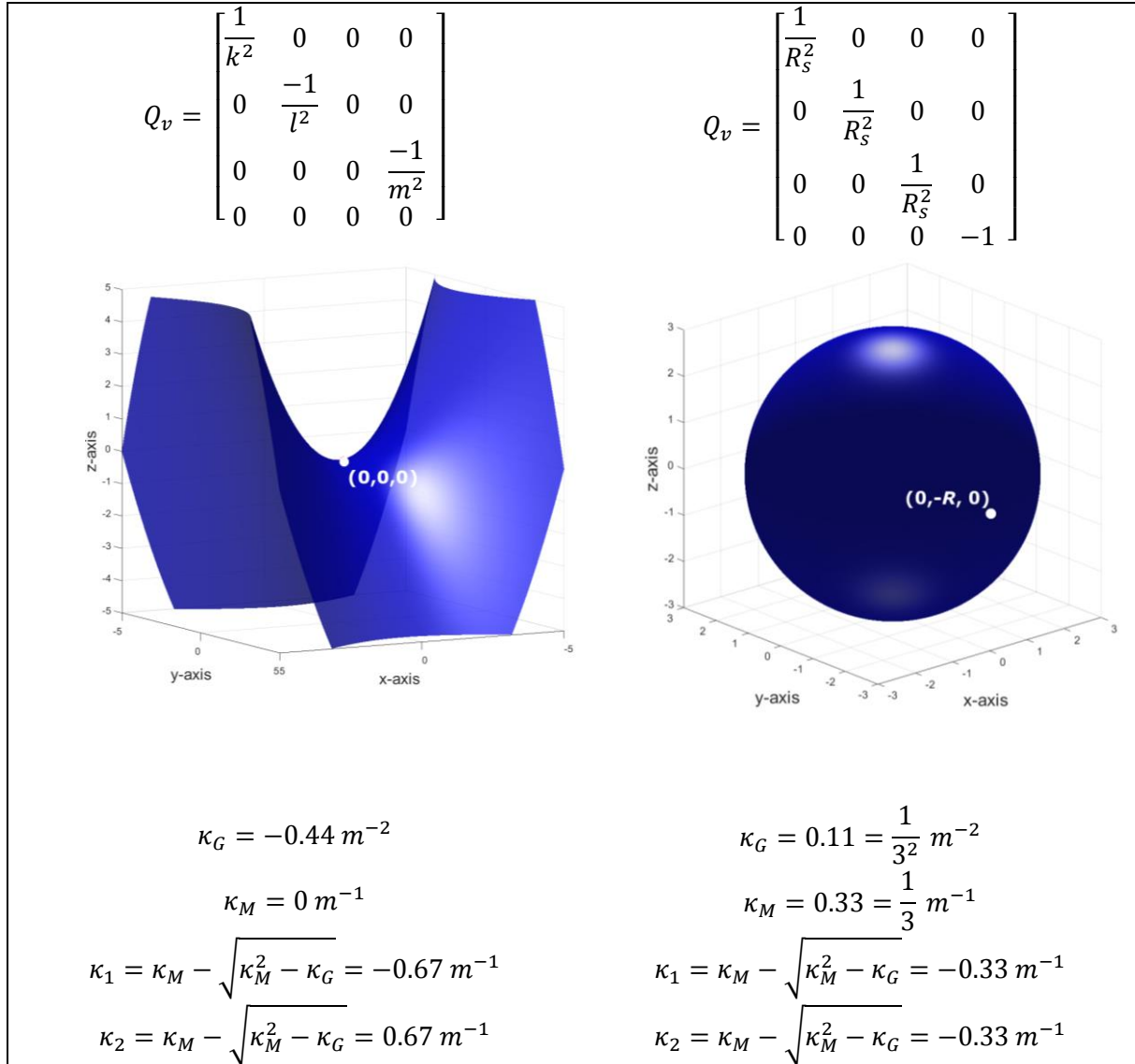


Figure 14 (Left) Quadric, visualisation and curvature for a saddle surface, (Right) Quadric, visualisation and curvature for a spherical surface

## 4.3.2 Quadric surface fitting

### 4.3.2.1 (Hemi-)Icosphere

Table 3, Figure 15 and Figure 16 show the results of the curvature estimations for the hemi- and full icosphere based on quadric surface fitting. Table 3 shows that for the hemi-icosphere (with border) the quadric surface fitting results in a small underestimation of the curvature, with a large standard deviation. The colour maps in Figure 15 are constant for all vertices of the hemi-icosphere, except for vertices at the borders. The figure shows that the width of this border increases when  $R_N$  increases and is approximately  $\frac{1}{2} \cdot R_N$ . Because the quadrics for the vertices with  $v_z < \frac{1}{2} R_N$  are based on less than 1 neighbourhood, with a minimum of  $\frac{1}{2}$  neighbourhood, it was expected that this results in asymmetrical



quadrics relative to the vertices. This results in a deviation in de curvature estimation, which could explain the large standard deviation. This was confirmed when the mean Gaussian curvature and standard deviation were computed for all vertices of the hemi-icosphere for which  $v_z > \frac{1}{2}R_N$  (*without border*). This resulted in a small over-estimation of the curvature, but smaller standard deviations. On account of these results the extended icosphere discussed in was chapter 4.2.2.5 introduced.

The results for the full icosphere in Table 3 show a small over-estimation of the curvature that increases when  $R_N$  increases. In comparison to the hemi-icosphere (*without border*), the standard deviations for the full icosphere are smaller. Figure 15 shows that the colour maps are constant for the entire surface of all spheres. Even though the results show some over-estimation of actual curvature, the results were found acceptable.

Table 3 Mean Gaussian curvature and standard deviations for different neighbourhood sizes of a hemi-icosphere (with and without border) and a full icosphere

$R_N$ (mm)	Hemi-icosphere <i>With border</i>		<i>Without border</i>		Full icosphere	
	$mean \kappa_G$ ( $m^{-2}$ )	$std \kappa_G$ ( $m^{-2}$ )	$mean \kappa_G$ ( $m^{-2}$ )	$std \kappa_G$ ( $m^{-2}$ )	$mean \kappa_G$ ( $m^{-2}$ )	$std \kappa_G$ ( $m^{-2}$ )
<b>10</b>	$2.75 \cdot 10^{-4}$	$3.51 \cdot 10^{-5}$	$2.85 \cdot 10^{-4}$	$8.01 \cdot 10^{-7}$	$2.85 \cdot 10^{-4}$	$7.86 \cdot 10^{-7}$
<b>15</b>	$2.70 \cdot 10^{-4}$	$4.22 \cdot 10^{-5}$	$2.85 \cdot 10^{-4}$	$8.01 \cdot 10^{-7}$	$2.86 \cdot 10^{-4}$	$5.69 \cdot 10^{-7}$
<b>20</b>	$2.68 \cdot 10^{-4}$	$4.85 \cdot 10^{-5}$	$2.85 \cdot 10^{-4}$	$8.11 \cdot 10^{-7}$	$2.90 \cdot 10^{-4}$	$4.12 \cdot 10^{-7}$
<b>25</b>	$2.68 \cdot 10^{-4}$	$5.44 \cdot 10^{-5}$	$2.85 \cdot 10^{-4}$	$8.28 \cdot 10^{-7}$	$2.97 \cdot 10^{-4}$	$3.29 \cdot 10^{-7}$
<b>30</b>	$2.69 \cdot 10^{-4}$	$6.04 \cdot 10^{-5}$	$2.85 \cdot 10^{-4}$	$8.54 \cdot 10^{-7}$	$3.05 \cdot 10^{-4}$	$2.47 \cdot 10^{-7}$
<b>35</b>	$2.72 \cdot 10^{-4}$	$6.65 \cdot 10^{-5}$	$2.85 \cdot 10^{-4}$	$8.71 \cdot 10^{-7}$	$3.16 \cdot 10^{-4}$	$3.68 \cdot 10^{-7}$

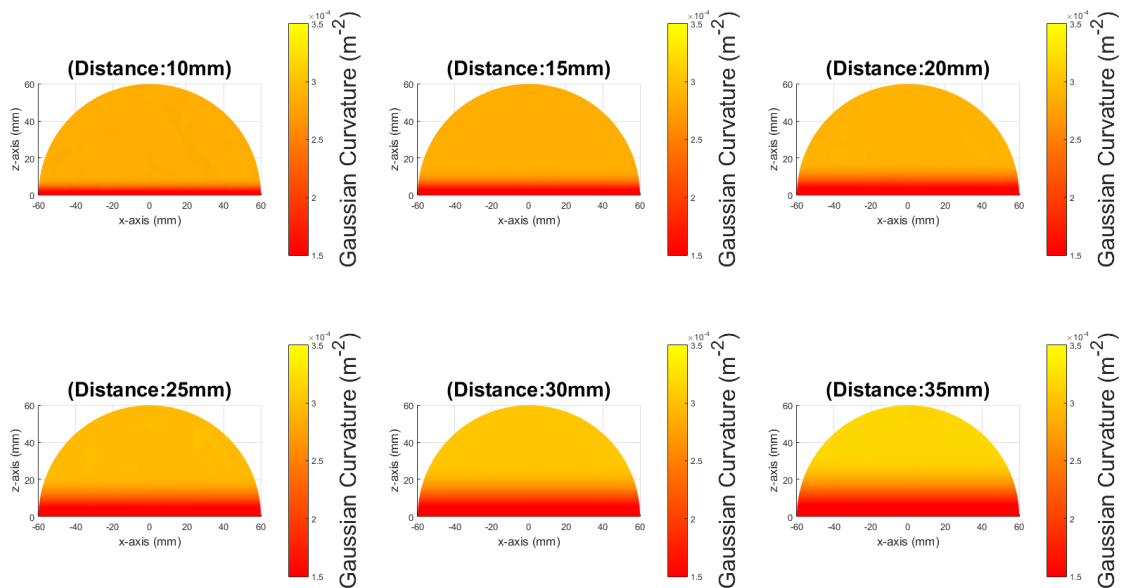


Figure 15 Curvature estimations for a hemi-icosphere ( $R_S = 60$  mm) with neighbourhood sizes:  $R_N = 10, 15, 20, 25, 30, 35$  mm

