

# **Segmentation of Craniofacial Anatomy using Deformable Models**

Ditte Høvenhoff Hald s062275  
Rikke Nørgaard Wiener s062239

Kongens Lyngby 2012  
IMM-Msc-2012-3

Technical University of Denmark  
Informatics and Mathematical Modelling  
Building 321, DK-2800 Kongens Lyngby, Denmark  
Phone +45 45253351, Fax +45 45882673  
[reception@imm.dtu.dk](mailto:reception@imm.dtu.dk)  
[www.imm.dtu.dk](http://www.imm.dtu.dk)

# Summary

---

This thesis presents methods for assessing and estimating the intracranial volume in children with unicoronal synostosis, on the basis of 3D CT images. Obtaining such a measurements provides the opportunity to compare the intracranial volume in these children with *i.a.* to normal data, in order to comment on possible deviations between such two groups. Furthermore, it provides a tool for a possible surgery evaluation, given as pre- and post measurements.

To solve this problem, two deformable models have been chosen; an image registration, and a graph cut based algorithm. The image registration model transforms a template image into a given reference, especially by means of a B-spline transformation. By extracting the transformation parameters and applying them to a manual segmented mask, derived from the template image, an estimation of the volume of interest is obtained. The graph cut model takes a gradient based approach to the problem, and based on construction of a weighted graph, consisting of nodes and edges, it finds the optimum cut *i.e.* the segmentation. The true challenge for this model, lies within the construction of this framework, *i.e.* establishing the proper weights and neighborhood connections.

The performance of the two models is validated on the basis of a voxel deviation found between the model based segmentation and a semi-automatic segmentation, performed with manual editing. Furthermore, visual interpretation of the segmentation surfaces has been performed. Both models showed great results, and presented a good foundation for further applicability.



# Resumé

---

Denne afhandling præsenterer to metoder til at vurdere og estimere det intrakranielle volumen hos børn med unicoronal synostosis. Begge metoder er baseret på 3D CT billeder. Et mål for det intrakranielle volumen, vil give mulighed for at sammenligne volumener hos børn med craniosynostosis med textit i.a. normal data med henblik på, at vurdere eventuelle afvigelser mellem sådanne to grupper. Desuden vil det give mulighed for at evaluere effekten af operationer ud fra en før og efter måling.

For at løse dette problem, er to deformerbare modeller valgt; en billederegistreringsmodel, samt en graf cut baseret algoritme. Billedregistreringsmodellen omdanner et templatebillede til en valgt reference. Denne omdannelse er her fortaget primært ud fra en B-spline transformation. Ved at udtage de fundne transformations parametre og påføre dem til en manuel segmenteret maske, fundet udfra templatebilledet, opnås et estimat af det intrakranielle volumen af referencen. Graf cut modellen benytter en gradientbaseret metode, samt en konstruktion af en vægtet graf bestående af knuder med dertilhørende vægte, for at finde det optimale cut, der vil definere segmenteringen. Selve opbygningen af denne konstruktion, er den store udfordring med denne model.

De to modeller er valideret på grundlag af en udregnet voxel afvigelse, fundet mellem den modelbaseret segmentering og en semi-automatisk segmentering, hvor den sidstnævnte er udarbejdet under indflydelse af manuel editering. Endvidere er segmenteringsresultaterne evalueret på baggrund af en både 2D og 3D visualisering. Begge modeller viste gode resultater, og er set at udgøre et godt grundlag for fremtidig anvendelighed indenfor dette felt.



# Preface

---

This master thesis was carried out in the period from the 1st of September 2011 to the 29th of February 2012, at DTU Informatics, at the Technical University of Denmark in partial fulfillment of the requirements for acquiring the M.Sc. degree in engineering in Medicine and Technology. The thesis treats the topic of intracranial volume estimations in children with unicoronal synostosis, and the project thereby combines clinical prospective, with theories of medical image analysis.

The work was performed in a close collaboration between DTU Informatics, at the Technical University of Denmark and the 3D Craniofacial Image Research Laboratory (School of Dentistry, University of Copenhagen; Copenhagen University Hospital Rigshospitalet; and DTU Informatics, Technical University of Denmark), Copenhagen, Denmark (3D-Laboratory).

The project was supervised by Professor, PhD, Rasmus Larsen from DTU Informatics, Research Engineer, PhD, Tron Darvann, Professor, PhD, dr.odont. Sven Kreiborg and Associate Professor, PhD, Nuno Hermann, where the three last-mentioned all are affiliated with the 3D-Laboratory.

Lyngby, February 2012

Ditte Høvenhoff Hald



Rikke Nørgaard Wiener





# Acknowledgements

---

Great thanks goes to all people who have been involved and provided their assistance throughout the preparation of this thesis. A special thanks to our main supervisor Professor, PhD, Rasmus Larsen for support and assistance during the entire process, and for providing new approaches when such were necessary. In that connection also thanks to Professor Knut Conradsen, for great feedback during the Friday meeting sessions.

Also thanks to all at the 3D-Laboratory for their guidance and support, and for providing the data needed for this thesis. A special thanks to Research Engineer, PhD Tron Darvann for assisting with the manual segmentations, answering question and for providing constructive feedback throughout the entire period. Further thanks are given to Research Engineer, MSc. Eng., Per Larsen and Associate Professor, PhD, Nuno Hermann, for contributing to rewarding discussions, and providing help when such were needed. Furthermore, a great thanks to Professor, PhD, dr.odont., Sven Kreiborg, for providing information and feedback regarding the clinical aspects for this thesis.

Special thanks also goes to PhD-student Mark Lyksborg, for patiently assisting with some of the underlying theoretical understanding. In the same connection also a great thanks to MSc. Eng., PhD Hildur Ólafsdóttir, for helpful conversations regarding the image registration software.

Finally, a special thanks to Research Engineer, PhD Tron Darvann, Professor, PhD, Rasmus Larsen, Professor, PhD, dr.odont., Sven Kreiborg, Martin Kristensson and Bo Thøgersen for feedback and proofreading.



# Abbreviations and Explanations

---

## List of Abbreviations and Acronyms

BCS	Bilateral Coronal Synostosis
BET	Brain Extraction Tool
CS	Craniosynostosis
CT	Computed Tomography
FAIR	Flexible Algorithms for Image Registration
fMRI	Functional Magnetic Resonance Imaging
FSL	Functional MR Imaging of the Brain Software Library
GC	Graph Cut
HU	Hounsfield Unit
ICP	Intracranial Pressure
ICV	Intracranial Volume
IR	Image Registration
LUCS	Left sided Unilateral Coronal Synostosis
MR	Magnetic Resonance
MS	Metopic Synostosis
PAT	Principal Axis Transformation

RUCS	Right sided Unilateral Coronal Synostosis
SPM	Statistical Parametric Mapping
SS	Sagittal Synostosis
SSD	Sum-of-Squared Differences
UCS	Unicoronal Synostosis

## Symbol Definitions

$\alpha$	Step size
$A$	Affine transformation parameters
$b(x_i)$	B-spline
$\beta$	Angle between two principal axes
$\beta_1$	Constant controlling the gradients influence
$\delta^d$	Physical voxel size in each dimension
$D$	Image dimensions, $d = 1 \dots D$
$Df$	The degree of the polynomial basis function
$D, D_{SSD}$	Dissimilarity measure based on the sum-of-squared-differences
$E$	Edges for the graph
$G$	A weighted graph
$h$	The descent direction
$I$	Image of concern
$J(w)$	The Jacobian
$\lambda$	Weight parameter for the regularization
$L_{level}$	The levels of the image pyramid. $L_0$ corresponds to the original image and the higher the level the coarser the image.
$m$	Image size in voxels, $m = [m^1, \dots, m^D]$
$N$	Number of voxels in the image, $i = 1 \dots N$
$\Omega$	Physical interval covered by the image, $\Omega = (\omega^1, \omega^2) \times (\omega^{2d-1}, \omega^{2d})$ for $d = 2 \dots D$
$P_d$	Knot points in each direction, $p_d = 1 \dots P_d$
$\phi$	Rotation in the $z$ -plane
$\varphi(\alpha)$	"Line" function
$q$	Basic function
$Q$	The staking of the basic functions
$r$	Radius from the center
$\hat{\mathbf{r}}$	radial unit vector
$r(w)$	Residual function
$rd$	Radial density
$R_1$	Principal axis of the reference
$\mathcal{R}(\mathbf{X})$	The reference image
$S(w)$	Regularization term
$s, S$	Source of a graph
$sca$	Ratio between two dimensions
$ss$	Amount of sampling points in a sampling sphere
$t, T$	Sink of a graph

---

$\theta$	Rotation in the $x - y$ -plane
$T_1$	Principal axis of the template
$\mathbf{t}$	Translation parameters
$\mathcal{T}(\mathbf{Y})$	Transformed template image
$\nabla T$	Image gradients in all directions
$u$	A node in the graph
$u(\mathbf{X}; \cdot)$	The deformation part (also $U$ )
$U1$	Skilled user
$U2$	Untrained user
$V$	Nodes of the graph
$\mathbf{w}, w$	Weight parameters
$x$	The first image direction, corresponds to $\mathbf{x}^1$
$\mathbf{x}^d$	The image directions
$\mathbf{x}_i$	One cell (or voxel) in the image
$\mathbf{X}$	The entire image grid
$\xi^d$	Knot sequence in the dimension $d$
$y$	The second image direction, corresponds to $\mathbf{x}^2$
$\mathbf{y}_i$	The transformed version of $\mathbf{x}_i$
$\mathbf{Y}(\mathbf{X}; \cdot)$	Transformed version of the entire grid $\mathbf{X}$
$z$	The third image direction, corresponds to $\mathbf{x}^3$





# Contents

---

<b>Summary</b>	<b>i</b>
<b>Resumé</b>	<b>iii</b>
<b>Preface</b>	<b>v</b>
<b>Acknowledgements</b>	<b>vii</b>
<b>Abbreviations and Explanations</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Project Description . . . . .	1
1.2 Previous Work . . . . .	4
1.3 Thesis Overview . . . . .	5
<b>2 Cranial Growth</b>	<b>7</b>
2.1 Normal Cranial Growth . . . . .	7
2.2 Craniosynostosis . . . . .	8
<b>3 Volume Estimations up until Today</b>	<b>15</b>
3.1 Historical Flashback . . . . .	15
3.2 Current Methods for Automatic Segmentation . . . . .	17
<b>4 Data Information and Definitions</b>	<b>23</b>
4.1 Dimension Terms . . . . .	25
4.2 Multilevel . . . . .	25
<b>5 Manual Segmentation</b>	<b>29</b>
5.1 Landmarker . . . . .	29

5.2 Analyze . . . . .	31
<b>6 Image Registration</b>	<b>35</b>
6.1 Image Representation . . . . .	36
6.2 Transformation . . . . .	38
6.3 Minimization of the Dissimilarity Measure . . . . .	44
6.4 Regularization . . . . .	48
6.5 Multilevel Image Registration . . . . .	48
<b>7 Graph Cut</b>	<b>51</b>
7.1 Background . . . . .	51
7.2 The Min-Cut/Max-Flow Algorithm . . . . .	54
7.3 Edge weights . . . . .	55
<b>8 Preprocessing</b>	<b>59</b>
8.1 Standard Orientation . . . . .	59
8.2 Removal of Noise Components . . . . .	60
8.3 Optimizing the Initial Condition . . . . .	63
<b>9 Implementation and Optimization - 3D Image Registration Based Model</b>	<b>69</b>
9.1 Implementation . . . . .	70
9.2 Optimization . . . . .	72
<b>10 Implementation and Optimization - 3D Graph Based Model</b>	<b>79</b>
10.1 Preprocessing . . . . .	80
10.2 Construction of the Source and Sink . . . . .	81
10.3 Radial Gradient . . . . .	82
10.4 Spherical Resampling . . . . .	83
10.5 Edge Weights . . . . .	87
10.6 Node Connections and Inter Column Weights . . . . .	88
10.7 Conversion Back to Cartesian Coordinates . . . . .	90
<b>11 Results</b>	<b>93</b>
11.1 Manual Segmentation . . . . .	93
11.2 Model Based Segmentations . . . . .	96
11.3 Model Comparison . . . . .	108
<b>12 Discussion</b>	<b>111</b>
12.1 Manual Segmentation . . . . .	111
12.2 Image Registration . . . . .	113
12.3 Graph cut . . . . .	114
12.4 Model Comparison . . . . .	117
<b>13 Future Work and Perspectives</b>	<b>119</b>

## CONTENTS

---

xvii

14 Conclusion	123
A Parameter test for optimizing the GC model	125
B Results - Illustrations	129



# Introduction

---

## 1.1 Project Description

The purpose of this thesis work is to implement an automatic segmentation model for estimation of the intracranial volume in children with craniosynostosis. Craniosynostosis is a condition where one or more sutures between the cranial bones have fused at a too early state, restricting skull growth and thereby leading to cranial dysmorphology. Moreover, it possibly entails a reduced intracranial volume, with the risk of development of increased intracranial pressure. In the Western world, almost all children born with craniosynostosis undergo surgery, in order to correct and adjust the deformity, and to secure adequate space for the growing bones. Estimating the intracranial volume in these children is from a medical point of view of high interest, in order to see how or if their volumes differ from normal values. Currently, applications for intracranial volume estimation are available; however, most of these are designed for MR images. MR scans of small children will require that they are under anesthesia, due to the noise level and long image acquisition time. Therefore, MR imaging is not always the preferred scan modality. Furthermore, most surgeries are based on a detailed 3D CT scan, where it is possible to obtain a good overview of the cranial bones, due to the contrast in this scan modality. The goal is therefore to find and implement a model that can be applied to CT scans. In the clinical aspect, the goal is to have a high speed intracranial volume application, with high precision

and acceptable costs.