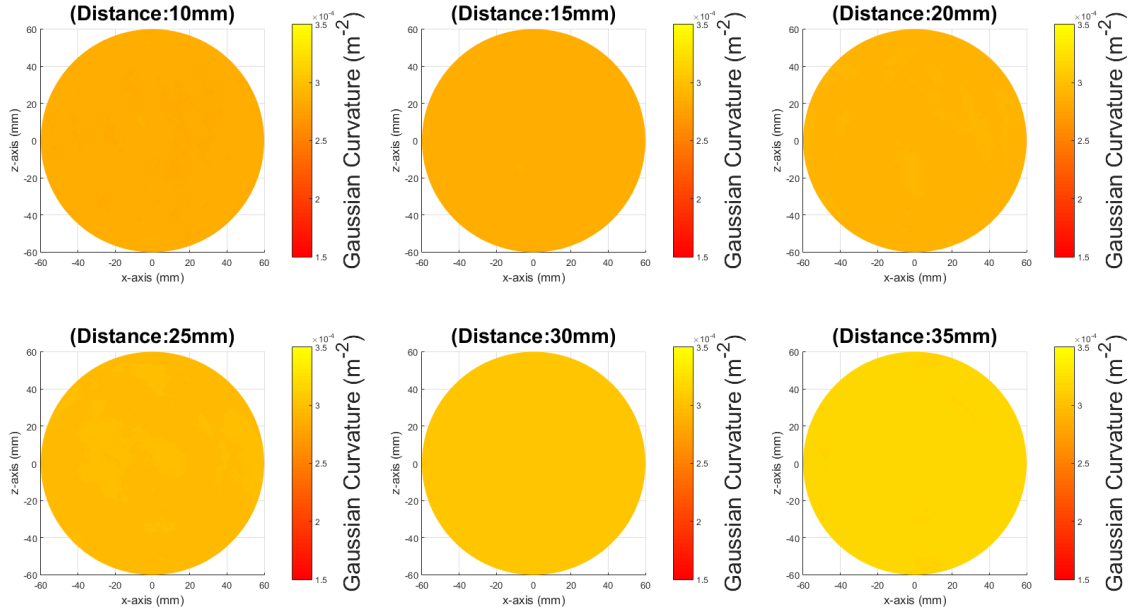


Figure 16 Curvature estimations for a full icosphere ( $R_S = 60$  mm) with different neighbourhood sizes:  $R_N = 10, 15, 20, 25, 30, 35$  mm

#### 4.3.2.2 11 Months old skull

Figure 17 shows the results of the curvature estimation for an 11 months old skull, for different neighbourhood sizes. Due to many different curvatures on the skull, a great variance in the curvature values existed. This makes direct representation of the curvature values by a colour map difficult. First, because a skull is spherical-like shaped and the Gaussian curvature of a sphere is defined as  $\frac{1}{R_S^2}$ , the curvature values were represented in  $R_S$  to remove the exponential distribution. Then the logarithmic filter defined in chapter 4.2.2.3 (see equation (6)) was applied in order to reduce the effect of large variations in the curvature values. This enabled appropriate representation of the results by a colour map. The results confirm the expectations on scaling described in chapter 4.2.2.4: the smaller the neighbourhood size, the more details of the skull are distinguished by the curvature estimation. The

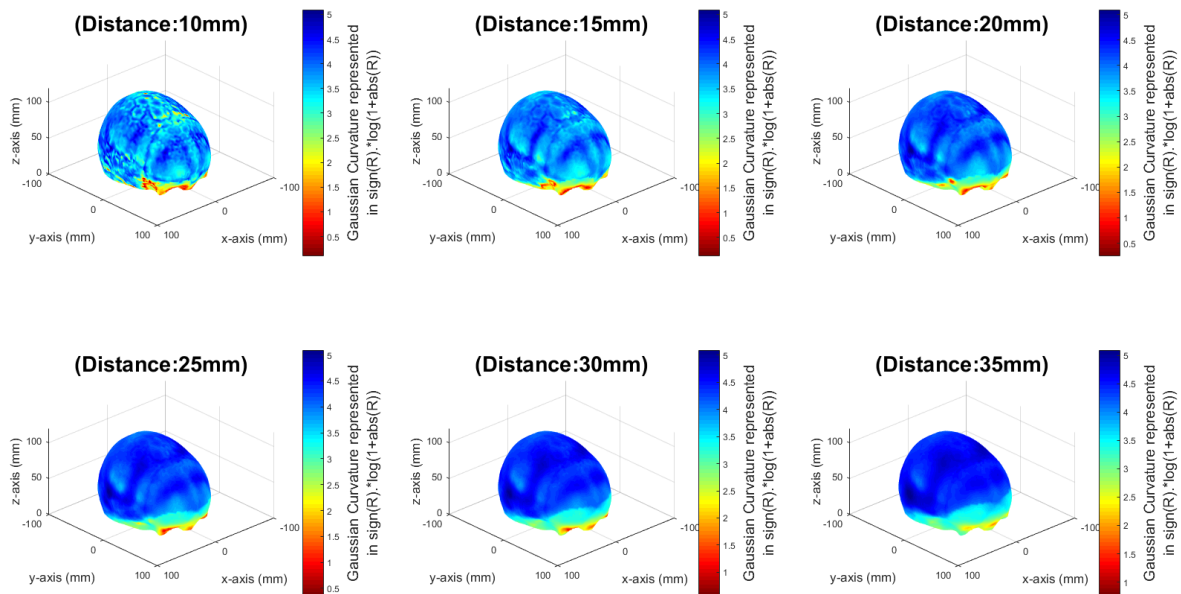


Figure 17 Curvature estimations for a 11 months old skull with different neighbourhood sizes  $R_N = 10, 15, 20, 25, 30, 35$  mm

neighbourhood radius of 15 mm appeared to give the best results on detail discrimination and was used for further analysis.

### 4.3.3 Shape comparison

#### 4.3.3.1 Test object

Figure 18 shows the results of the shape comparison for the test object. The results show that the curvature colour maps of both objects were similar, which confirmed that both test objects contained similar shapes. (Figure 18(*Middle*)). Hereby, the method was able to find the region that matches the reference region (Figure 18(*Top and Middle*)). The curvature maps of the matched regions are shown in (Figure 18(*Bottom*)) and show similar patterns, with small differences.

#### 4.3.3.2 Shape comparison within a single skull

The curvature colour map in Figure 19(*Top right*) confirms that the shape of the skull is symmetrical, with some subtle differences between left and right. The figure also shows the two best matching regions found by the shape comparison. The first match was found close to the reference region. The second match was the temple on the opposite side of the reference region. The curvature maps in Figure 19(*Bottom*) show that the curvature maps of both matches contain similar patterns to the curvature map of the original region. The curvature map of the first match shows more similarity to the reference than the second match.

#### 4.3.3.3 Shape comparison in a trigonocephaly case

Results of the shape comparison for the trigonocephaly case are shown in Figure 20. The normative skull model and its curvature colour map are shown on the left side of the figure. The trigonocephaly skull and its curvature colour map are shown on the right side of the figure. The trigonocephaly case is shown in the x-y plane to show the distinguishable sharp curved forehead of the patient. The sharp curved forehead is also represented by the curvature colour map: a relatively large curvature can be distinguished on the forehead. Also, the colour map of the trigonocephaly case shows more varieties than the colour map of the normative skull. This agrees with the larger amount of irregularities shown in the skull model surface compared to the surface of the normative skull. The results show that the best match was found at the parietal bone of the trigonocephaly skull.

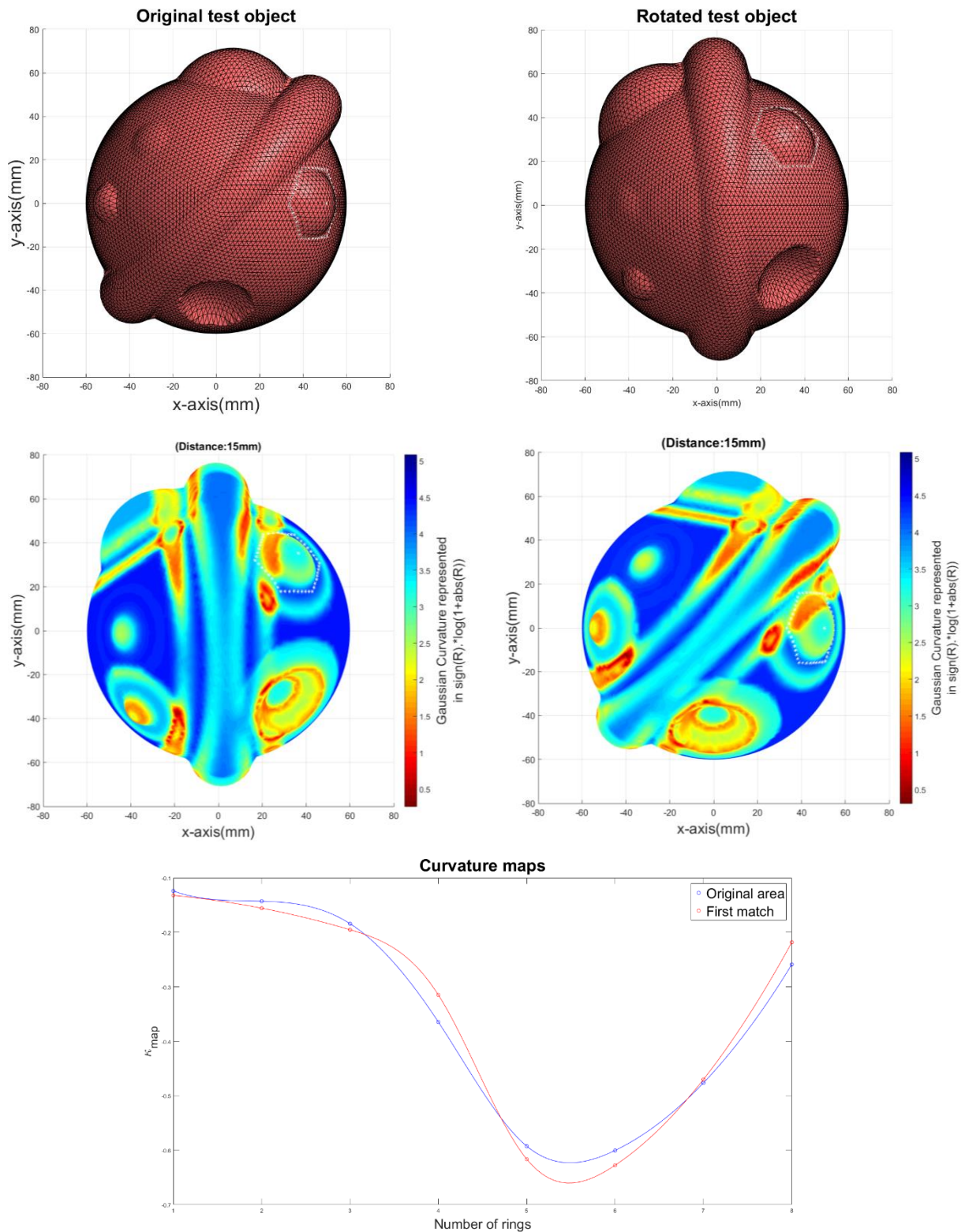


Figure 18 Shape comparison of a test object: (Top left) original test object with selected reference region (Middle left) Curvature colour map of original test object with selected reference region (Top right) Rotated test object with mathed region (Middle left) Curvature colour map of rotated test object with mached region (Bottom) Curvatures maps of reference region and matched region

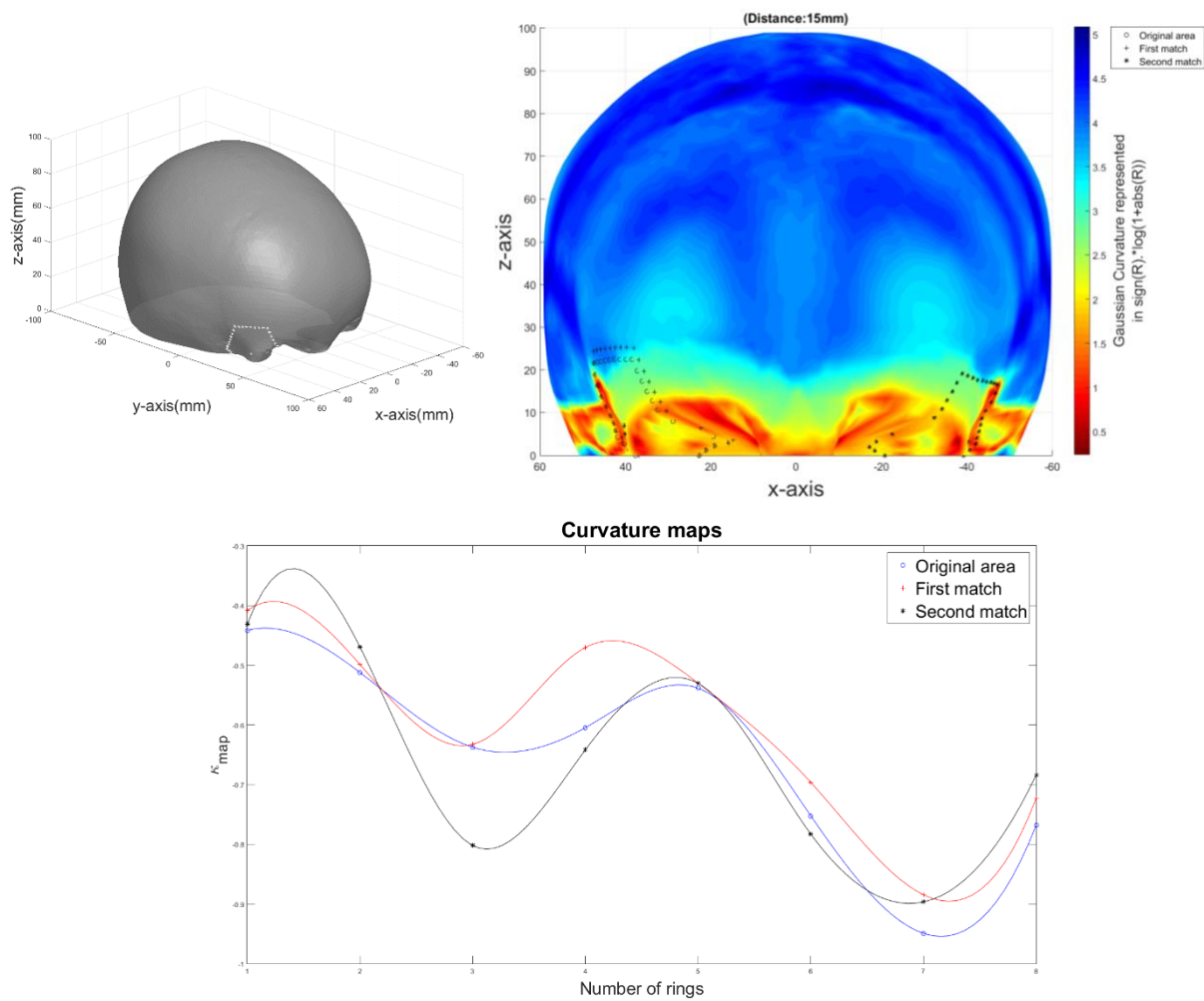


Figure 19 Shape comparison within a single skull: (Top left) Skull with selected reference region (Top right) Curvature colour map of skull with selected reference region and first 2 matches

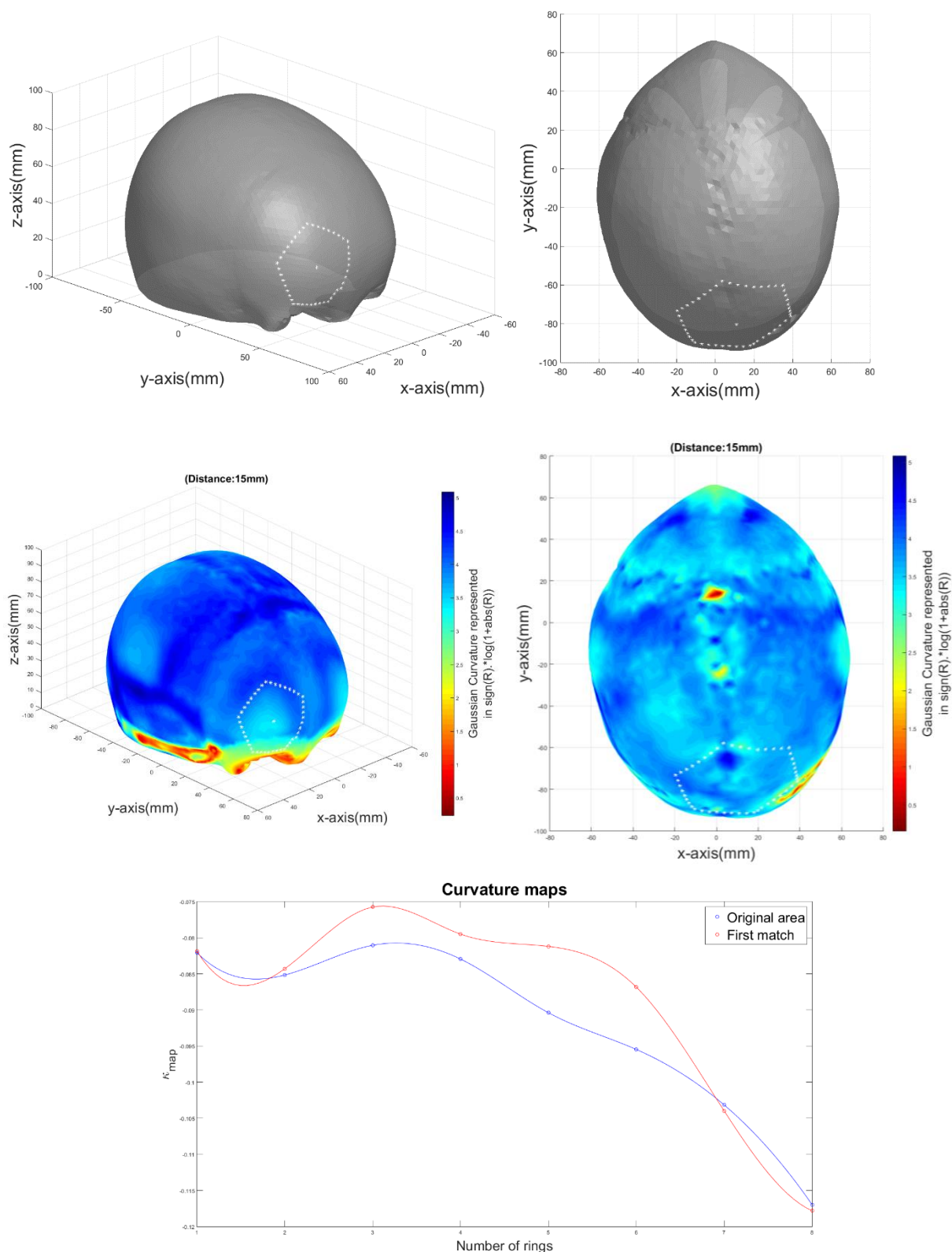


Figure 20 Shape comparison of a trigonocephaly with an age-appropriate normative skull: (Top left) Normative skull with selected reference region (Top right) Trigonocephaly skull with matched region (Middle left) Curvature colour map of normative skull with selected reference region (Middle right) Curvature colour map of trigonocephaly skull with matched region (Bottom) Curvature maps of reference region and matched region

## 4.4 DISCUSSION

Some challenges arose for the curvature computation of mesh data. The definition of curvature in Supplement 0 is based on the curvature computation of parametric continuous closed functions. Due to the mesh representation of the objects and skulls no such functions were available. Mesh data contains discrete data and provides no information about the surface around the specific point on which the curvature is computed. Therefore, the curvature of mesh data could only be estimated. Several authors have paid attention to curvature estimation methods on 3D mesh data[37]–[44]. Gatzke et al.[38] have published a review in which they compared discrete estimation methods with surface fitting methods. Discrete methods directly estimate the curvature based on the normal vectors of the mesh and the contribution of faces and edges of a neighbourhood to the surface. Surface fitting methods however, first solve a least square fitting problem to obtain a continuous function that approximates a neighbourhood surface for a specific vertex. This function is then used to estimate the curvature in that vertex. The review found that discrete functions were fast but very sensitive to noise and required a regular mesh valence and resolution. Fitting methods were slower but appeared less sensitive to noise and valence, if the valence was over 4. Also, the fitting methods were found more accurate when the surface fitting was based on a multi-ring neighbourhood. All methods were found valence independent when there were more than 4 vertices within a one-ring neighbourhood. Another review was published by Magid et al.[37] in which they compared the influence of resolution and neighbourhood sizes for both discrete and surface fitting methods. This review also concluded that multi-ring surface fitting methods were most accurate and robust for the curvature estimation of mesh data. Finally, within all fitting methods quadric fitting appeared to be the most accurate and robust[37]–[39], [42]–[44]. In conclusion, multi-ring quadric surface fitting appeared to be the best choice in curvature estimation techniques for skull mesh data and was therefore used within this study.