Two approaches have been chosen for investigation in this thesis, both chosen on the hypothesis of their ability to overcome the lack of cranial bone information. The first is an image registration algorithm, where the volume is estimated by registering the image to a template. Together with the template image, a manual segmentation of its intracranial volume, referred to as volume mask, is available. The registration is performed between the template image and the image of interest, by applying two 3D transformation types; an affine followed by a B-spline transformation, to the template image. The transformations allow the template image to deform, such that a high similarity between the transformed template and the image of interest is established. The success of the transformation is evaluated based on a cost function, defined as the intensity differences in the two images. When a satisfying registration is found, the transformation parameters are extracted and applied to the intracranial volume mask obtained from the manual template segmentation, *i.e.* the mask undergoes the same deformation as the template image. In this way a mask corresponding to the intracranial volume of the image of interest is constructed. However, this model has the disadvantage of needing a template image, which should have a good and trustworthy volume estimate. Furthermore, by using such a model, the final result will unfortunately be biased against the chosen template image.

In order to mitigate these issues, a graph cut model, is investigated. This is a model, which over the last couple of years, has been seen as an increasingly popular technique for medical image segmentation. The model is an attractive model for surface segmentation, due to its guaranteed convergence to the global minimum. However, the true challenge is to construct this framework. Graph cut is based upon the construction of a weighted graph, which is built based on a set of nodes, corresponding to voxels, and edges, specifying the cost of a connection between a neighborhood of nodes. In order to construct this setup, the segmentation problem is transformed from its original image space into a terrain-like surface, hence a graph space, by performing a spherical resampling. Additional nodes are added to the graph setup, referred to as source and sink, specifying each side of a sought segmentation. The neighborhood connection is then defined together with the edge costs, which consist of both intra and inter column edge costs. The intra column costs are based on the image gradient, while the inter column cost is a constant, seen as a separation penalty restricting a smoother surface, hence a smoother cut. A minimum cut/maximum flow algorithm is used in order to find the "cheapest" cut of the graph. In order to evaluate the cut, the segmentation, the problem is transformed back from the spherical coordinates to Cartesian image coordinates.

The idea for the image registration model is found in [26], where the approach was used to compute mouse atlases, for analyzing craniofacial dysmorphology in

Crouzon mice, on the basis of 3D micro CT scans. In 2008, an image registration software package was implemented by Martin Vester-Christensen, which kindly was passed on to be used in this project. The idea for the graph cut model is derived from [37], where the model was used for segmenting the respiratory tree based on a 3D CT dataset. The algorithm uses the minimum cut/maximum algorithm introduced in [7], which builds on the Ford-Fulkersons augmenting path.

## Motivation

Information concerning size, shape and volume of the brain and skull has been of high interest for many years. First of all, it provides basic anatomical knowledge, but it also provides the opportunity to examine how *e.g.* genetic and environment affect the cranial growth and development. Medically, knowledge about the cranial growth and development provides the opportunity to analyze crania with abnormal growth patterns and malformations, as *e.g.* found in craniosynostosis. As prior mentioned, an estimate of the intracranial volume in these children is, from a medical point of view, of high interest. Both to form an impression of whether the volume in these children differs from the values found in unaffected children or in between the different classes of craniosynostosis, and finally to be able to evaluate the effect of surgery.

Data processed in this thesis, are all 3D CT image scans from children with unicoronal synostosis (UCS) . The scans have been obtained from children between approximately 6 to 18 months of age.

## Main Objective

The main aim is an implementation and investigation of two different 3D segmentation models, which can establish a satisfactory estimation of the intracranial volume from CT images in children with craniosynostosis. The evaluation of a satisfactory estimate is based on a comparison between a semi-manual and the model based segmentation.

## Part Objectives

- Compare the performance of the two segmentation models.
- Evaluate the veracity of the manual segmentations.

- Compare the obtained segmented volumes with normative data.

## External collaborators

The project was performed in collaboration with Research Engineer, MSc., PhD, Tron Darvann, Professor, dr. odont, PhD, Sven Kreiborg, Associate Professor, PhD, Nuno Hermann and Research Engineer, MSc. Eng., Per Larsen, who are affiliated with the 3D Craniofacial Image Research Laboratory (School of Dentistry, University of Copenhagen; Copenhagen University Hospital Rigshospitalet; and DTU Informatics, Technical University of Denmark)(3D-Laboratory), Copenhagen, Denmark. They have provided data and assisted in discussion throughout this thesis.

## 1.2 Previous Work

Estimation of the intracranial volume in children with craniosynostosis has been of interest in many years, and different segmentation approaches have been tried, in order to reach a satisfying result. Within the field of MR imaging automatic segmentation forms exist, but these segmentation forms are not optimal for intracranial volume estimation, in craniosynostotic cases. In order to get the best possible evaluation of the cranial bones, CT scans are preferred. Furthermore, CT scans are the most common used image modality to obtain information for diagnosis and surgery planning, and are therefore, the preferable basis for volume estimation. Unfortunately, the methods are still limited within the field of CT imaging. Today, the most used method is a semi-automated segmentation process, based on a threshold and a seed-expansion technique. However, this method needs manual editing in the various foramina in the skull base, but more critically, manual editing is also necessary in regions where craniosynostosis or late suture fusion have caused gaps between the cranial bone plates. A number of studies [4, 41, 42] have been performed with this estimation method. The physician Sgouros has performed research studies regarding volume estimation in both normal children and children with craniosynostosis, [41, 42]. The study included 68 healthy children, in the age range from 1 to 187 months of age, and 84 children in the age range from 1 to 84 months of age with craniosynostosis. The children were affected with different types of craniosynostosis, both non-syndromic cases, including sagittal, unicoronal, bicoronal and metopic synostosis, but also a small number of syndromic cases were included. In this thesis the volume estimations [41] are extracted and used as a reference foundation for a normal population.

## 1.3 Thesis Overview

Throughout the work of this thesis two different models have been processed. Initially, the main focus should have been on the graph cut model, and the image registration model just utilized for comparison foundation. Unfortunately, an old version of the image registration software was first distributed, which required significantly more time and deeper examination of the software, than first expected. Both models have therefore received the same amount of focus on their theoretical parts. Subsequently, the main focus is leveled at the implementation and optimization of the graph cut model. The theory, implementation and results of the two models will be given concurrent throughout the paper. However, first an introduction to the anatomy and physiology behind normal cranial growth is provided in Chapter 2. This is in the same chapter followed by the definition of different types of craniosynostosis, together with the clinical aspects concerning the condition. In Chapter 3 a glance is taken at the volume estimation methods tried out through the last couple of decades up until today. Today's methods are evaluated and a pros and cons comparison between CT and MR imaging techniques concerning craniosynostosis is conducted. Chapter 4 provides the reader with data information and a description of how data dimensions are referred to throughout the thesis.

Both methods are validated against a semi-manually performed segmentation, which is described in Chapter 5. In Chapter 6 and Chapter 7 the theory behind the two models is presented. Chapter 6 describes the image registration process, including the used transformation types, the cost function and the optimization process. In the final section a short description of the multilevel approach taken for this model, is given. Chapter 7 provides information of the graph cut model, including a description of the minimum cut/maximum flow algorithm and the constellation of neighborliness and edge weights.

From Chapter 8, a description of the actual processing of data starts taking place. Chapter 8 describes the preprocessing, including standard orientation and noise removal, which is common for both methods. Afterwards the implementation and optimization of the image registration, and graph cut model are given in Chapter 9 and Chapter 10, respectively. Chapter 11 presents the results obtained from the two models, while Chapter 12 provides a discussion of these, together with an evaluation of the advantages and limitations of the two models. Chapter 13 provides thoughts and ideas for future work and prospects for this thesis, while a final conclusion is found in Chapter 14.



# Cranial Growth

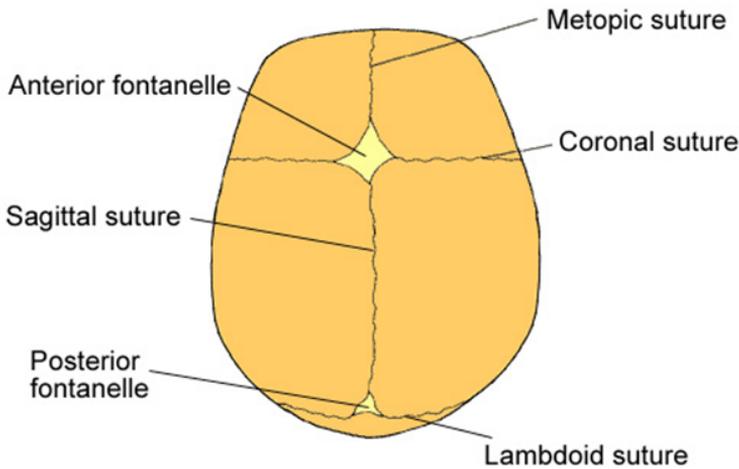
---

The aim of this thesis is to implement a segmentation model, which can give a satisfactory estimation of the intracranial volume. The data processed in this thesis are obtained from children with craniosynostosis, so this chapter provides an introduction to the condition. However, first a short description of the normal cranial growth pattern is given in Section 2.1. Hereafter, the different types of craniosynostosis is described in Section 2.2, comprising clinical manifestation and treatment consisting of surgery.

## 2.1 Normal Cranial Growth

When we are born, the cranium consists of primarily five separate bones; paired frontals bones, paired parietal bones and the occipital bone. These bones are held together by strong elastic tissues called sutures, and two "soft spots" called fontanelles, the anterior and posterior, respectively, [34]. The anatomy is outlined in Figure 2.1.

During birth the sutures and fontanelles provide flexibility to the skull, allowing the cranial bones to overlap, hence enabling the head of the child to pass through the birth canal. After birth, the sutures remain open, where they serve as growth zones between the cranial bones, allowing the cranium to expand to follow the



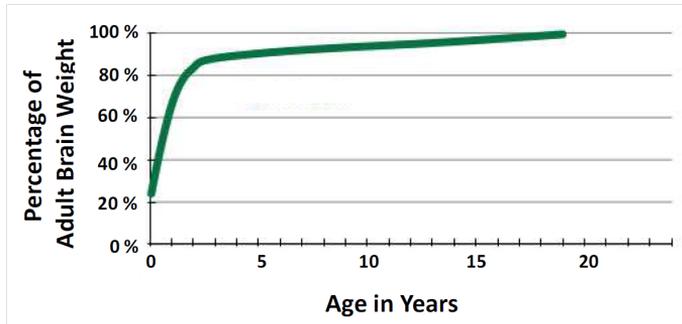
**Figure 2.1:** The normal skull of a newborn with identification of the sutures and fontanelles. From [2].

growing brain. Cranial growth results from increased osteoblast activity at the edges of the cranial bones. Sutures are, therefore, not seen as profound growth zones, but they provide a place for bone deposition, [39, 19].

The growth of the brain is not linear during its time of development, but it can, however, be seen partly linear during given time periods, *e.g.* during the first couple of years. In this period, the brain is seen to have its most significant expansion, as seen in Figure 2.2, which results in an increase of the brain size from approximately  $60 \text{ cm}^3$  and  $90 \text{ cm}^3$  to  $1000 \text{ cm}^3$  and  $1150 \text{ cm}^3$ , for girls and boys, respectively. This means, that the brain is approximately 80 % of its full size at the age of two, [41, 40]. The growth of the brain still continuous throughout the next decade, but with a significant slower growth rate, [34].

## 2.2 Craniosynostosis

The brain has its most significant growth during the early years of life, as seen in Figure 2.2. A fusion of one or more of the cranial bones in this period will, therefore, have severe consequences for the development of the cranium, thus the growing brain. Premature fusion of cranial sutures is referred to as craniosynostosis (CS), and is estimated to affect 1 in 2100-2900 newborns, [22, 30].



**Figure 2.2:** Brain growth (percentage of adult brain weight) as a function of time. It should be noted that the figure depicts brain weight and not brain volume. From [21].

CS can be divided into syndromic and non-syndromic cases, of which non-syndromic is by far the most common form. Looking at non-syndromic cases, these can be further divided into a number of types, depending on the location of the closed suture. Depending on the involved suture, the head develops a characteristic shape, due to dysplastic and compensatory growth. The different types and their prevalence are briefly outlined in Table 2.1, together with a short description of the skull deformity involved. The most common form is sagittal synostosis, characterized by a long and narrow head shape, as can be seen in Figure 2.3(b). In some cases, 4–8 %, two or multiple sutures are affected, [22, 38], but these cases are not addressed further in this context.