Furthermore, this study took variation in resolution and connectivity of the mesh data, and holes within mesh data into account. First, the mesh data of the skulls were sampled by ray casting with an evenly distributed spatial resolution of rays. Nonetheless, chapter 2 notes that because all rays were casted from a single central point, an irregular resolution of the mesh data was created due to the different intersection distances (Figure 5). Due to the sphere-like shape of the skulls this effect was expected to be minimal. Still, both the curvature estimation and the curvature map algorithm compensated for variations in sample resolution by normalising with respect to the area of faces. Secondly, because all mesh data was sampled based on the structure of the icosphere, the connectivity was known. This provided the knowledge of the number of neighbours for each vertex and enabled the implementation of the ring-based neighbourhood search. Finally, results in chapter 4.3.2.1 showed that the accuracy of the curvature estimation was reduced when the quadric surface fitting was not based on a symmetrical neighbourhood. Therefore, holes within the mesh could lead to inaccurate curvature estimations for vertices near the borders of the holes. Because all holes were filled this problem was prevented.

The appropriate neighbourhood size for the quadric surface fitting method was based on the amount of detail shown by the curvature colour map. Small ridges in the skull surface are irrelevant for the establishment of the appropriate shape during the reconstruction of the cranial vault. Therefore, the neighbourhood size of 10 *mm* appeared to be inappropriate, because the corresponding curvature colour map showed high detail discrimination. However, a lack of knowledge about the detail of the skull's curvature could lead to unwanted ridges and bumps on the skull's surface. This excluded the neighbourhood sizes 20 – 35 *mm*, because the corresponding curvature colour maps showed a lack of detail. Moreover, Rusinkiewicz et al.[43] stated that where a small neighbourhood results in very accurate curvature estimation for clean data, increasing the neighbourhood size reduces the sensitivity for noise. The results in chapter 4.3.2.1 confirmed that a smaller neighbourhood size resulted in more accurate curvature estimations. In conclusion, the neighbourhood distance ( $R_N$ ) of 15 *mm* appeared to be the best choice.

It should be noted that the choice for this neighbourhood size was based on the 11 skull. This is a common age during open cranial vault reconstruction at the Radboudumc and the same age as the presented trigonocephaly case. It is expected that different scaling is needed for skulls of different ages. The cranial vault is smaller at lower age and the details to be detected are smaller as well. This could require smaller neighbourhood sizes. On contrast, at older age the scaling should be based on larger details. Thus, it could be possible that for older ages a larger neighbourhood size should be selected. Further evaluation on the scaling is needed in order to determine the appropriate scaling for each age group.

Within this study, the behaviour of the quadric surface fitting method was not fully understood. The curvature estimation for the icospheres resulted in an overestimation of the curvature. For the full icosphere, this overestimation increased with the neighbourhood size. Because the curvature was estimated for an icosphere that contained minimal noise, it was not expected that this inaccuracy could be explained by the larger influence of noise due to the contribution of a larger neighbourhood. In contrary, it was expected that a larger neighbourhood size would result in a better approach of the surface by the quadric surface and thereby a better estimation of the curvature: larger neighbourhood sizes contain more vertex neighbours and therefore more information about the surface around a vertex. However, analysis of the quadric surfaces showed that the quadric surface fitting method was more inaccurate when the neighbourhood size increased. For the analysis of the quadric surface fitting method, it was desirable to visualise the quadric surfaces. However, problems existed due to memory restrictions of Matlab®. In order for Matlab® to visualise the quadric surface, an  $x$ -,  $y$ -,  $z$ -grid is defined and each coordinate combination is filled in equation (5). Then, Matlab® creates a surface plot of all solutions  $F(x, y, z) = 0$ . Figure 21 illustrates that for a grid with interval  $[0,1]$  with step size 0.1 for  $x$  and  $y$  and  $z = a$  only 2 solutions are found. The more dense the created grid, the greater the chance that more solutions are found. Due to the memory capacity of Matlab® the density of the grid was limited, whereas the results for all  $x, y$ , and  $z$ -combinations need to be stored. As result, it was not possible to use a grid that was dense enough to find enough solutions to create a surface plot. A solution for this problem might be to encapsulate the quadric surface. This method is illustrated in Figure 21 by the coloured lines. An iterative process is followed that finds the matching coordinates belonging to each separate  $x, y$  and  $z$ -coordinate in the grid that results in the solution of equation (2). For each  $x, y$ , and  $z$ -coordinate a solution would be found and the quadric surface could be visualised and analysed. An expected drawback of this method is that the process will be very slow and will only be feasible for simple surfaces.

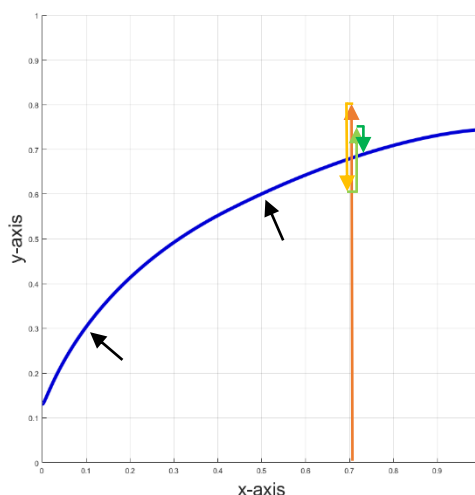


Figure 21 Quadric surface represented in the  $x, y$ -plane for  $z = a$  (blue line) with the found solutions for the chosen grid by the current method (black arrows) and the solution found by encapsulation (coloured arrows)

The curvature maps that were used for the shape comparison did not include angular information about the curvature variance within regions. For each vertex the mean Gaussian curvature per ring was computed. A function was created describing the relation between the distance to the vertex and the mean Gaussian curvature of  $n$ -ring neighbourhoods. The shape comparison algorithm compared these functions, the curvature maps, in order to define the shape correspondence between different regions. The angular information about the curvature variance within the regions is lost with the use of the mean Gaussian curvatures per ring. Gatzke et al. [33] also describe curvature maps that contains angular information of the shape correspondence. This method uses geodesic fans instead of  $n$ -ring neighbourhoods. Next to the computation of the mean Gaussian curvature at a certain distance from a centre vertex, the mean Gaussian per geodesic spoke is computed. This allows both the shape



comparison of the curvature related to the distance from the centre vertex and the curvature related to the angle of the geodesic fan. It could be useful to implement this method because the shape comparison would then not only give information about which region matches the reference region, but also about how to rotate the matched region to fit the reference region best.

The results on the test object confirm that the shape comparison was able to identify areas with similar shape. The curvature colour maps showed corresponding curvature results, which is expected for objects with similar shape. However, the curvature map of the matched area showed small differences with the curvature map of the reference area. This difference was assigned to the estimation inaccuracy of the quadric surface fitting method. The matched region was not located at the torus. This showed that the algorithm distinguishes different shapes. Because the spherical bulges with different radii were not matched, it is shown that the curvature maps are scale variant. This is important for the planning technique, because small ridges and bulges should be discriminated from big ridges and bulges. Moreover, due to the similar radii of the hole and the reference area, the inside surface of the hole is similar to the outside shape of the bulge. Because it is not possible to use the inside surface of the cranium for the reconstruction, it is important that the shape comparison discriminates bulges from holes of similar shape. The test results show that the current algorithm is able to do so. Also, when testing the algorithm on a normal skull, we expected that a shape on one side of the skull could also be found on the other side of the skull, given the fact that a healthy skull is (almost) symmetrical. Indeed, the algorithm led to a first match that was almost the same area as the reference area.

Furthermore, when testing the algorithm on a normal skull, we expected that a shape on one side of the skull could also be found on the other side of the skull, given the fact that a healthy skull is (almost) symmetrical. Indeed, the algorithm led to a first match that was almost the same area as the reference area. Because the second match was found at the symmetrical region on the left side of the skull, it could be concluded that the shape comparison algorithm is able to identify corresponding shapes correctly.

For the trigonocephaly case, the area with the optimal shape for the selected reference area, the right side of the forehead, was located at the parietal bone. This result is in contradiction with the common practice, whereas literature only describes reconstruction techniques for trigonocephaly skulls that concern only the use of the frontal bone [10]–[13], [16], [17], [21]. Because no appropriate age-specific 3D normative data was available till now, these techniques were originated from experience, subjective decision making (what is beauty?) and common practice. However, the results of this study were fully based on objective shape comparison with age-appropriate 3D normative data and propose that for the presented case, a totally different reconstruction technique could be considered. According to the results of this preliminary study, the parietal bone and not the contralateral frontal bone, would give rise to the ‘best’ frontal reconstruction of the presented trigonocephaly case. Albeit a first trial of the developed algorithm, this case shows clearly that the use of a fully objective automated pre-operative planning technique could be able to achieve better results by introducing out-of-the box solutions, based on reference data.

#### 4.5 CONCLUSION

This study has implemented an algorithm for the local shape comparison of skulls. First, the curvature estimation for predefined quadrics verified the curvature formulas for implicit functions. Then, it was shown that the Gaussian curvature based on quadric surface fitting was estimated correctly and that the shape comparison algorithm was able to identify similar shapes. The shape comparison algorithm was tested for a trigonocephaly case. Hereby, the algorithm identified a region on the parietal bone of the patient's skull that maximally corresponded with the reference shape. In conclusion, the results showed the feasibility of the algorithm to identify the region on a patient's skull that maximally corresponds in shape with a reference shape.



## 5 GENERAL CONCLUSION AND FUTURE PROSPECTS

This study has set the first steps towards an automated pre-operative planning technique for open cranial vault reconstructions in craniosynostosis patients. Age-specific normative skulls were created for different age groups. This provided knowledge about the normal shape of the cranium at different ages. Also, an algorithm was developed that enables the identification of the region on the patient's skull that contains the shape that maximally corresponds with an age-appropriate reference shape. By integrating this algorithm within the automated planning technique, appropriate regions on the patient's skull that can be used to create the optimal shape for the reconstruction site can be found. As result, the open cranial vault reconstructions will not be based on the subjective decision making of the surgical team, but on objective foundations.

There is still a long way to go in the development of the automated pre-operative planning technique for open cranial vault reconstructions. A next step is to create a cost function that is able to define the optimal osteotomies and relocation of the created osseous panels in order to create the optimal cranial vault shape. This cost function should add restrictions to the location of osteotomies, the number of osseous panels and should take the establishment of a stable reconstructed structure into account.

The location of the ostomies is important to guarantee the patient's safety. The sigmoid sinus and superior sagittal sinus are major superficial vascular structures that can easily be damaged by the craniotomy drill. Also, at the site of the existing sutures the dura is often attached to the cranium. Crossing a suture with the drill could lead to dura tears. Other examples of structures that should be avoided are the supra-orbital artery and nerves. For the full automation of the pre-operative planning technique it is important that critical structures can be recognised by the algorithm. Jafarian et al. [45] describe a method for the segmentation of the sutures in the new-born's skull. The feasibility of the localisation of the other structures by imaging analysis techniques is yet to be determined.

The number of osseous panels is dependent on the reference region size. The shape comparison algorithm identifies an region on the patient's skull surface with the same size as the reference region. Small osseous panels facilitate the creation of the optimal cranial shape but increase the number of panels. A large number of panels complicate the creation of a stable structure and require more screws and plates for the fixation. This increases the costs and duration of the procedure. In contrast, by selecting a reference region that is too large, ideal regions could be missed. This is because it is possible that an region on the patient's skull exists that ideally fits only a part of the selected reference region. In conclusion, the optimal reference size should be determined. Variation of the reference size, but also shape should be considered.

The stability of the structure is dependent on the fixation of the reconstructed structure. The osseous panels are secured with plates and screws. A large distance between the panels or between the panels and the cranium increases the risk of fracturing the plates and failure of the screws. Fixation of screws in unstable osseous tissue, for example where the tissue is thin, can also lead to failure of the structure. Overlay of the panels should be avoided because this results in ridges on the skulls surface. Moreover, the reconstructed structure should have a stable foundation. For example, in frontal orbital reconstructions the structure is often built up with the orbitae ridge as base. A stable structure is essential to prevent the structure from failing and to avoid brain tissue damage when external pressure on the structure is applied.

This study expects that a pre-operative plan for open cranial vault reconstructions in craniosynostosis patients created by a fully automatic and objective algorithm, will give better results than a plan created based on the subjective decision making of the surgical team. In order to confirm these expectations, the results of both plans should be compared. To avoid ethical issues the reconstructions can initially be performed on 3D printed skulls that are based on CT-scans. First, the feasibility of the automatic algorithm should be determined. Reconstructions on printed skulls of several craniosynostosis cases

should be performed according to preoperative plans created by the algorithm. Then, for several craniosynostosis cases predefined plans should be created by both the automatic algorithm and the surgical team. Both plans should be executed by the same surgeon and the results can be evaluated by comparing the reconstructed skulls with the age-specific normative skull.

## 6 REFERENCES

- [1] W. A. Borstlap, E. B. Wolvius, S. J. Berge, R. Dammers, M. J. Koudstaal, and E. J. Van Lindert, "Centralisatie van craniosynostose in Nederland," *Tijdschr. Kindergeneeskd.*, vol. 82, no. 3, pp. 106–111, 2014.
- [2] European Commission, "Public Health," *Rare Diseases*, 2016. .
- [3] S. Raj, R. Sheth, and A. Kao, "MedScape," *Pediatric Craniosynostosis*, 2015. [Online]. Available: <http://emedicine.medscape.com/article/1175957-overview#a5>. [Accessed: 21-Apr-2016].
- [4] S. Nagaraja, P. Anslow, and B. Winter, "Craniosynostosis.," *Clin. Radiol.*, vol. 68, no. 3, pp. 284–92, Mar. 2013.
- [5] D. Johnson and A. O. M. Wilkie, "Craniosynostosis.," *Eur. J. Hum. Genet.*, vol. 19, no. 4, pp. 369–76, Apr. 2011.
- [6] O. Kirmi, S. J. Lo, D. Johnson, and P. Anslow, "Craniosynostosis: A Radiological and Surgical Perspective," *Semin. Ultrasound, CT MRI*, vol. 30, no. 6, pp. 492–512, Dec. 2009.
- [7] Nederlandse vereniging van Neurochirurgen, "Craniosynostose," *august, 29*, 2014. [Online]. Available: [http://www.nccn.nl/nccn/nieuws/craniosynostose\\_hier\\_vindt\\_u\\_alle\\_informatie](http://www.nccn.nl/nccn/nieuws/craniosynostose_hier_vindt_u_alle_informatie).
- [8] J. Thijssen and I. Mathijssen, "Richtlijn Behandeling en Zorg voor Craniosynostose." pp. 1–26, 2012.
- [9] M. C. Van Veelen, D. Mihajlović, R. Dammers, H. Lingsma, L. N. A. Van Adrichem, and I. M. J. Mathijssen, "Frontobiparietal remodeling with or without a widening bridge for sagittal synostosis: comparison of 2 cohorts for aesthetic and functional outcome," *J. Neurosurg. Pediatr.*, vol. 16, no. July, pp. 86–93, 2015.
- [10] R. Guzman, J. F. Looby, S. a Schendel, and M. S. B. Edwards, "Fronto-orbital Advancement Using an En Bloc Frontal Bone Craniectomy," *Neurosurgery*, vol. 68, no. March, p. ons68-ons74, 2011.
- [11] A. Shah, A. Patel, and D. M. Steinbacher, "Simulated Frontoorbital Advancement and Intraoperative Templates Enhance Reproducibility in Craniosynostosis," *Plast. Reconstr. Surg.*, vol. 129, no. 6, p. 1011e–1012e, 2012.
- [12] A. K. Hormozi, R. Shahverdiani, H. R. Mohammadi, A. Zali, and H. R. H. Mofrad, "Surgical treatment of metopic synostosis.," *J. Craniofac. Surg.*, vol. 22, no. 1, pp. 261–5, 2011.
- [13] F. Ozlen, A. M. Kafadar, B. Abuzayed, M. O. Ulu, C. Isler, R. Dashti, and P. Erdinçler, "Surgical treatment of trigonocephaly: technique and long-term results in 48 cases.," *J. Neurosurg. Pediatr.*, vol. 7, no. 3, pp. 300–10, 2011.
- [14] G. Silav, G. Avci, M. Akan, G. Taylan, I. Elmaci, and T. Akoz, "The surgical treatment of plagiocephaly," *Turk. Neurosurg.*, vol. 21, no. 3, pp. 304–314, 2011.
- [15] P. P. Sun and J. A. John, "Craniosynostosis - surgery," in *Principles and Practice of Pediatric Neurosurgery*, L. Albright, I. Pollack, and D. Adelson, Eds. New York: Thieme Medical Publishers, Inc., 1999, pp. 227–237.
- [16] L. Bottero, E. Lajeunie, E. Arnaud, D. Marchac, and D. Renier, "Functional outcome after surgery for trigonocephaly.," *Plastic and reconstructive surgery*, vol. 102, no. 4. pp. 952-8-60, 1998.
- [17] S. Park, Tae and S. Robinson, "Nonsyndromic Craniosynostosis," in *Paediatric Neurosurgery: Surgery of the Developing Nervous System*, 4th ed., R. Zorab and S. Reilly, Eds. Philadelphia: W.B. Saunders Company, 2001, pp. 345–395.

- [18] N. R. Saber, J. Phillips, T. Looi, Z. Usmani, J. Burge, J. Drake, and P. C. W. Kim, "Generation of normative pediatric skull models for use in cranial vault remodeling procedures.," *Childs. Nerv. Syst.*, vol. 28, no. 3, pp. 405–10, Mar. 2012.
- [19] J. R. Marcus, L. F. Domeshek, and R. Das, "Objective Three-Dimensional Analysis of Cranial Morphology," *J. Plast. Surg.*, pp. 175–187, 2008.
- [20] S. Mardini, S. Alsubaie, C. Cayci, H. Chim, N. Wetjen, and C. A. D. Cam, "Three-dimensional preoperative virtual planning and template use for surgical correction of craniosynostosis," *Br. J. Plast. Surg.*, vol. 67, no. 3, pp. 336–343, 2014.
- [21] J. Soleman, F. Thieringer, J. Beinemann, C. Kunz, and R. Guzman, "Computer-assisted virtual planning and surgical template fabrication for frontoorbital advancement," *Neurosurg. Focus*, vol. 38, no. May, pp. 1–8, 2015.
- [22] T. Clijmans, M. Mommaerts, P. Suetens, and J. Vander Sloten, "Automated reconstruction planning of cranial malformations based on reference data," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 2, no. Supplement 1, pp. S228–S229, 2007.
- [23] T. Clijmans, F. Gelaude, M. Mommaerts, P. Suetens, and J. Vander Sloten, "Computer Supported Pre-Operative Planning of Craniosynostosis Surgery: a Mimics-Integrated Approach," *Innovation*, pp. 1–12, 2006.
- [24] E. W. Lorensen and E. H. Cline, "Marching cubes: a high resolution 3D surface construction algorithm," *Comput. Graph. (ACM)*, vol. 21, pp. 163–169, 1987.
- [25] M. Tolhuisen, G. de Jong, J. Meulstee, and H. Delye, "A Method for the Creation of Normative Paediatric Skull Models: A Pilot Study," *10th European Craniofacial Congress*, 2015. .
- [26] P. Silva da Costa, Marcos Devanir MD; Fernandes, Bruno MD; de Araujo Paz, Daniel MD; Rodrgiues, Thiago Pereira MD; Abdala, Nitamar MD, PhD; Centeno, Ricardo Silva MD, PhD; Cavalheiro, Sergio MD, PhD; Lawton, Michael T. MD; Chaddad-Neto, Feres MD, "Anatomical Variations of the Anterior Clinoid Process: A Study of 597 Skull Base Computerizes Tomography Scans," *Oper. Neurosurg.*, 2015.
- [27] E. Kapur and A. Mehić, "Anatomical variations and morphometric study of the optic strut and the anterior clinoid process," *Bosn. J. Basic Med. Sci.*, vol. 12, no. 2, pp. 88–93, 2012.
- [28] C. L. Bendon, F. B. J. Sheerin, S. A. Wall, and D. Johnson, "The relationship between scaphocephaly at the skull vault and skull base in sagittal synostosis," *J. Cranio-Maxillofacial Surg.*, vol. 42, no. 3, pp. 245–249, 2014.
- [29] E. J. van Lindert, F. J. Siepel, H. Delye, A. M. Ettema, S. J. Bergé, T. J. J. Maal, and W. a Borstlap, "Validation of cephalic index measurements in scaphocephaly.," *Childs. Nerv. Syst.*, vol. 29, no. 6, pp. 1007–14, Jun. 2013.
- [30] J. Van Der Meulen, "Metopic synostosis," *Child's Nerv. Syst.*, vol. 28, no. 9, pp. 1359–1367, 2012.
- [31] H. Delye, T. Clijmans, M. Y. Mommaerts, J. van der Sloten, and J. Goffin, "Creating a normative database of age-specific 3D geometrical data, bone density, and bone thickness of the developing skull: a pilot study," *J. Neurosurg. Pediatr.*, pp. 1–16, 2015.
- [32] A. A. Waitzman, J. C. Posnick, D. C. Armstrong, and G. E. Pron, "Craniofacial Skeletal Measurements Based on Computed Tomography: PartII Normal Values and Growth Trends," *Cleft Palate-Craniofacial J.*, vol. 29, no. 2, pp. 118–128, 1992.
- [33] T. Gatzke, C. Grimm, M. Garland, and S. Zelinka, "Curvature Maps For Local Shape Comparison," in *Shape Modeling and Applications, 2005 International Conference*, 2005, pp. 244–253.