Syndromic craniosynostosis is less common than non-syndromic craniosynostosis. It arises from known disorders causing malformations. Today more than 130 different syndromes are identified to cause these synostotic malformations, [30]. Some of the most common are the Crouzon and Apert syndromes, which both are estimated to have a prevalence of around 15 in 1,000,000 – Crouzon though slightly more prevalent, [22]. Syndromic synostosis may be present in only a single cranial suture, but typically multiple sutures are involved. Dysmorphology is therefore expressed with a high variability, which is seen pronounced also within each syndromic type. This variability is caused by a complex interaction between a number of factors *e.g.* genetic factors, cellular events and local forces, which all affect the normal growth and pattern of development, [17].

### 2.2.1 Clinical Manifestations and Craniotomy

As stated above, CS can be expressed in many different forms, and with a high degree of variability, but symptoms in one or the other form will always be

**Table 2.1:** Classification of the four most common forms of non-syndromic craniosynostosis, based on [22, 38].

Suture	Abbreviation	Prevalence	Deformity description
<b>Sagittal</b>	SS	50 – 60 %	Long and narrow skull, often referred to as boat-shaped
<b>Coronal</b> Unilateral	RUCS (right) LUCS (left)	~ 20 %	Flat lateral forehead over the suture involved, which in compensation often results in a more prominent forehead on the opposite side
Bilateral	BCS	~ 10 %	Broad and tall skull (anterior-posterior distance is restricted)
<b>Metopic</b>	MS	5 – 10 %	Narrow and flattened frontal bone, which gives the skull a triangular shape

present. This is due to the fact, that the development and growth of the brain will persist, regardless the restriction caused by the fused sutures. If brain growth is limited in a given direction it will compensate and expand in another possible direction, hence where fusion of the suture has not yet taken place. In Figure 2.3 three different examples of craniosynostosis are shown; unicoronal, sagittal and metopic, respectively. In all three cases, dysplastic compensatory growth is seen, but due to the different locations of the closed sutures, the compensatory effect is manifested differently in each case. If no sutures are open (worst case scenario), the cranium can still remodel by performing bone resorption on the inner surface of the skull (osteoclast activity), and bone apposition on the outer surface (osteoblast activity). This is a common mechanism, which also takes place in normal cranial growth in order to provide the right cranial curvatures. However, this mechanism will, in cases with severe craniosynostosis, serve as a compensatory remodeling process, which leads to cranial and facial dysmorphology, [34].

Besides the obvious malformation in craniofacial shape, CS can manifest itself with a high number of different consequences. Craniofacial deformity, which is the most pronounced one, can be seen in both severe and mild cases. In some situations, especially in the syndromic cases, it can lead to nasal breathing problems, which for some children can give rise to sleep apnea. Sleep apnea



**Figure 2.3:** Three characteristic manifestations, caused by different types of craniosynostosis. From [25].

can, if lasting over a longer period, affect the growth pattern, speech ability, and cognitive functions, [23]. All together, these effects will often cause a reduced ability or desire for social involvement, from which a consequence often is lack of personal development. It must be noted, that these complications only most often are seen in severe cases.

Another aspect in CS is the intracranial volume (ICV). A decrease in the ICV may cause a physical constriction on the growing brain, leading to increased intracranial pressure (ICP). A number of studies have therefore been conducted in order to get an impression of the ICV measure in children with craniosynostosis. When exploring the field concerning ICV estimation in children with UCS, no consistent conclusion is, however, made on the topic, but in [8], the conclusions of a number of studies have been collected. However, here some indicate a decreased ICV, whereas others conclude a small increase. In [40], a decreased volume is indicated until an age of 6 months, after which there is no significant difference between children with UCS and unaffected children. Furthermore, studies concerning the ICP give very tenuous results, so no evident conclusion is made concerning the relationship between the ICV and ICP.

To recapitulate, CS does not only manifest itself with craniofacial deformities, but also other and maybe more severe complications can be seen. Surgery is, therefore, in the Western world, functional performed in, more or less, all cases of CS, in order to provide the best possible conditions for these children. In some mild situations surgery can be performed by endoscopic repair. By means of two or more small incisions made near the fused suture, small portions of bone are removed and the suture is reopened. This type of surgery can, however, only be performed in mild cases of CS and only in children younger than 6 months, [13]. More comprehensive surgery, which involves craniotomy and remodeling of bone, is performed in more severe cases. Before surgery, an overall evaluation

of the cranial growth has to be established. Even though only one suture is involved, the involvement can be of high complexity, due to the compensatory growth in the other sutures [22, 30]. In general, each skull must be treated in its own way, in order to compensate for the malformations and functional problems present in the specific case.

In these cases of surgery, medical imaging plays an important role. Today magnetic resonance (MR) imaging and computed tomography (CT) are the standard imaging techniques for these cases, each of them having its own pros and cons, as later discussed in Section 3.2.1. For preoperative evaluation, CT scanning is the preferred modality, due to the clear outlining of sutures and synchondroses. MR imaging is often only used in more complex cases, where the brain and cerebrospinal fluid must be evaluated, [30].

In Figure 2.4, before and after surgery pictures are presented of a case of left unicoronal craniosynostosis. The two images to the left are preoperative, the middle and right hand images are postoperative, taken in ages of 6 and 15 months, respectively.



**Figure 2.4:** Left unicoronal craniosynostosis before and after surgery. The two images to the left are preoperative, the middle and right hand images are postoperative, taken in the age of 6 and 15 month, respectively. From [35].

In order to establish a better foundation for analyzing craniosynostotic cases, an estimation of the intracranial volume is, as mentioned above, of high interest. Such estimations have been attempted throughout the last century, but today difficulties are still seen among some of the applications. A historical review concerning application approaches are given in [Chapter 3](#).



## CHAPTER 3

# Volume Estimations up until Today

---

Information concerning size, shape and volume of the brain has been of high interest for many years. First of all, it provides basic anatomical knowledge, but it also provides the opportunity to compare crania from different populations, and examine how geography, race, gender, ethnicity *etc.* influence the cranial growth and development. Medically, knowledge about the cranial growth and development provides the opportunity to analyze crania with abnormal growth patterns and malformations. In Section 3.1 a short flashback to earlier methods and studies are given, followed by currently used methods in Section 3.2.

### 3.1 Historical Flashback

Studies trying to estimate the ICV have been performed for more than ten decades, where various approaches have been taken into use. In the following a brief introduction to different approaches is given. It should be noted that results and uncertainties are not elaborated on. The aim is only to give a description of the development in the field.

The first attempts were made on dry skulls, mostly with different filling tech-

niques. One of the first scientists, who came up with a method, was a German student in 1896 [29]. The idea was to lower a balloon into the cranial capacity and fill it with water. However, the method was not satisfactory. When using thin rubber balloons, these tended to rupture before a measurement could be made. When using thicker rubber material, air filled holes were seen between the balloon and the skull, which caused erroneous results. Other filling materials *e.g.* gunpowder, paraffin wax, millet and mustard seeds were subsequently tried. In the beginning, filling was performed manually, but in the end of the 1920s a cranial capacity machine was developed by M.S. Goldstein, which should eliminate the human errors, [29].

In 1901, a more mathematical method was also developed by Lee and Pearson, who designed a linear equation, based on three cranial measures; maximum anterior-posterior length, maximum width and cranial height. The measures were found with simple measuring equipment and were in the beginning still found on dry skulls, [29].

In the 1950s, the x-ray machine had made its entry, and made it possible to perform volume estimations on internal measures, instead of external. In the beginning, volume estimations were still based on length, width and height, but in 1972 the x-ray was combined with tomography. A brain scan could now consist of six slices, where the volume of each slice was found by taking the average of the two planes bounding it, and multiplying with the distance between them, [29]. Lee-Pearsons equation, which had been modified a number of times over the last couple of decades, was further modified to incorporate the thickness of the skull, calculated from x-ray. In 1977, a large study on ICV estimation was conducted on more than a thousand living Caucasians in different age groups, and has been used as reference materials for other studies in the years to come, [12].

With the development within the field of CT imaging, more slices and a better resolution has been obtained. This has led to semi-automatic slice wise volume estimation techniques, as described in Chapter 5. Also, a fast development was seen within the field of MR imaging. Similar to CT, slice and resolution improved significantly for MR imaging, and slice wise volume estimation can now be performed for this modality as well. Furthermore, automatic segmentation processes can be applied for MR images as outlined in the following section.

## 3.2 Current Methods for Automatic Segmentation

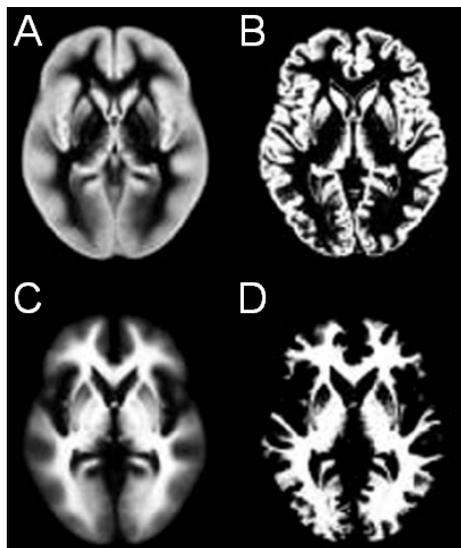
At the moment no fully automatic CT-based application for segmentation of the intracranial volume is accessible. However, applications used within others areas, *e.g.* for segmentation of the lung tree, are available, and inspiration has been found in these. A various segmentation methods have been developed for MR images, and the following will give a short introduction to some of the existing methods. These automatic MR applicable methods are considered in order to see whether a direct transfer to a CT application can be made.

Today's MR techniques have evolved significantly, and the assessment of the brain volume is a well-known and often used feature for multiple purposes. Many software packages for automatic estimation of the brain volume exist. Note the notation brain volume instead of intracranial volume. This is due to, that most MR applications has an interest in assessing brain volume instead of ICV. In the following two MR imaging software packages are briefly described below; Statistical Parametric Mapping (SPM) and Functional MR Imaging of the Brain Software Library (FSL).

**SPM** is a software package designed for analysis and hypotheses testing of functional imaging. It handles data from a variety of brain imaging sequences, inter alia functional MR imaging (fMRI) and positron emission tomography. In the preprocessing of fMRI data, it is possible to segment the different matters of brain, for *e.g.* subsequent facilitation of co-registration in between subjects. The estimation of the volume is primarily based on segmentation of the brain into cerebrospinal fluid, gray matter and white matter, using voxel intensities and a prior tissue probability map, see Figure 3.1. The resulting segmentation map is an estimation of the probability distribution for each of the tissue clusters, found from a maximization of the posteriori solution, [6, 36]. In the study performed by Pengas *et al.* [36] the intracranial volume is estimated by summing the number of voxels, with high enough probability of belonging to one of the tree classes, and multiplying with the volume of one voxel. The volume segmentation is thereby not performed on the basis of information of the skull.

**FSL** is an analysis library for fMRI and MR brain imaging data. FSL has a 3D brain/non-brain incorporated, which "*uses a deformable model that evolves to fit the brain's surface by the application of a set of locally adaptive model forces*", [43]. The algorithm, called Brain Extraction Tool (BET), works with a spherical surface, constructed with a triangular tessellation, and expands until the local intensity structure changes and the brain edge is

reached. Depending on the model forces the algorithm could find the brain surface or the surface of the skull, [43]. The initial spherical surface, and final brain/non-brain surface is seen in Figure 3.2.



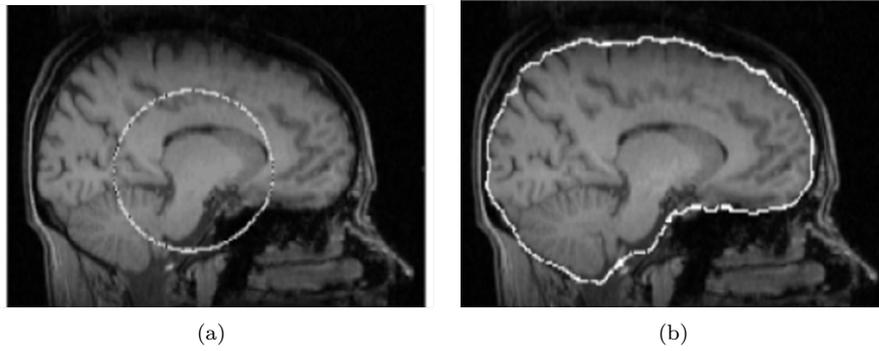
**Figure 3.1:** Prior probability maps for gray (A) and white matter (C), and the resulting segmentation (B and D). Modified from [6].

The two brain matter based segmentation is not directly applicable to CT images due to their low contrast between soft tissues, but an algorithm like the BET could, with modification, be considered for CT. Other algorithms suitable for both modalities could be *e.g.* graph cut or an image registration approach as implemented for CT in this thesis. In Section 3.2.1 the pros and cons for volume estimation in CT versus MR are briefly outlined.

### 3.2.1 Considerations of MR Versus CT in Volume Estimation

When considering both MR and CT imaging for estimation of the intracranial volume in children with craniosynostosis, a variety of considerations must be taken. In Table 3.1 and Table 3.2 a brief overview of the pros and cons for estimating the intracranial volume with MR and CT is provided.