- [34] H. Hoppe, "New quadric metric for simplifying meshes with appearance attributes," *Proc. Vis. '99 (Cat. No.99CB37067)*, 1999.
- [35] M. Garland and P. S. Heckbert, "Simplifying surfaces with color and texture using quadric error metrics," *Proc. Vis. '98 (Cat. No.98CB36276)*, pp. 263–270, 1998.
- [36] R. Goldman, "Curvature formulas for implicit curves and surfaces," *Comput. Aided Geom. Des.*, vol. 22, no. 7, pp. 632–658, 2005.
- [37] E. Magid, O. Soldea, and E. Rivlin, "A comparison of Gaussian and mean curvature estimation methods on triangular meshes of range image data," *Comput. Vis. Image Underst.*, vol. 107, pp. 139–159, 2007.
- [38] T. D. Gatzke and C. M. Grimm, "Estimating Curvature on Triangular Meshes," *Int. J. Shape Model.*, pp. 1–21, 2005.
- [39] P. Krsek, G. Luk, and R. R. Martin, "Range Data Differential parameters in 3D computer vision," *Math. Surfaces VIII, Inf. Geometers*, pp. 1–16, 1998.
- [40] P. Krsek, T. Pajdla, and V. Hlavac, "Estimation of Differential Parameters on Triangulated Surface Reference," *Electr. Eng.*, 1997.
- [41] H. S. Martial and H. Katsushi, "On 3D Shape Similarity," in *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference*, 1996, no. June, pp. 526–531.
- [42] B. Hamann, "Curvature approximation for triangulated surfaces," *Geometric modelling*. pp. 139–153, 1993.
- [43] S. Rusinkiewicz, "Estimating Curvatures and Their Derivatives on Triangle Meshes," in *3D Data Processing, Visualization and Transmission*, 2004, pp. 486–493.
- [44] A. M. McIvor and R. J. Valkenburg, "A comparison of local surface geometry estimation methods," *Mach. Vis. Appl.*, vol. 10, pp. 17–26, 1997.
- [45] N. Jafarian, K. Kazemi, H. Abrishami Moghaddam, R. Grebe, M. Fournier, M. S. Helfroush, C. Gondry-Jouet, and F. Wallois, "Automatic segmentation of newborns' skull and fontanel from CT data using model-based variational level set," *Signal, Image Video Process.*, vol. 8, no. 2, pp. 377–387, Mar. 2012.
- [46] D. Dietz and H. Iseri, "Calculus and Differential Geometry : An Introduction to Curvature."
- [47] Mark.L.Irons, "The shape operator of a Torus," 2004. [Online]. Available: <http://www.rdrop.com/~half/math/torus/shape.operator.xhtml>. [Accessed: 18-May-2016].
- [48] B. O'Neill, *Elementary Differential Geometry*, vol. 85, no. 503. 2010.
- [49] M. Penna, "Quadric Surfaces," *Mathematica*, vol. 71, pp. 2005–2005, 2005.
- [50] Wikipedia, "Barycentric coordinate system," 2016. [Online]. Available: [https://en.wikipedia.org/wiki/Barycentric\\_coordinate\\_system](https://en.wikipedia.org/wiki/Barycentric_coordinate_system). [Accessed: 29-Aug-2016].
- [51] Wikipedia, "Icosahedron," 2016. [Online]. Available: <https://en.wikipedia.org/wiki/Icosahedron>. [Accessed: 29-Aug-2016].





## 7 SUPPLEMENT

### 7.1 MARCHING CUBES

Lorensen et al.[24] describe a fast intensity-to-mesh algorithm called marching cubes. This method ‘marches’ a cube, consisting of 8 logical vertices, through the 3D voxel data. The four upper vertices and the four lower vertices are always located in two different adjacent slices of the image data, so that the connectivity of slices is known. For each step the method determines how the surface with a certain intensity threshold intersects the cube. Hereby, all vertices of the cube for which the intensity value exceeds the intensity threshold are given the value 1. The other vertices are given the value zero. This enables the storage of the intersections in only 8 bits per cube. Because the cube can only be intersected in a limited amount of ways, the method can use a look-up table that describes 14 different options (see Figure 22). Symmetrical versions of the options are given the same index. Linear interpolation is finally used to determine the exact intersection of the surface with the edges of the cube. For each step in the marching through the voxel data, the intersection index of the cube is determined and the triangulation of the surface is constructed.

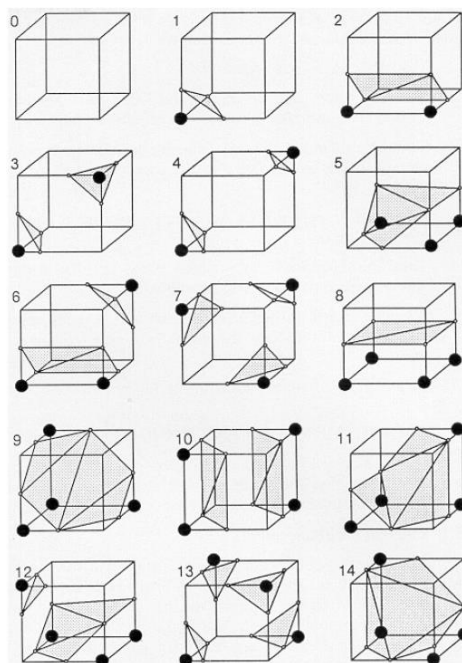


Figure 22 Look-up table of the marching cubes algorithm that is used to describe the surface triangulation, source: Lorensen et al.[24]

### 7.2 ICOSPHERE

An icosphere is built up from an icosahedron, which consists of 8 equilateral triangles that are defined by the corners of 4 orthogonal congruent rectangles. A more detailed icosphere is obtained by an iterative process in which the triangles are divided in smaller equilateral triangles. With every iteration the icosphere will consist of smaller triangles, thus will contain more data and will be more detailed. Because the icosphere is built up by equilateral triangles it consists of evenly distributed data points.

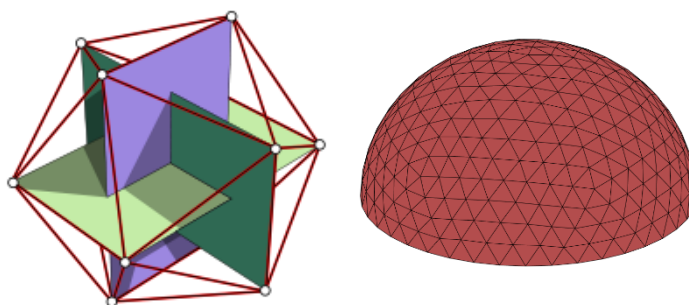


Figure 23 (Left) icosahedron, source: Wikipedia[51], (Middle) icosphere

### 7.3 MÖLLER-TRUMBORE

The Möller-Trumbore technique is a quick method that can determine whether a ray intersects a triangular face of the mesh. Each face consists of three vertices ( $V_1, V_2, V_3$ ). The method defines points on the triangle with barycentric coordinates. The triangle is transformed to a unit triangle and translated to the origin so that the coordinates of the vertices are  $(1,0,0)$ ,  $(0,1,0)$  and  $(0,0,1)$ . The location of an arbitrary point on the surface of the triangle is then expressed with the ratios  $u, v$  and  $w = 1 - u - v$  (see Figure 24). Using this coordinate system a coordinate of a point on a surface can be expressed as:

$$T(u, v) = (1 - u - v)V_0 + uV_1 + vV_2$$

A ray is defined by:

$$R(t) = O + tD$$

$D$  is the normalised direction,  $O$  the origin of the ray and  $t$  the length of the ray. To determine whether the ray crosses the triangle, the algorithm then solves the following equation:

$$T(u, v) = R(t)$$

### 7.4 CURVATURE

Dietz et al.[46] use Figure 25 to illustrate the definition curvature of a curve in  $\mathcal{R}^2$ . A blunt curve and sharp curve are shown on the left and centre respectively. The average curvature is defined as the rotation of the unit normal vector with respect to the arc length  $s$  over which the rotation occurs:

$$\text{average curvature} = \frac{\text{total rotation}}{\text{arc length}} \quad (8)$$

The total rotation is given by  $\theta$  and is equal for both the blunt and the sharp curve. The arc length over which the rotation occurs is greater for the blunt curve than for the sharp curve. Therefore, according to (8), the average curvature of the sharp curve is greater than of the blunt curve, which is intuitively correct.