Evaluating the pros and cons, the key points are, that CT scans are the cornerstone for diagnosis and surgery planning, [30]. Especially the possibility for easy



**Figure 3.2:** Illustration of the initial spherical surface (a) and the resulting brain/non-brain surface (b) for the BET algorithm. Modified from [43].

**Table 3.1:** Pros and cons for ICV estimation using MR.

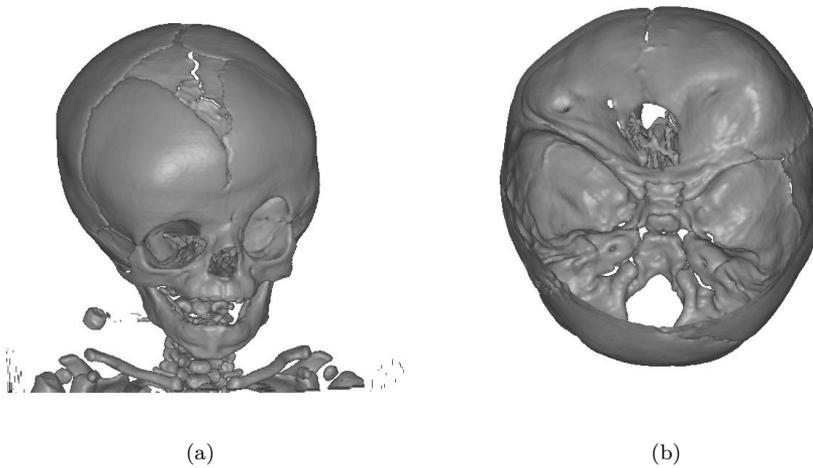
Pros - MR	Cons - MR
Well-known methods and packages exist	Primarily based on brain matter and not the skull
No biological hazards	Distortions and bias field variations give a higher risk of error - mainly peripheral, <i>i.e.</i> near the skull
Easier to overcome foramina, if using brain matter instead of bone for segmentations	A bit more expensive (compared to CT)
	Noisy and time consuming, which will require newborns to be under anesthesia
	No simple preoperative evaluation of the bone and sutures

thresholding and surface rendering makes CT preferable for 3D bone visualization. Furthermore, CT imaging provides a faster and cheaper imaging tool than MR imaging, but it must be emphasized that a CT scan holds risk in the form of radiation. Another key aspect is the MR scan time and noise level, which

Table 3.2: Pros and cons for ICV estimation using CT

Pros - CT	Cons - CT
Clear illustration of the skull - also in 3D	Hard to incorporate new or existing MR segmentation techniques
No peripheral distortion	Harmful due to radiation
Currently used for surgery planning	Harder to overcome the lack of information in some part of the skull of a newborn. See Figure <a href="#">3.3</a>
Cheaper and faster	

requires that small children must be under anesthetic, in order to obtain avoid movement artifacts. By using CT, this can in some cases be avoided. However, other difficulties will arise, when automatic intracranial volume segmentation should be based on CT images. One of the most profound challenges is to overcome the foramina and lack of skull information, as seen in Figure [3.3](#). This is one of the aspects investigated in this thesis, by applying an image registration and graph-based segmentation on CT data.



**Figure 3.3:** 3D surface rendering from dataset #6, showing a thresholded CT image of a child with unilateral synostosis. (a) depicts the full skull, and a severe gap between the cranial bones is seen. In (b) the skull base is viewed from the top. The foramen magnum is seen at the bottom of the image.



# Data Information and Definitions

---

3D CT head scans were provided by the 3D Laboratory. Data were recorded for the prospect of surgery planning, and collected by the 3D-Laboratory with the prospect of further research in the field of craniosynostosis. Data processed in this thesis, were primarily provided by Copenhagen University Hospital Rigshospitalet, with the exception of dataset #7, which was acquired from Helsinki University Central Hospital<sup>1</sup>. All data were obtained from CT scans of children in the age range from a half to one and a half years of age, all diagnosed with either RUCS and LUCS. In Table 4.1 the place of origin for the datasets in use is provided, together with the diagnosis, sex and age at the scanning.

The voxel dimensions are seen to differ between datasets, but the number of voxels are seen to have been held constant at 512 in the  $x$ - and  $y$ -direction. The number of voxels in the  $z$ -direction, however, is seen to vary significantly among datasets. An overview of the data resolution is provided in Table 4.2<sup>2</sup>.

---

<sup>1</sup>Data has kindly been provided by: Jyri Hukki, Professor, MD, Cleft Palate and Craniofacial Center, Helsinki University Central Hospital, Helsinki, Finland.

<sup>2</sup>It is noted that dataset #1 had a shift in its intensity range. This has been corrected before application of segmentation algorithms

**Table 4.1:** Data information, including place of origin and patient information.

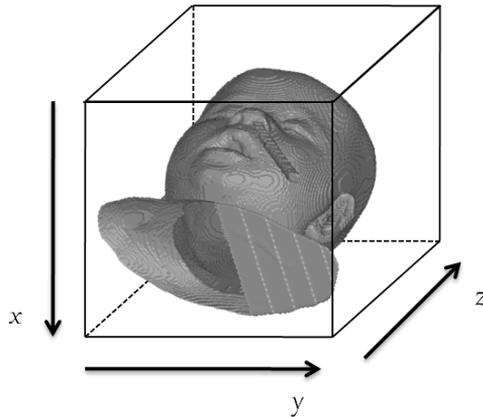
#	Origin	Diag.	Sex	Age at Scan (days [months])
1	Copenhagen	LUCS	M	129 [4.5]
2	Copenhagen	RUCS	M	127 [4.5]
3	Copenhagen	RUCS	F	306 [10.2]
4	Copenhagen	LUCS	M	257 [8.5]
5	Copenhagen	LUCS	F	413 [13.8]
6	Copenhagen	RUCS	M	136 [4.5]
7	Helsinki	RUCS	F	437 [14.5]
8	Copenhagen	RUCS	M	162 [5.4]
9	Copenhagen	RUCS	M	130 [4.3]
10	Copenhagen	LUCS	M	289 [9.6]
11	Copenhagen	LUCS	F	159 [5.3]
12	Copenhagen	LUCS	F	157 [5.3]
13	Copenhagen	LUCS	F	483 [16.1]
14	Copenhagen	RUCS	F	444 [14.8]
15	Copenhagen	LUCS	F	206 [6.8]

**Table 4.2:** Data and image information.

#	Resolution [mm]			Voxels in $z$
	$x$	$y$	$z$	
1	0.37	0.37	1	178
2	0.36	0.36	1	167
3	0.43	0.43	0.65	289
4	0.43	0.43	0.6	325
5	0.43	0.43	1	184
6	0.39	0.39	0.8	241
7	0.45	0.45	0.6	313
8	0.39	0.39	0.9	189
9	0.34	0.34	0.8	247
10	0.39	0.39	0.8	251
11	0.39	0.39	1	178
12	0.49	0.49	0.8	226
13	0.47	0.47	0.8	226
14	0.39	0.39	0.6	350
15	0.37	0.37	1	311

## 4.1 Dimension Terms

A brief introduction to the different dimension terms is provided in the following. When data were acquired, all subjects were equally placed in the scanner, which gives the possibility to globally define the three directions -or dimensions. In Figure 4.1 a simple illustration of a patient in a "scanner" is seen. The cube indicates the CT image that is obtained, with the defined directions,  $x$ ,  $y$  and  $z$ . Here  $x$  defines the image rows,  $y$  the image columns, and  $z$  the image slices. In the image registration process, these parameters are referred to as  $\mathbf{x}^1$ ,  $\mathbf{x}^2$  and  $\mathbf{x}^3$ , respectively, and the entire cube, *i.e.* the image, is referred to as  $\mathbf{X}$ .

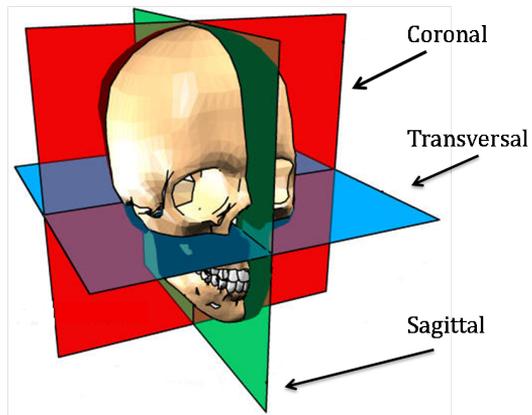


**Figure 4.1:** Simple illustration, defining the three scan directions.

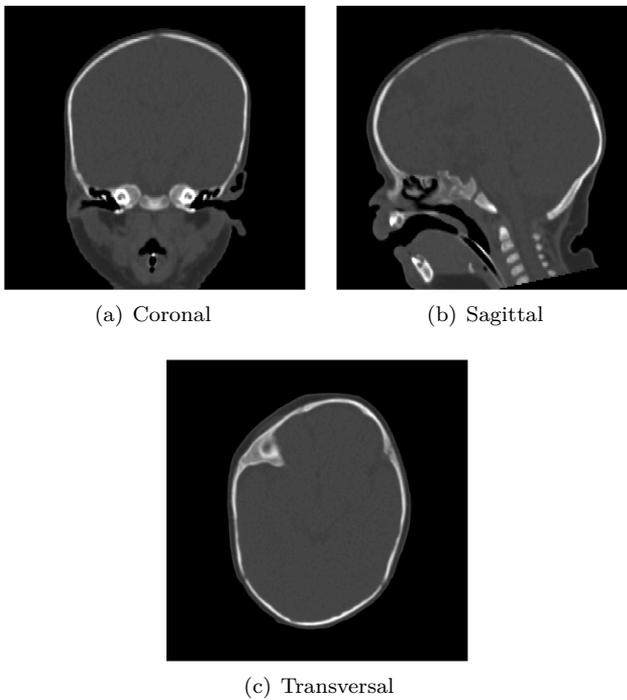
In addition to the terms outlined above, 3D images are often viewed or evaluated in 2D, *i.e.* a slice viewed in a given plane. In medical imaging these planes are globally defined, as coronal, sagittal and transversal, as in Figure 4.2. An example of a 2D slice in each plane is seen in Figure 4.3.

## 4.2 Multilevel

All images are analyzed with the two deformable models; image registration and graph cut. Since both models demand a relatively large setup, it can be difficult to process the images with their original resolution, due to limitations in time and capacity. All images have therefore been downsampled. In fact



**Figure 4.2:** Definition of image planes. Modified from [14].



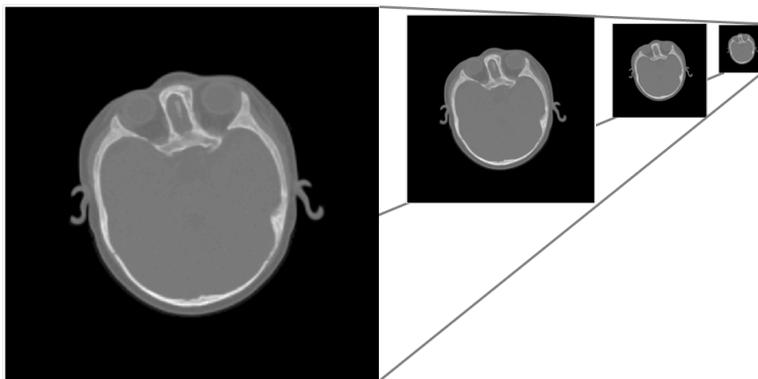
**Figure 4.3:** Examples of 2D images in the three image planes; coronal, transversal and sagittal.

an image pyramid approach is taken, mainly for the use of scale space in the image registration. An image pyramid is a multilevel representation of the same image, where each level represents the image with a lower resolution. The higher the level, the lower the resolution. An image pyramid consisting of four levels is constructed, as seen in Figure 4.4.  $L_0$  defines the original image and thereby the finest level, and  $L_3$  the coarsest level. The lower levels are achieved by applying a low-pass filter to the previous level, in the form of an Gaussian kernel with a size of  $3 \times 3 \times 3$  voxels and a standard deviation of 0.8, and thereafter extracting every other voxel. The filtering is performed in order to minimize the loss of information by smoothing the image.

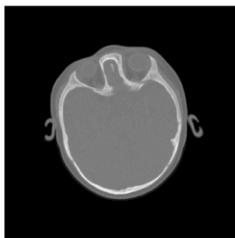
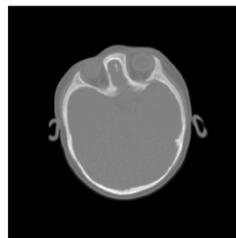
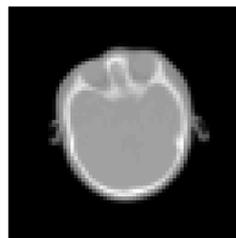
The four levels are defined in Table 4.3, together with the resulting image size and mean voxel dimensions, found from the 15 datasets in Table 4.2.  $m^3$  defines the slice number which varies between datasets (the notation originates from Chapter 6). It should be noted that the voxel size of the Gaussian kernel is maintained through the levels. However, the size is modified to each level, in consequence of the change in voxel dimensions.

**Table 4.3:** Definition of the four levels defining the image pyramid.  $L_0$  gives the image with its original size and resolution, followed by the coarser levels. For each downsampled level, the size is reduced, which entails in larger voxel dimensions.  $m^3$  defines the slice number, which varies between the datasets.

Level	Size [Voxel]	Mean Voxel Dimensions [mm]
$L_0$	$512 \times 512 \times m^3$	$0.40 \times 0.40 \times 0.86$
$L_1$	$256 \times 256 \times m^3/2$	$0.79 \times 0.79 \times 1.73$
$L_2$	$128 \times 128 \times m^3/2^2$	$1.58 \times 1.58 \times 3.45$
$L_3$	$64 \times 64 \times m^3/2^3$	$3.17 \times 3.17 \times 6.90$



(a) Image pyramid

(b)  $L_0$ (c)  $L_1$ (d)  $L_2$ (e)  $L_3$ 

**Figure 4.4:** Illustration of the four level image pyramid. In (a) the idea of the image pyramid is outlined, with the original image,  $L_0$ , with full resolution in front, followed by the lower downsampled levels. In (b)-(e) a slice image from each level is shown. The voxel size is increased for each level, giving a resolution that at level  $L_3$  results in a clearly blurred image presentation.

# Manual Segmentation

---

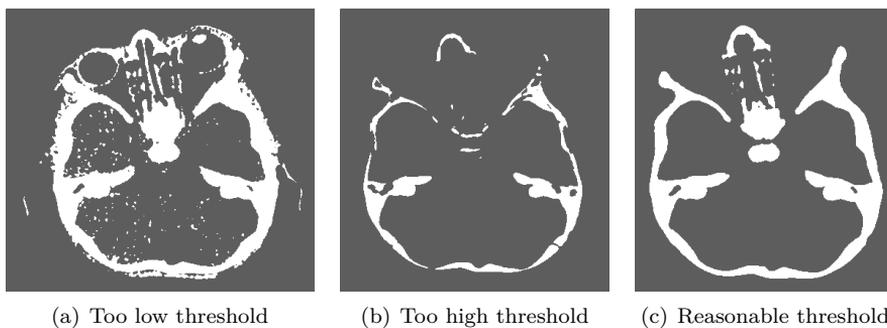
In order to validate the two segmentation methods all images have been manually segmented and their calculated volume used as a reference for the two model based segmentations. The manual segmentation is actually a semi-automatic method, based on a user chosen intensity threshold in Hounsfield Unit (HU), and a seed unifying algorithm. This segmentation form will throughout this thesis be referred to as manual segmentation. Two programs were involved in the process; the intensity threshold was chosen in *Landmarker*, based on the best possible bone segmentation and the actual intracranial segmentation was performed in *Analyze*. The two programs and their use are described in Section 5.1 and 5.2, respectively.

## 5.1 Landmarker

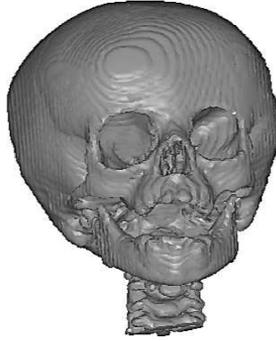
*Landmarker* is a visualization program for volumetric data, with a graphical user interface for landmark placing as its main functionality. Furthermore, it holds very good visualization and inspection possibilities, both regarding slices and 3D surfaces. Landmarker is public accessible through the homepage of the 3D-Laboratory, [11], and is used for 3D surface rendering throughout this thesis.

Before the segmentation process, an image is loaded into the program and an

intensity threshold is chosen. The threshold is decided by the user, based on the best achievable bone segmentation, hence allowing as much bone as possible without inducing noise artifacts in the image, as *e.g.* information from ears or brain structures. Especially ears and pupils have been used as indicators for choosing an appropriate threshold. In Figure 5.1, three images with different thresholds are presented. In (a) and (b) examples of the consequences of choosing a too low or a too high threshold, respectively, is seen. In (a) noise components from *e.g.* the eyes can clearly be seen, whereas in (b) the bone structure is seen incomplete. In (c) an image with a threshold providing a reasonable result is depicted. No significant noise artifacts are present and the bone structure is kept intact. When a satisfactory intensity threshold is chosen, a polygonal surface rendering is constructed as seen in Figure 5.2, where it is possible to inspect and evaluate the result. Also here, the orbits are used to validate the result. The orbits can, due to their very thin bone line, give a good indication of a reasonable intensity threshold, *e.g.* a too high threshold will cause small gaps in the orbits. When a satisfactory result is achieved the threshold is noted, and *Landmarker* is closed and the image is loaded into *Analyze*. The chosen intensity thresholds for the processed datasets are seen in Table 5.1.



**Figure 5.1:** Same axial slice of a dataset presented with three different thresholds. In (a) a too low threshold is seen, which induces a lot of noise components. The image in (b) has a too high threshold, meaning that some of the wanted information of the cranial bones is lost. In (c) a threshold in between the two previous was selected, which entails that the noise components are removed, while the cranial bones are still remained, more or less, unaffected.



**Figure 5.2:** Surface reconstruction of the image, when using the threshold found in 5.1(c). The surface can in Figure *Landmarker* be rotated and seen from all angles and thereby give the user a good possibility to evaluate the result. Here an angle is chosen, where the orbits clearly can be seen, showing no holes in that area.

**Table 5.1:** User chosen intensity thresholds for all datasets.

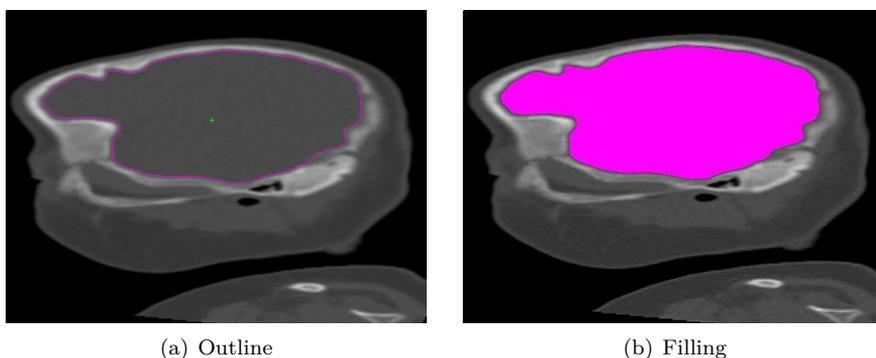
#	1	2	3	4	5	6	7	8	9
Threshold [HU]	129	139	100	120	216	105	105	114	105

#	10	11	12	13	14	15
Threshold [HU]	135	109	100	109	124	230

## 5.2 Analyze

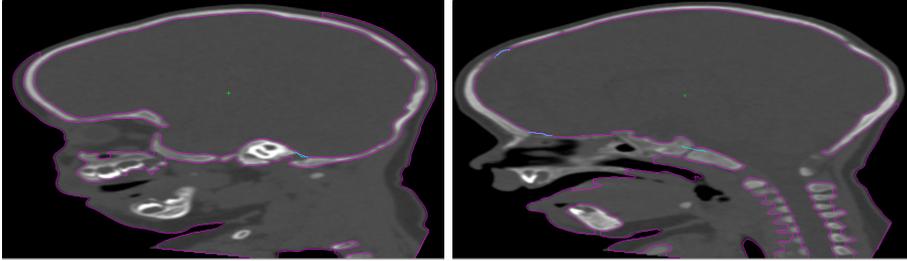
*Analyze* is a program developed for medical volumetric data, and is used for analysis, processing and visualization. Furthermore, *Analyze* enables a segmentation process based upon thresholding and/or morphology. The following will be a walk-through of the steps used in the segmentation process. It should be noted, that all illustrations do not possess the right anatomical scaling, due to these are obtained as screen dumps from the segmentation process.

When an image is loaded, a volume render function is chosen. Subsequently the threshold found in *Landmarker* is specified to the program. A slice editor is chosen which gives the user the possibility of running through all slices in all three planes; transversal, coronal and sagittal. Furthermore, it is possible for the user to adjust the contrast window and level such that the best possible image contrast is achieved. A sagittal slice in the beginning of the left or right side of the cranium is chosen, preferably where the intracranial volume is properly bounded. In this slice a seed is manually placed inside the volume of interest, and is given the properties of unifying all voxels having intensities lower than the specified threshold found previously in *Landmarker*. A curve then marks the prescribed area, as seen in Figure 5.3(a). If the area is satisfactory, *i.e.* proper bounded, the area is filled as seen in Figure 5.3(b) and the next slice is chosen. As long as the intracranial volume is properly bounded, the program segments the area of interest in each slice, but as soon as a small breach appears, the segmentation is out of bound.



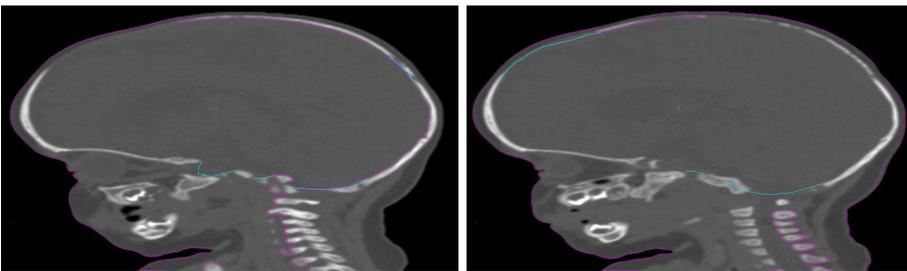
**Figure 5.3:** A sagittal slice showing the seed, given by the green cross, and found outlining illustrated with the pink line (a). In (b) the region inside the line is filled with pink color.

Gaps between bones will however be present, due to uneven growth caused by craniosynostosis or in some cases due to a normal late closure of the skull bones. Breaches will also appear in the cranial floor, due to the natural foramina to *e.g.* the brain stem. When a slice holds one or more breaches, the thresholded segmentation fails. In Figure 5.4, two examples of such failures are illustrated. In these situations it is up to the user to manually edit the segmentation and outline the cranial boundary in the necessary regions. Editing is performed best possible, but errors can and will be introduced, *e.g.* due to different interpretations of the anatomical map, especially in the cranial floor. Note that the process in *Landmarker* and *Analyze* in total takes approximately two hours, which has resulted in the sparse number of data processed in this thesis.

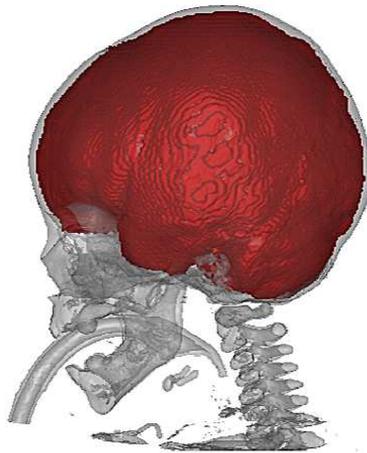


**Figure 5.4:** Two situations where the thresholded segmentation must be manually corrected. Manual corrections are visualized in green.

Breaches in the cranial ceiling can be of either mild or severe form. The mild cases are relatively easy to get past, due to the natural curvature of the cranial ceiling. Unfortunately, in some severe cases the breaches are of such size, that the user is left with a relatively demanding task of drawing a smooth and anatomically correct curve, see Figure 5.5. Furthermore, it must be noted that all drawings are performed by use of a mouse as the drawing tool. All together, these uncertainties will induce a certain error to the manually segmented volume, caused by the person performing the segmentation. The extent of this is further examined in Section 11.1. When the entire process is conducted a segmentation as presented in Figure 5.6 is obtained.



**Figure 5.5:** Examples of situations where manual editing for larger regions is necessary.



**Figure 5.6:** Example of a segmented mask seen as the red surface superimposed on its corresponding bone surface.

# Image Registration

---

Image registration is a process, which can transform different data into a common coordinate system. This chapter provides a description of the processes needed to obtain this alignment. However, before going into the actual process a short description on image representation is given in Section 6.1. In Section 6.2 an introduction to image transformation is outlined, followed by the two transformation forms used in this thesis; the affine and the B-spline transformation. The effect of the transformations are evaluated based on a dissimilarity measure described in Section 6.3. A short description of the regularizer linked to the B-spline transformation is given in Section 6.4. Lastly the multilevel approach used for this model is described in Section 6.5.

Image registration has become an important tool within medical diagnostic. It enables comparison of features of interest, *e.g.* to follow organ growth or shrinkage in patients. Furthermore, it enables comparison between different image modalities or comparison in between patients. It also gives the opportunity to compare a patient to a known image, or even to an atlas. An atlas is typically represented by a high resolution image and is obtained as a mean of many aligned and adjusted images. In order to perform these prior mentioned comparisons the images must be aligned best possible to each other. From this it follows that transformation of one of the images is a necessity, and finding the optimal transformation is exactly the goal behind image registration.

Image registration involves two images, a known template image *e.g.* an atlas and a reference image, which is the image of interest. The registration problem can then be phrased as by Jan Modersitzki [33], "*Find a reasonable transformation such that a transformed version of a template image is similar to a reference image*". The success of the transformation is often evaluated by a cost function, also referred to as an object function or dissimilarity measure, which can be based on different properties, dependent on the modality and appearance of the images.