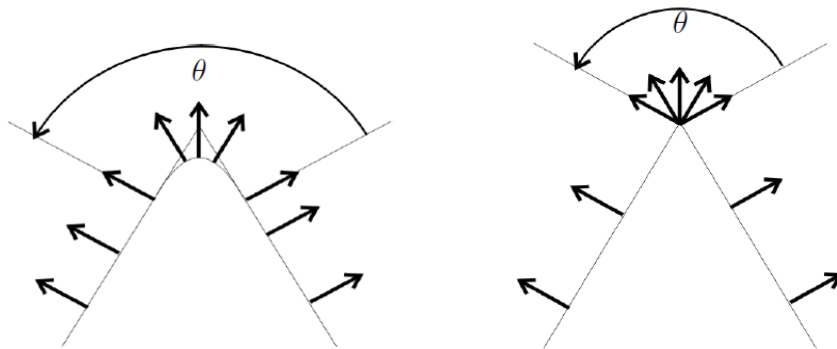


Figure 25 Example of a blunt curve (left) and a sharp curve (right), source: Dietz et al.[46]

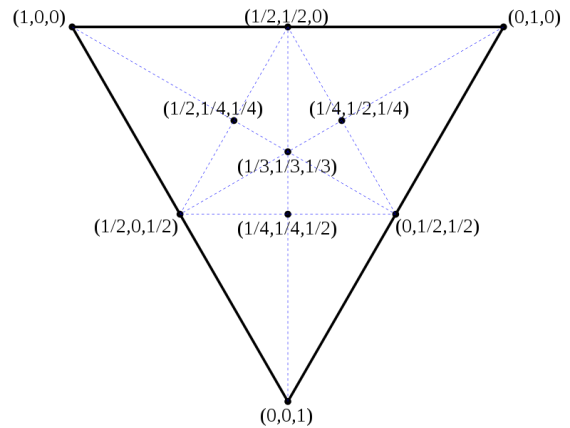


Figure 24 Barycentric coordinates expressed in three ratios, source: Wikipedia[50]

To obtain specific information about the shape of a curve in each point on the surface, the arc length should reach to zero. This gives curvature as a point metric, being defined as the derivative of the unit normal, over the length of rotation:

$$\kappa(s) = \left\| \frac{d\mathbf{n}}{ds} \right\|$$

Because the rotation of the normal vector is used, this curvature is called *normal curvature* in literature.

For the computation of the normal curvature in  $\mathcal{R}^3$  literature refers to the *shape operator*[47], [48]. The shape operator  $S_p(\mathbf{v})$  is the covariant derivative of a normal vector field at point  $\mathbf{p}$  in the direction  $\mathbf{v}$ . In other words, the shape operator provides information about the direction change of the normals to a surface when moving over the surface in a certain direction  $\mathbf{v}$ . Consider a surface  $M \subset \mathcal{R}^3$  and its normal vector field  $N$  (Figure 26). Figure 27 shows a unit normal vector  $N(\mathbf{p})$  in point  $\mathbf{p}$  on this surface with its corresponding tangent plane. Imagine following the surface  $M$  in the direction of  $\mathbf{v}$ , the normal vector field is expected to rotate to the right and slightly forward. The shape operator representing this rotation is shown in Figure 27.

Because the shape operator provides information about the change in direction of the normal vectors, the rate of change of the shape operator will provide information about the strongness of the curve, i.e. the normal curvature. Hereby it is possible to compute the normal curvature in direction  $\mathbf{v}$ , which can be in any direction on the tangent plane: rotating from 0 to 360° around the normal vector (Figure 28). The minimal and maximal normal curvature  $\kappa_1$  and  $\kappa_2$  of a surface in a point  $\mathbf{p}$  and their corresponding directions  $\mathbf{v}_1$  and  $\mathbf{v}_2$  are called the principal curvatures and principal directions respectively. Figure 29 shows the curves corresponding to the principal curvatures at the centre of a saddle shaped surface.

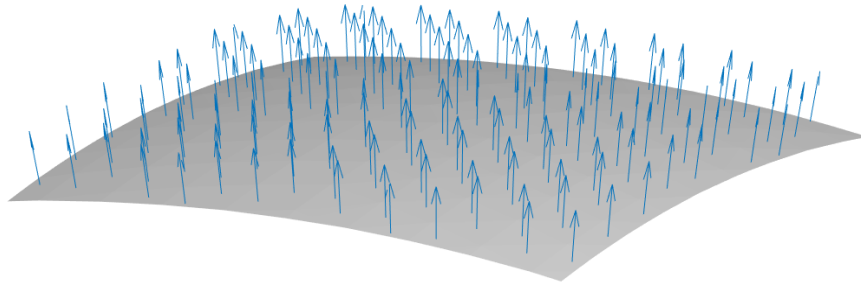


Figure 26 Surface  $M$  with unit vector field  $N$

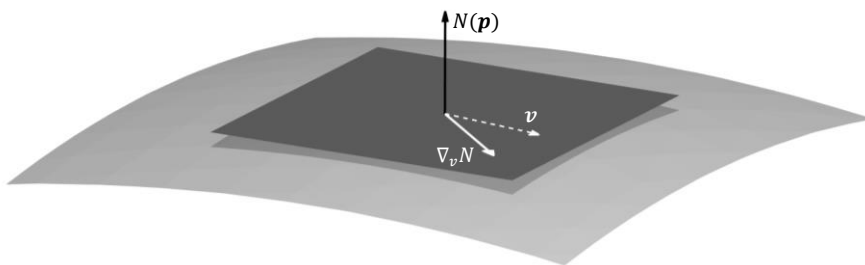


Figure 27 Surface  $M$  with the unit vector  $N(\mathbf{p})$  and the corresponding tangent plane and the shape operator  $\nabla_{\mathbf{v}} N$  in the direction of  $\mathbf{v}$

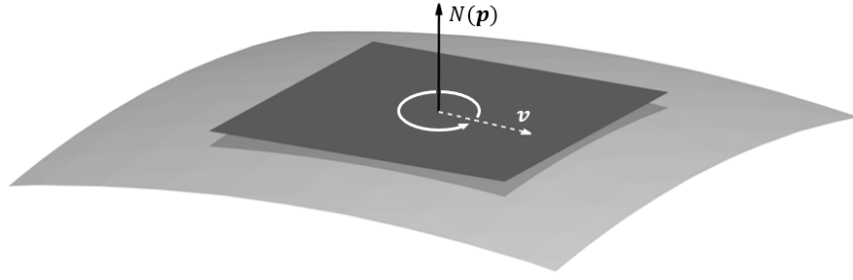


Figure 28 Surface  $M$  with the unit normal vector  $N(\mathbf{p})$  and the corresponding tangent plane illustrating that direction  $\mathbf{v}$  can be any direction, rotating from 0 to 360 degrees around  $N(\mathbf{p})$

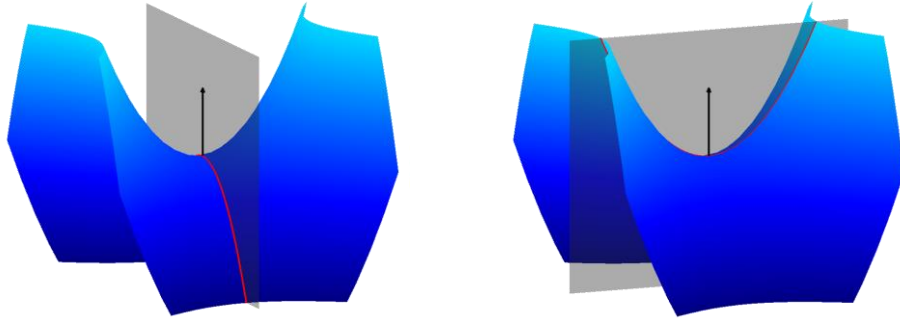


Figure 29 The principle curvatures of a saddle surface

The principal curvatures (and all other normal curvatures) only provide information of the surface in specific directions. Therefore, not the principal curvatures but the mean and Gaussian curvatures are used to describe a surface. Their definitions are:

$$K_M = \frac{\kappa_1 + \kappa_2}{2} \quad (9)$$

$$K_G = \kappa_1 \kappa_2 \quad (10)$$

$K_M$  is the mean curvature and  $K_G$  the Gaussian curvature

O'Neill[48] clearly describes that the Gaussian curvature has the most important geometrical meaning; it directly provides information about the shape of a surface and is independent on the choice of unit normal vectors. They elaborate the four situations in which the Gaussian curvature is positive, negative or zero.

- I.  $K_G(\mathbf{p}) > 0 \text{ m}^{-1}$ , both  $\kappa_1(\mathbf{p})$  and  $\kappa_2(\mathbf{p})$  are positive; the surface is paraboloid
- II.  $K_G(\mathbf{p}) < 0 \text{ m}^{-1}$ ,  $\kappa_1(\mathbf{p})$  or  $\kappa_2(\mathbf{p})$  is negative; the surface is saddle-shaped
- III.  $K_G(\mathbf{p}) = 0 \text{ m}^{-1}$ ,  $\kappa_1(\mathbf{p})$  and  $\kappa_2(\mathbf{p})$  is zero; the surface is a plane
- IV.  $K_G(\mathbf{p}) = 0 \text{ m}^{-1}$ ,  $\kappa_1(\mathbf{p})$  or  $\kappa_2(\mathbf{p})$  is zero; the surface is trough-shaped

Even though the mean curvature has less important geometrical meaning, it is more stable because it is an average instead of the product of the principle curvatures.

## 7.5 FACE QUADRICS

A quadric is a second degree polynomial and can have many different shapes, depending on its coefficients[49]. Often described forms of quadrics are the sphere, ellipsoid, cylinder, cone, saddle, paraboloid and the hyperboloid. This section describes how a quadric surface is defined, that solve the least square fitting problem for a plane.