## 6.1 Image Representation

Before going closer into image registration, a short description of image acquisition is necessary. Even in cases where images have been acquired with the same modality, as the ones used in this thesis, differences are seen, due to the acquisition settings of the used scanners. An important feature, which makes image comparison difficult, and therefore must be taken into consideration, is the voxel dimensions. Looking at images, these can be seen as sampled in a  $D$  dimensional grid, where a grid point  $\mathbf{x}_i$  is defined as in Equation 6.1 and each image dimension can be grouped into a vector as in Equation 6.2. A superscript notation is in the following used for accessing higher dimension arrays,  $d = 1 \dots D$ .  $\mathbf{x}^1$  represents the coordinates for the image rows,  $\mathbf{x}^2$  the image columns and  $\mathbf{x}^3$  the image slices, as described in Section 4.1.  $N$  defines the number of voxels in the image.

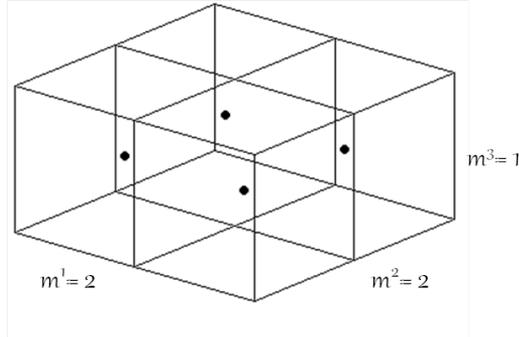
$$\mathbf{x}_i = [x_i^1 \ x_i^2 \ x_i^3] \text{ for } i = 1 \dots N; \quad (6.1)$$

$$\begin{aligned} \mathbf{x}^1 &= (x_1^1 \dots x_i^1)^\top \\ \mathbf{x}^2 &= (x_1^2 \dots x_i^2)^\top \\ \mathbf{x}^3 &= (x_1^3 \dots x_i^3)^\top \end{aligned} \text{ for } i = 1 \dots N; \quad (6.2)$$

Together, the three vectors define the 3D grid  $\mathbf{X}$ , of voxel coordinates.

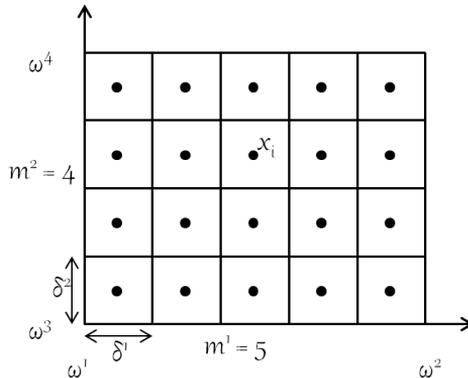
$$\mathbf{X} = [\mathbf{x}^{1\top} \ \mathbf{x}^{2\top} \ \mathbf{x}^{3\top}]^\top \quad (6.3)$$

Each point defines a cell in the regular 3D grid of size  $m = [m^1 \ m^2 \ m^3]$ ; for which a simple illustration is seen in Figure 6.1. Note that one row in  $\mathbf{X}$  corresponds to one voxel  $\mathbf{x}_i = [x_i^1 \ x_i^2 \ x_i^3]$ .



**Figure 6.1:** 3D grid illustration. The dots specify the center of each cell. Modified from [5].

However, as mentioned previously, images are often recorded with different voxel dimensions, which means that the voxel size in each dimension must be taken into consideration, in order to obtain the actual physical coordinates, hence the actual voxel grid. A simple 2D illustration can be seen in Figure 6.2, where the grid consists of the interval  $\Omega = (\omega^1, \omega^2) \times (\omega^3, \omega^4)$ . The superscriptions are here defined as  $(\omega^{2d-1}, \omega^{2d})$  for  $d = 1 \dots D$ . In order to access the center of the cells, the grid points from Equation 6.2 must be modified according to the voxel size in each dimension, found from  $\delta^d = (\omega^{2d} - \omega^{2d-1})/m^d$ , [33].



**Figure 6.2:** Simple 2D grid illustration. Modified from [33].

## 6.2 Transformation

With the image representation in place, the focus can now be returned to image registration and transformation forms. A registration example is illustrated in Figure 6.3, where the square in the top left figure is the template image and the circle in the top right, the reference. A transformation of the square to the circle can be found from Equation 6.4.

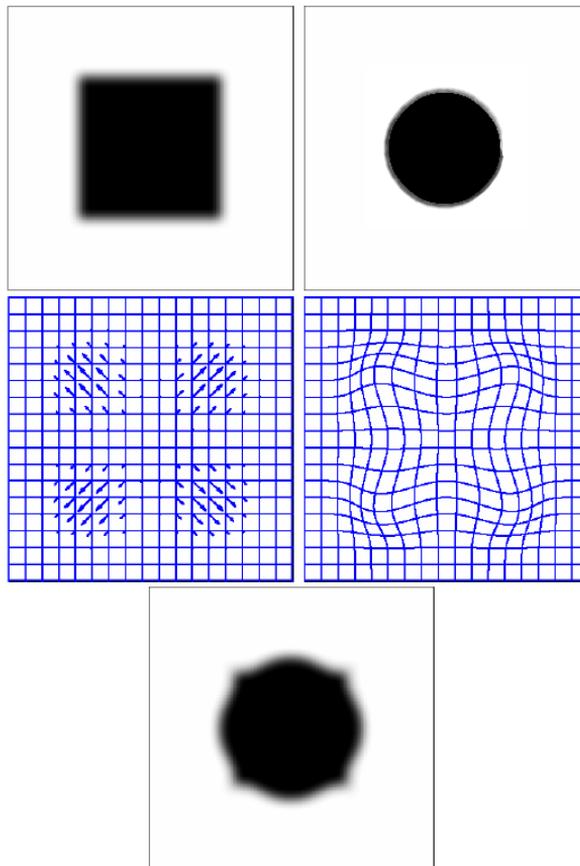
$$\mathbf{Y}(\mathbf{X}; \cdot) = \mathbf{X} + u(\mathbf{X}; \cdot) \quad (6.4)$$

$\mathbf{X}$  forms the identity part represented by the grid points defined in Equation 6.3, and  $u(\mathbf{X}; \cdot)$  the deformation part, represented by the deformation vectors, all illustrated in Figure 6.3 left middle. The  $\cdot$  represent the actual deformation, which is a number of parameters found from a chosen transformation form. When the deformation vectors are applied to the underlying grid, the grid will deform as seen in the right middle, which will result in a transformed version of the grid,  $\mathbf{Y}(\mathbf{X}; \cdot)$ , as seen in the bottom image. Difficulties are though seen for the corners of the square, which still is slightly present in the transformed image. However, the transformation of a square to a circle is a relative demanding task, even though they have simple shapes, and errors are therefore expected. The transformation is always applied to the template image, due to the fact that the reference image is the one of interest and all information in that image is wished kept unchanged.

To perform a transformation, as the one seen in Figure 6.3, a non-rigid transformation is necessary, in order to transform the linear lines of the square to the curvature of the circle. This will be the same case in this thesis, where the transformation must be able to adjust for non-rigid dissimilarities, between two images. First an affine transformation is applied. This transformation form can correct for differences in position, orientation and general size of the head, and thereby provide a rough alignment of the two images. Afterwards another non-linear transformation, based on B-splines is applied, which can correct for curvatures and finer differences. The two transformation forms are described in Section 6.2.1 and 6.2.2.

### 6.2.1 Affine Transformation

The affine transformation is a geometric transformation, which transforms an image by use of translation, rotation, anisotropic scaling and shear. The affine transformation is non-linear due to the opportunity of translation, and non-rigid due to the induction of shear. It preserves the straightness and parallelism of lines, equal to its similar rigid transformation forms, but facilitates a rotation



**Figure 6.3:** The two top images show the template and reference image, respectively. The template represented by the square and the reference as the circle. The aim is now to transform the square to the circle. In the two images below, the initial grid point with the deformation vectors is seen to the left and to the right the resulting warped grid is shown. The bottom image shows the transformed template image, which is obtained when applying the warp grid to the original template image. A high similarity is now seen between the initial reference image and the transformed template. Modified from [27].

between the otherwise perpendicular lines. This can be beneficial, especially in CT images, where the gantry angle can be incorrectly recorded, which can induce skew into the images, [15].

According to [27], the affine transformation can in its short form be written as in Equation 6.5, where the rotation, shear and scaling parameters can be derived from  $\mathbf{A}$  and the translation directly from  $\mathbf{t}$ .  $\mathbf{y}_i$  defines the transformed grid point. Further it can be written in its matrix form as seen in Equation 6.6, where the matrix  $\mathbf{A}$  and the vector  $\mathbf{t}$  is compiled in one matrix,

$$\mathbf{y}_i(\mathbf{x}_i; \mathbf{A}, \mathbf{t}) = \mathbf{A}\mathbf{x}_i + \mathbf{t} \quad (6.5)$$

$$\begin{bmatrix} y_i^1 \\ y_i^2 \\ y_i^3 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & t_1 \\ a_{21} & a_{22} & a_{23} & t_2 \\ a_{31} & a_{32} & a_{33} & t_3 \end{bmatrix} \begin{bmatrix} x_i^1 \\ x_i^2 \\ x_i^3 \\ 1 \end{bmatrix} \quad (6.6)$$

In order to solve the system in a universal equation system, the matrix system is rewritten and expressed in Equation 6.7, [44],

$$\begin{aligned} y_i^1 &= x_i^1 + \sum_{j=1}^4 w_j^1 q_j(\mathbf{x}_i) \\ y_i^2 &= x_i^2 + \sum_{j=1}^4 w_j^2 q_j(\mathbf{x}_i) \\ y_i^3 &= x_i^3 + \sum_{j=1}^4 w_j^3 q_j(\mathbf{x}_i) \end{aligned} \quad (6.7)$$

where  $q$  denotes the very simple basis function  $q(\mathbf{x}_i) = [x_i^1 \ x_i^2 \ x_i^3 \ 1]^T$  and  $w^d$  denotes the weight parameter  $[a_{d1} - 1 \ a_{d2} \ a_{d3} \ t_d]$ . This can be gathered in a single equation system,

$$\mathbf{Y} = \mathbf{X} + \begin{bmatrix} Q & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & Q \end{bmatrix} \mathbf{w} = \mathbf{X} + \mathbf{I}_3 \otimes Q \mathbf{w} = \mathbf{X} + \mathbf{Q}\mathbf{w} \quad (6.8)$$

where  $\mathbf{Y}$  gives the transformed grid,  $\mathbf{X}$  denotes the initial grid, and  $Q$  a matrix formed by compiling all row vectors  $q(\mathbf{x}_i) = [x_i^1 \ x_i^2 \ x_i^3 \ 1]^T$  and  $\mathbf{w}$  is a vector holding the affine parameters. Together  $\mathbf{Q}$  and  $\mathbf{w}$  corresponds to the deformation part, which in Equation 6.4 was defined as  $u(\mathbf{X}; \cdot)$ , where  $\mathbf{w}$  constitute the transformation parameters and takes the place of  $\cdot$ . Note that  $\otimes$  denotes the Kronecker product.

### 6.2.2 B-Spline Transformation

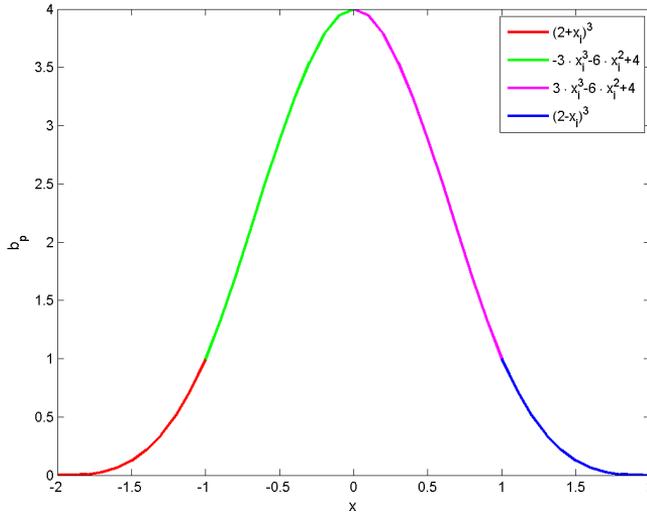
By the use of the affine transformation the template image has been aligned and scaled to match the reference best possible. However, since the affine transformation holds the straightness of lines, deformations regarding curvatures have not been taken into consideration. A way to introduce such non-linearity into the transformation is by use of splines. Splines are piecewise polynomial basis functions of degree  $Df$  defined over a knot sequence  $\xi$  with  $P$  knots, and where each basis function covers a subinterval of knot sequence. The most common used spline is the cubic spline, which is given when  $Df = 3$ . When working with optimization of image registration the cubic basic-spline (B-spline) is preferred, since one of its main features is local support. This means that it is possible to make changes that only affect a local neighborhood. Moreover, for higher dimensional purpose the B-spline model has the advantage that it can be constructed from a number of 1D cubic splines, [27].

The 1D cubic B-spline with  $P = 4$  can be described by the *mother* function,  $b(x_i)$  in Equation 6.9, [44], which gives the curve illustrated in Figure 6.4. The B-spline is twice differentiable and has the property of only being active in a four neighboring knot interval, which gives the B-spline its important feature of providing local support, [28].

$$b(x_i) = \begin{cases} (2 + x_i)^3 & \text{for } -2 \leq x_i < -1; \\ -(3x_i + 6)x_i^2 + 4 & \text{for } -1 \leq x_i < 0; \\ (3x_i - 6)x_i^2 + 4 & \text{for } 0 \leq x_i < 1; \\ (2 - x_i)^3 & \text{for } 1 \leq x_i < 2; \\ 0 & \text{otherwise} \end{cases} \quad (6.9)$$

At each knot point a B-spline is centered, and a knot interval is thereby covered by shifting the mother function in Equation 6.9. The B-splines will overlap such that when the influence of one function declines, another function will increase its influence, as seen in Figure 6.5. A point can however never be influenced by more than four B-splines. Furthermore, each basis function is associated with a weight  $w_p$ , which determines the influence of the given function. The B-splines can be implemented with either free or fixed boundaries. Free boundaries can be obtained by placing three additional knots equidistantly outside the image boundary, while three knots on top of each boundary knot, will force zeros outside the image, hence giving a fixed boundary.

The B-splines illustrated in Figure 6.5 are shown for only one dimension. Since the data used in this thesis is in 3D, similar basis functions will be seen in all



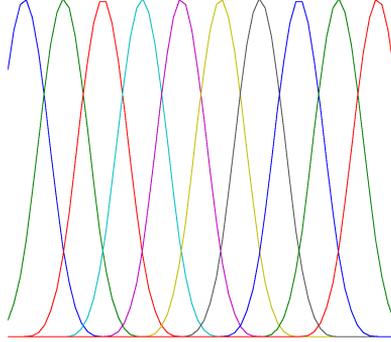
**Figure 6.4:** The cubic spline basis functions given for  $-2 \leq x < 2$ .

three dimensions. The extended 3D transformation can therefore be written as a product of B-spline functions, as seen in Equation 6.10, where  $p_1$ ,  $p_2$ , and  $p_3$  defines the knots in the three dimensions,  $p = 1 \dots P_1 P_2 P_3$ , and  $\xi^d$  defines the knot sequence in the dimension  $d$ . In this way new basis and weight functions are created similar to Equation 6.7, which entails that the transformed coordinates can be found based on Equation 6.11, [44].

$$q_p(\mathbf{x}_i, \xi^1, \xi^2, \xi^3) = b_{p_1}(x^1, \xi^1) b_{p_2}(x^2, \xi^2) b_{p_3}(x^3, \xi^3) \quad (6.10)$$

$$\begin{aligned} y_i^1 &= x_i^1 + \sum_{p_1=1}^4 \sum_{p_2=1}^4 \sum_{p_3=1}^4 b_{p_1}(x_i^1) \cdot b_{p_2}(x_i^2) \cdot b_{p_3}(x_i^3) \cdot w_{p_1 p_2 p_3}^1 \\ y_i^2 &= x_i^2 + \sum_{p_1=1}^4 \sum_{p_2=1}^4 \sum_{p_3=1}^4 b_{p_1}(x_i^1) \cdot b_{p_2}(x_i^2) \cdot b_{p_3}(x_i^3) \cdot w_{p_1 p_2 p_3}^2 \\ y_i^3 &= x_i^3 + \sum_{p_1=1}^4 \sum_{p_2=1}^4 \sum_{p_3=1}^4 b_{p_1}(x_i^1) \cdot b_{p_2}(x_i^2) \cdot b_{p_3}(x_i^3) \cdot w_{p_1 p_2 p_3}^3 \end{aligned} \quad (6.11)$$

The 3D basis functions for a given point can now be defined by a long vector,



**Figure 6.5:** Distribution of shifted B-splines in one dimension.

$q(\mathbf{x}_i)$ , holding all basis functions pronounced in that given point. Since all tensor B-spline basis functions are given as functions of individual coordinates, the  $q$  vector can be parted into three vectors, each holding one dimension, as in Equation 6.12.

$$q(\mathbf{x}_i) = \begin{bmatrix} q_1(\mathbf{x}_i) \\ q_2(\mathbf{x}_i) \\ \vdots \\ q_{P_1 P_2 P_3}(\mathbf{x}_i) \end{bmatrix}$$

$$q(\mathbf{x}_i) = \begin{bmatrix} b_1(x_i^3; \xi^3) \\ \vdots \\ b_{P_3}(x_i^3; \xi^3) \end{bmatrix} \otimes \begin{bmatrix} b_1(x_i^2; \xi^2) \\ \vdots \\ b_{P_2}(x_i^2; \xi^2) \end{bmatrix} \otimes \begin{bmatrix} b_1(x_i^1; \xi^1) \\ \vdots \\ b_{P_1}(x_i^1; \xi^1) \end{bmatrix} \quad (6.12)$$

The entire basis functions,  $Q$ , can now for all coordinates be written as

$$Q = Q^3 \otimes Q^2 \otimes Q^1 \quad (6.13)$$

The transformation can then be written, similar to the affine transformation, as

$$\mathbf{Y} = \mathbf{X} + \begin{bmatrix} Q & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & Q \end{bmatrix} \mathbf{w} = \mathbf{X} + \mathbf{I}_3 \otimes Q \mathbf{w} = \mathbf{X} + \mathbf{Q} \mathbf{w} \quad (6.14)$$

## 6.3 Minimization of the Dissimilarity Measure

Two transformation forms, which both contribute to minimizing the cost function,  $D$ , have now been outlined. The cost function can be obtained based on different features, *e.g.* mutual information, cross-correlation or the sum-of-squared differences. In this thesis, where the images to be registered all are obtained from CT scans, it is assumed that proper calibration has been performed, and as a consequence, attenuation, hence voxel intensities can be directly compared from data set to data set. The sum-of-squared-differences (SSD), which calculates the Euclidean distance between intensities of corresponding voxels in the two images, is therefore a valid dissimilarity measure for this problem. The SSD dissimilarity measure ( $D_{SSD}$ ) can now be written as in Equation 6.15, [27, 31], where  $\mathcal{T}(\mathbf{Y})$  is the transformed template image,  $\mathcal{R}(\mathbf{X})$  the reference image, and  $w$  represents the transformation parameters. It is noted that in the following section the weights are referred to as  $w$ , but corresponds to the bold  $\mathbf{w}$  written in the previous sections.

$$\begin{aligned} D_{SSD}(w) &= \frac{1}{2} \sum_{i \in N} (\mathcal{T}(\mathbf{y}(\mathbf{x}_i; w)) - \mathcal{R}(\mathbf{x}_i))^2 \\ &= \frac{1}{2} \|\mathcal{T}(\mathbf{Y}) - \mathcal{R}(\mathbf{X})\|^2 \end{aligned} \quad (6.15)$$

### 6.3.1 Gauss-Newton

In order to find the best possible registration the  $D_{SSD}$  (from now on denoted  $D$ ) can be seen as a minimization problem, where the solution is to find the values for the weights  $w$  that provide the lowest possible value. The solution can be found iteratively by use of the Gauss-Newton algorithm. The Gauss-Newton algorithm takes point of origin in the residual function,  $r(w)$ , and its derivative, the Jacobian  $J(w)$ . The residual function can be found by re-writing the cost function in Equation 6.15 to Equation 6.16 [28, 33].

$$D(w) = \frac{1}{2} \|r(w)\|^2 = \frac{1}{2} r(w)^T r(w) \quad (6.16)$$

Finding the first order Taylor approximation of Equation 6.16 gives

$$r(w + h) = r(w) + J(w)h \quad (6.17)$$

where  $h$  yields the descent direction.

By inserting the Taylor approximation in Equation 6.16 the equation system will give the descent direction,  $h$ , which minimizes the specified cost function, [28].

$$D(w + h) = \frac{1}{2} \cdot (r(w) + J(w) \cdot h)^T (r(w) + J(w) \cdot h) \quad (6.18)$$

By finding the first derivative of Equation 6.18 and setting it to zero, the descent direction  $h$  can be found. For more profound explanation and derivation the reader is referred to [28].

$$\begin{aligned} \nabla D(w + h) &= J(w)^T \cdot r(w) + J(w)^T \cdot J(w) \cdot h = 0 \\ \Leftrightarrow J(w)^T \cdot J(w) \cdot h &= -J(w)^T \cdot r(w) \\ \Leftrightarrow h &= -(J(w)^T \cdot J(w))^{-1} \cdot J(w)^T \cdot r(w) \end{aligned} \quad (6.19)$$

The Jacobian is here formed by

$$J(w) = \nabla T \cdot \mathbf{Q} \quad (6.20)$$

where  $\mathbf{Q} = I_3 \otimes Q^1 \otimes Q^2 \otimes Q^3$  contains all basis functions and  $\nabla T = [\nabla T^1 \nabla T^2 \nabla T^3]$  is given by a large matrix constructed of the image gradients for each dimension, *e.g.*  $\nabla T^1$  holds all image gradients with respect to the first dimension in its diagonal. Both  $\nabla T$  and  $\mathbf{Q}$  are of such size, that a multiplication, hence a formation of  $J(w)$  in many cases will be impossible in practice. If one could succeed to form  $J(w)$ , an inversion, as needed in Equation 6.19, will definitively rule out a possible calculation. Since a solution to this system then only exists in theory, an approximation of the system must therefore be applied in practice. This is done by use of the Truncated Gauss-Newton system, which can be solved by a conjugated gradient method. The approach of this method is to go back a step in Equation 6.19 and find an approximated solution to  $h$  on the basis of Equation 6.21, [28]. Iteratively, the Truncated Gauss-Newton algorithm will search for a minimum, in order to find the optimal  $h$ .

$$J(w)^T \cdot J(w) \cdot h = -J(w)^T \cdot r(w) \quad (6.21)$$

In the algorithm for finding the optimal descent direction,  $h$ , a number of stopping criteria are often specified, in order to prevent the algorithm from keeping running, when no proper progress is seen. When *e.g.* the variation of the cost function between two iterations is small enough the process is forced to stop. However, a *safe guard* is always included, to ensure that the algorithm will terminate, if not sooner, than after a user defined number of iterations.

### 6.3.2 Line Search

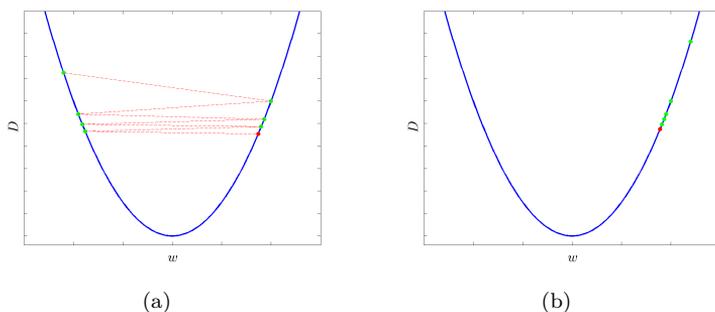
When the descent direction has been determined, the step size,  $\alpha$ , in that given direction must be decided. To estimate the step size, a line search method is used, which evaluates the cost function,  $D$ , along the defined descent direction,  $h$ . This is defined by the "line" function,  $\varphi(\alpha)$ , below.

$$\varphi(\alpha) = D(w + \alpha h), \quad \alpha > 0 \quad (6.22)$$

Solving the line search problem exactly can be rather computational costly. The solution to these kinds of problems are therefore often approximated, by minimizing the function  $\varphi(\alpha)$ . The most simple condition which must be satisfied is,

$$\varphi(\alpha_{k+1}) < \varphi(\alpha). \quad (6.23)$$

Unfortunately, this is often not sufficient to produce a proper line search, since this does not prevent the step size of being either very large nor very small, as illustrated in Figure 6.6. The decrease in the cost function could therefore not be satisfactory and a proper convergence can be difficult to reach, [28, 1].



**Figure 6.6:** Line Search with a too large step size in(a) and a too small step in (b). Idea from [20].

#### 6.3.2.1 Armijo Backtracking Line Search

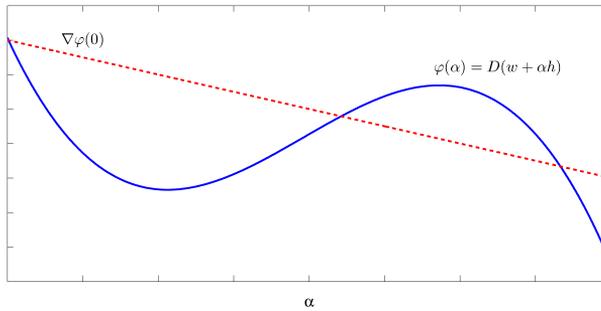
To find a more suitable way to solve the problem, a new condition is taken into consideration. The condition is often referred to as the *Armijo condition* and is defined in Equation 6.24, [28, 20, 1]. The Armijo condition takes in

the gradient, calculated from the initial condition,  $\nabla\varphi(0)$ , and forms a linear function, as illustrated in Figure 6.7 as the dotted line.  $\alpha$  values must be found under that line, in order to be accepted as an applicable step size.

$$D(w + \alpha h) \leq D(w) + \beta_1 \nabla D(w) h \alpha$$

$$\Leftrightarrow \varphi(\alpha) \leq \varphi(0) + \beta_1 \nabla \varphi(0) \alpha \quad (6.24)$$

where  $\beta_1$  is a constant given in the interval  $[0 \ 1]$ , controlling the influence of the initial gradient.