The distance error metric of a quadric surface to a face is defined as the squared distance of an arbitrary point  $\mathbf{p}$  at the quadric surface to the plane  $P \subset \mathcal{R}^3$  in which the face is present. The distance of a point  $\mathbf{p} = (p_1, p_2, p_3, 1)$  to the plane  $P$  is given by the length of the projection of the vector  $\mathbf{p}$  on the plane:

$$D = \mathbf{n}^T \cdot \mathbf{p} + d$$

Therefore, the squared distance gives:

$$(\mathbf{n}^T \cdot \mathbf{p} + d)^2 = \mathbf{p}^T (\mathbf{n}\mathbf{n}^T) \mathbf{p} + 2d\mathbf{n}^T \mathbf{p} + d^2 \quad (11)$$

Hereby, the unit normal vector of a face  $\mathbf{n}$  is given by:

$$\mathbf{n} = \frac{(\mathbf{v}_1 - \mathbf{v}_2) \times (\mathbf{v}_1 - \mathbf{v}_3)}{\|(\mathbf{v}_1 - \mathbf{v}_2) \times (\mathbf{v}_1 - \mathbf{v}_3)\|}$$

$\mathbf{v}_1, \mathbf{v}_2$  and  $\mathbf{v}_3$  are the three vertices of the face. Variable  $d$  is defined as:

$$d = -\mathbf{n}^T \mathbf{p}$$

Equation (11) can be rewritten so that the error metric is written in the quadric form:

$$\mathbf{p}^T Q \mathbf{p} = \epsilon \quad (12)$$

With  $Q$ :

$$Q = (\mathbf{A}, \mathbf{b}, c) = ((\mathbf{n}\mathbf{n}^T), (d\mathbf{n}), (d^2))$$

$Q$  can also be represented as the homogeneous symmetrical matrix:

$$Q(\mathbf{p}) = \begin{bmatrix} \boxed{\mathbf{A}} & \boxed{\mathbf{b}} \\ \boxed{\mathbf{b}^T} & \boxed{d^2} \end{bmatrix} = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ q_{14} & q_{24} & q_{34} & q_{44} \end{bmatrix}$$

By filling in (11),  $\mathbf{A}$  and  $\mathbf{b}$  can be defined as:

$$\mathbf{A} = \mathbf{n}\mathbf{n}^T = [n_1 \quad n_2 \quad n_3] \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix} = \begin{bmatrix} (n_1)^2 & n_1 n_2 & n_1 n_3 \\ n_1 n_2 & (n_2)^2 & n_2 n_3 \\ n_1 n_3 & n_2 n_3 & (n_3)^2 \end{bmatrix}$$

$$\mathbf{b} = d\mathbf{n} = d([n_1 \quad n_2 \quad n_3]) = [dn_1 \quad dn_2 \quad dn_3]$$

Then, the homogeneous matrix for the quadric  $Q(\mathbf{p})$  is:

$$Q(\mathbf{p}) = \begin{bmatrix} (n_1)^2 & n_1 n_2 & n_1 n_3 & n_1 d \\ n_1 n_2 & (n_2)^2 & n_2 n_3 & n_2 d \\ n_1 n_3 & n_2 n_3 & (n_3)^2 & n_3 d \\ n_1 d & n_2 d & n_3 d & d^2 \end{bmatrix} \quad (13)$$

And thus:

- $\mathbf{A}$  is a symmetrical  $3 \times 3$  matrix
- $\mathbf{b}$  is a  $3 \times 1$  column vector
- $d^2$  is a constant.

In conclusion, the quadric functional  $Q$  is represented in a symmetrical  $4 \times 4$  matrix and stores 10 coefficients.

Equation (12) is very useful because solving  $\mathbf{p}^T Q \mathbf{p} = \varepsilon$  gives a set of points that is located at  $\varepsilon$  distance from the original surface. Note that to find the quadric surface that approximates the plane in which the face is present, a set of points should be found for which the distance of the quadric surface to this plane should be zero. In other words, if the distance is zero the quadric is congruent to the plane in which the face is present. Considering this: the least square problem can be defined as:

$$\mathbf{p}^T Q \mathbf{p} = [x \quad y \quad z \quad 1] \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ q_{14} & q_{24} & q_{34} & q_{44} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} =$$

$$= q_{11}x^2 + 2q_{12}xy + 2q_{13}xz + 2q_{14}x + q_{22}y^2 + 2q_{23}yz + 2q_{24}y + q_{33}z^2 + 2q_{34}z + q_{44} = 0$$

When the coordinates of the three vertices of a face are known, the general equation of the plane in which the face is present can be determined:

$$ax + by + cz + d = 0$$

This equation provides normal vector  $\mathbf{n} = [a \quad b \quad c]$  and  $d$  and gives the quadric functional defined in equation (13). Thus, for each face (if the vertex coordinates are known) the quadric functional is:

$$Q = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix}$$

## 7.6 CURVATURE FORMULAS FOR IMPLICIT FUNCTIONS

Goldman[36] describes the formulas for curvature from implicit formulas. The formulas are given in the following textboxes:

$\nabla F = \left( \frac{\partial F}{\partial x} \quad \frac{\partial F}{\partial y} \quad \frac{\partial F}{\partial z} \right)$	<i>gradient</i>
$H(F) = \begin{bmatrix} \frac{\partial^2 F}{\partial x^2} & \frac{\partial^2 F}{\partial x \partial y} & \frac{\partial^2 F}{\partial x \partial z} \\ \frac{\partial^2 F}{\partial y \partial x} & \frac{\partial^2 F}{\partial y^2} & \frac{\partial^2 F}{\partial y \partial z} \\ \frac{\partial^2 F}{\partial z \partial x} & \frac{\partial^2 F}{\partial z \partial y} & \frac{\partial^2 F}{\partial z^2} \end{bmatrix}$	<i>Hessian</i>
$H^*(F) = \begin{bmatrix} \frac{\partial^2 F}{\partial y^2} \frac{\partial^2 F}{\partial z^2} - \frac{\partial^2 F}{\partial y \partial z} \frac{\partial^2 F}{\partial z \partial y} & \frac{\partial^2 F}{\partial y \partial z} \frac{\partial^2 F}{\partial z \partial x} - \frac{\partial^2 F}{\partial y \partial x} \frac{\partial^2 F}{\partial z^2} & \frac{\partial^2 F}{\partial y \partial x} \frac{\partial^2 F}{\partial z \partial y} - \frac{\partial^2 F}{\partial y^2} \frac{\partial^2 F}{\partial z \partial x} \\ \frac{\partial^2 F}{\partial x \partial z} \frac{\partial^2 F}{\partial z \partial y} - \frac{\partial^2 F}{\partial x \partial y} \frac{\partial^2 F}{\partial z^2} & \frac{\partial^2 F}{\partial x^2} \frac{\partial^2 F}{\partial z^2} - \frac{\partial^2 F}{\partial x \partial z} \frac{\partial^2 F}{\partial z \partial x} & \frac{\partial^2 F}{\partial x \partial y} \frac{\partial^2 F}{\partial z \partial x} - \frac{\partial^2 F}{\partial x^2} \frac{\partial^2 F}{\partial z \partial y} \\ \frac{\partial^2 F}{\partial x \partial y} \frac{\partial^2 F}{\partial y \partial z} - \frac{\partial^2 F}{\partial x \partial z} \frac{\partial^2 F}{\partial y^2} & \frac{\partial^2 F}{\partial y \partial x} \frac{\partial^2 F}{\partial x \partial z} - \frac{\partial^2 F}{\partial x^2} \frac{\partial^2 F}{\partial y \partial z} & \frac{\partial^2 F}{\partial x^2} \frac{\partial^2 F}{\partial y^2} - \frac{\partial^2 F}{\partial x \partial y} \frac{\partial^2 F}{\partial y \partial x} \end{bmatrix}$	<i>adjoint Hessian</i>
$K_G = \frac{\nabla F * H^*(F) * \nabla F^T}{2 \ \nabla F\ ^4}$	<i>Gaussian curvature</i>
$K_M = \frac{\nabla F * H(F) * \nabla F^T -  \nabla F ^2 \text{Trace}(H)}{2 \ \nabla F\ ^3}$	<i>mean curvature</i>
$k_1, k_2 = K_M \pm \sqrt{K_M^2 - K_G}$	<i>principal curvatures</i>

## 8 ACKNOWLEDGEMENTS

In September 2014 I started my first M2 internship at the department of Neurosurgery of the Radboudumc, in which I got infected by Hans Delye's enthusiasm, ambitiousness and hardworking spirit. From that moment on it was clear that I would put maximal effort into this project. There was no time that year in which I was not working on or thinking/talking about this fantastic, gratifying, but also challenging project. During my other internships I went to several meetings on this project, published my article, went to the Erasmus University Medical Centre to initiate our collaboration and I worked further on the project as preparation for the 10<sup>th</sup> European Congress for Craniofacial Surgery in Gothenburg. Hans pushed me to the fullest: he often brought me outside my comfort zone with as goal to broaden my knowledge or to do just a little bit more than I thought was possible, so that I took all the advantages that I could get. On the other hand, he was a great support when there were troubles with my health and when I was struggling to keep positive. In some way he could always read my expression correctly, even when I did not want him to, and he always knew the right thing to say. I would like to thank Hans for this great experience and for all the opportunities that he provided. It was a great pleasure to work for him and I am grateful for all his lessons, criticisms and positive encouragements.

Another person that I have worked with in the past 2 years is Ferdi van der Heijden. He was my teacher in all the imaging analysis courses and his lessons inspired me to continue on this topic during my internships. Ferdi was my technical supervisor on 3 projects, and I really enjoyed his enthusiasm and willingness to help me on my technical challenges. He was always very interested in my work, but also taught me to be critical about my work. I am very impressed by the amount of great work that he does and I would like to thank him for the many inspirational conversations that we had.

Since 2014 the neurosurgery research department has expanded substantially. In the past 2 years I have met many new researchers who have made working at the 5<sup>th</sup> floor way less humdrum. While in September 2014 the lights turned off very often (because there was no movement), there is now always someone to talk to and grab a cup of coffee with. One of the researchers who has made my work more joyful was Guido de Jong. We were on the same project and we have worked together through the entire year. For some we seemed inseparable (even after work) and, even though we have had some struggles due to our stubbornness, I regret that I will have to leave the department and our paths will separate. I would like to thank Guido for being like a big brother to me, showing me what to do, but also what not to do. I have really enjoyed working with him, our weekly dinners and movie nights.

Furthermore, there are some persons I would like to thank because they have helped me many times during my research. First, I would like to thank Marleen Groenier and Marieke Hofman for always being there during hard times and for always having the patience to listen to my long-winded stories. Also, I would like to thank Ruud van Damme for taking the effort to help me with my mathematical challenges. I have really enjoyed our conversations which have really encouraged my interest in mathematics. And finally, I would like to thank Ivo Broeders for being my chairman and arranging all logistics through the year.

And last but not least, I would like to thank Wim Hoek, for being the most important person in my world, for all his patience, his capacity to always rationalise and for being the big joker who is always able to make me laugh. I am really looking forward to our future adventures.