**Figure 6.7:** Illustration of the Armijo condition. The blue curve indicates the cost function, while the dotted line defines the gradient. When the cost function lies below the gradient, the corresponding  $\alpha$  values are accepted. Modified from [1].

For each iteration where the Armijo condition is not fulfilled, the value is divided by two. The algorithm runs until a value meeting the criteria, in Equation 6.24 is found. However, in order to prevent the algorithm from running without reaching an acceptable value, a stopping criterion of a maximum number of iterations is also given here.

It must be noted, that the Armijo condition does not prevent the line search to result in a very small step size, which will cause a rather slowly progress of finding the minimum. This can however be prevented by adding a further condition, called the *curvature condition*. A disadvantage is increased calculation time, since the gradient must be calculated for each new  $\alpha$  value, but the implementation of this criteria will, however, often lead to fewer iterations in the minimization process. For more profound information the reader is referred to *e.g.* [1], since only the Armijo condition is applied in this thesis.

## 6.4 Regularization

Often when working with non-linear transformations, the registration problem is ill-conditioned, hence a solution cannot be found uniquely. Furthermore no constraints are linked to the deformation field, which entails that the deformation field can work freely and thereby introduce foldings into the resulting field. To accommodate for these problems it can be an advantage to be able to control the degree of deformation, hence ensuring that the grid formation is kept. This is done by adding a regularization term  $S(w)$  to the dissimilarity measure in Equation 6.15. Different regularizers exist, but in this problem a diffusion regularizer is used, which has the ability to enforce a smooth deformation field. The diffusion regularizer defined in Equation 6.25, where  $U$  denotes the deformation field,  $\mathbf{Q}w$ , is therefore often also referred to as the smoothness regularizer.

$$\begin{aligned} S(w) &= \sum_{d=1}^3 \left( \frac{\partial U}{\partial \mathbf{x}^d} \right)^2 \\ &= \left( \frac{\partial U}{\partial \mathbf{x}^1} \right)^2 + \left( \frac{\partial U}{\partial \mathbf{x}^2} \right)^2 + \left( \frac{\partial U}{\partial \mathbf{x}^3} \right)^2 \end{aligned} \quad (6.25)$$

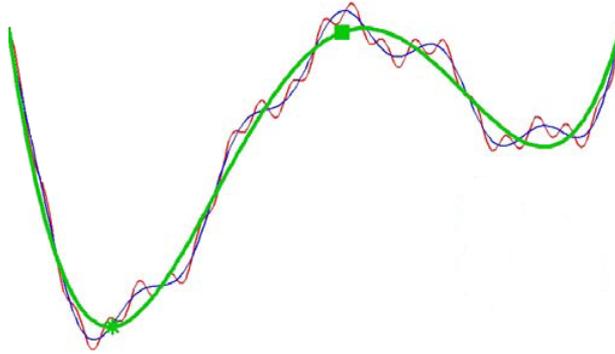
How much impact the regularizer should have on the problem can be controlled by a weight parameter  $\lambda$ . In this specific problem it is seen that, when going to a finer level, hence introducing more B-splines, the  $\lambda$  value must be increased in order to avoid foldings in the deformation field. It is noted that this effect not necessarily can be transferred to other problems. The dissimilarity measure is now in its full form defined as in Equation 6.26.

$$D_{SSD}(w) = \frac{1}{2} \sum_{i \in N} (\mathcal{T}(\mathbf{y}(\mathbf{x}_i; w)) - \mathcal{R}(\mathbf{x}_i))^2 + \lambda S(w) \quad (6.26)$$

## 6.5 Multilevel Image Registration

Since some level of noise will be present in medical images, the cost function will have a risk of getting caught in a local minimum in the minimization process. This problem can, to some degree, be avoided by introducing a multi-level solution approach. The idea of a multilevel approach is to downsample the images, often to several levels, and thereby obtain coarser levels of the same image, cf. Section 4.2. By doing this, noise components are strongly reduced,

*i.e.* smoothed, which will give significantly fewer local minima, as seen in the illustration in Figure 6.8.



**Figure 6.8:** Three curves illustrating a multilevel problem with three levels. Green = coarsest level, Blue = coarse level and Red = fine level. It is seen that the coarser the level is, the less fluctuating the curve and a significant decrease in the number of local minima is seen. Modified from [32].

By using the multilevel approach to solve the image registration problem, three favourable advantages are presented. First of all, the registration is much easier to solve on a coarse level, giving fewer iterations, and thereby strongly reduces the computation time. Secondly, the image details are decreased, which reduces the risk of getting caught in a local minimum, as mentioned earlier. Thirdly, the optimization result found on a coarser level can be used as an initial condition to the next level, which can reduce the number of iteration on the following level, [31].

The images processed within this thesis are all of the size  $512 \times 512 \times m^3$ , where  $m^3$  defines the number of slices in a given image. These are all downsampled to a four level problem, as described in Section 4.2, where  $L_3$  defines the most coarse level with a resolution of  $64 \times 64 \times \frac{m^3}{2^3}$  and  $L_0$  the finest level holding the original resolution.

As stated in Section 6.2, two different transformation forms are used, where the multilevel approach can be used. However the parameter transfer to the next level must be taken into consideration in the two situations. Due to the fact that affine transformation is solved by using the actual physical coordinates, the optimization parameters can directly be transferred to the next level. It is a little different for the B-spline transformation, where the optimization parameters are found based on a number of basic functions. On the coarsest level a number of B-splines are defined. These are at the next level increased, and again further

increased at the last level. This implies that the parameters cannot directly be transferred to the next level, but must be resampled to fit the next level.

The substantial cornerstones for image registration have now been outlined. Further implementation and optimization are discussed in [Chapter 9](#).

# Graph Cut

---

In image segmentation, surface detection is of high significance - especially in volumetric data where a 3D surface is sought. For an optimal surface segmentation, graph-based algorithms are highly utilized due to their relatively easy extension to 3D, and due to the guaranteed convergence to the global optimization. Graph-based algorithms divide nodes of a graph into two subsets, and the following chapter will describe the theory behind graph cuts and the algorithm in use. In Section 7.1, the background and graph construction is specified. Section 7.2 is based upon the algorithm used to partition a graph - the so called min-cut/max-flow algorithm by Boykov *et al.* [7]. Finally, graph construction is further considered, Section 7.3, in terms of edge weights for the topic in concern. This chapter provides a basic foundation for understanding the implementation of the graph-based segmentation in Chapter 10.

## 7.1 Background

Graph-based procedures use (among others) max-flow algorithms to minimize an energy problem. Through time they have been used on a variety of vision problems, *e.g.* image restoration and image segmentation. The history of graph cut algorithm goes back to the first min-cut/max-flow algorithm by Greid *et al.*

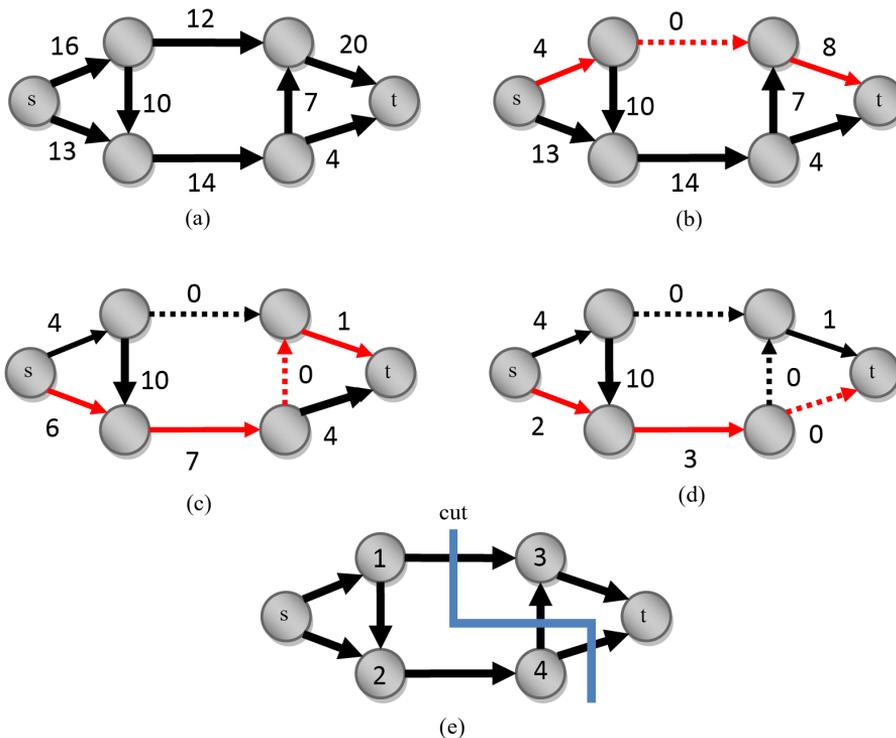
[18], where the energy framework was used to maximize *a posterior* of Markov Random Fields in order to restore binary images.

Graph construction procedures are based upon the construction of a weighted graph,  $G = (V, E)$ , with a set of nodes and edges,  $V$  and  $E$ , respectively. The nodes, or vertices,  $v \in V$ , correspond to *e.g.* voxels of an image, or of resampled versions of this. The edges, or weights, specify the cost of the connections between a neighborhood of nodes, see Figure 7.1(a) and Figure 7.3 for a simple visualization of a graph setup. Two additional nodes, called terminal nodes, are added to indicate the labels of the cut. The terminal nodes are referred to as a source,  $s$ , and sink,  $t$ , respectively, specifying each side of the segmentation, *i.e.* each side of the sought surface. The edges also have a similar classification into subgroups: t-links (to terminal nodes) and n-links (edges in between nodes). The structure of the n-links can moreover specify the graph type. A graph is said to be *undirected* when two nodes only have one connection, and *directed* when the nodes have two opposite edges with different costs associated. See Figure 7.1(a) (undirected) and Figure 7.3 (directed) for visualization of this and the following definitions.

The reference to a graph *cut*, the so called *s-t cut*, refers to the division of the graph into an  $S$  and a  $T$  part, *i.e.* nodes belonging to the source and the sink, respectively. Since this framework is to be minimized, the term *minimum cut* is used for the optimal solution. This corresponds to the cut with the minimum cost among all possible deviations of the nodes. A theorem by Ford and Fulkerson [16] states that the minimum cut can be found at the edges saturated by the maximum flow. Consequently, the min-cut and max-flow are two solutions to the same problem.

In order to propose a figurative easy interpretation of the max-flow, a pipeline analogy is usually chosen. Visualize a pipe structure as in Figure 7.1(a). Here the cost of the edges correspond to the capacity (*i.e.* width) of the pipes, and the nodes are distribution stations. The maximum flow that can be pushed through from the source to the sink is thereby limited by the different capacities of the pipes. Algorithms that finds the maximum flow of such a system are primarily divided into two subgroups - *augmenting path algorithms* and *push-relabel algorithms*. This thesis is based on an algorithm using augmenting path, and the focus in the following will therefore be kept to this type of approach. The concept behind the augmenting path is based on the simple formula: "*Find a Residual Path, decrease Residual Flow and repeat*". This is visualized in Figure 7.1(a) - (e), where the maximum flow iteratively is pushed through the shortest path with the highest possible flow from  $s$  to  $t$ . This is repeated until saturated edges (dotted lines) are restricting further flow. The maximum flow that can be pushed from the source to the sink is found to be  $(16 - 4) + (13 - 2) = 12 + 4 + 7 = 23$  (calculated based on the maximum flow from the source or the cost

of the edges being cut, respectively). The edges that are saturated restrict this maximum flow, and as stated by the theorem, the minimum cut will therefore separate node 1, 2 and 4 from node 3. In the pipe analogy, the minimum cut can be seen as the  $s$ - $t$  cut where the least amount of "water" is wasted when the pipes are cut off. This is a simple, however not fully comprehensive introduction to the concept of augmented path. The reader is referred to the work by e.g. [16, 7] for further explanations.

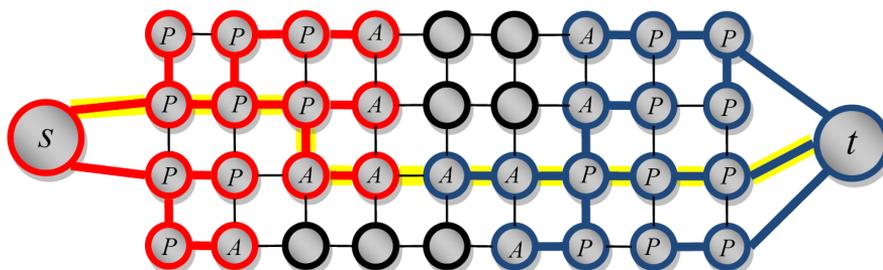


**Figure 7.1:** The concept behind the augmenting path. (a) The initial graph setup. The terminal nodes from the source set the maximum possible flow to 29. (b) The shortest path from the source to the sink with the highest possible flow is found and marked in red. The residual flow is then decreased by the cost of the edge that is saturated, here 12. (c) Repetition of the procedure results in at least one new edge saturation and a decrease of the residual flow by 7. (d) A new path is now necessary in order to saturate an edge. The flow thereby decreases by 4. (e) No further flow can be pushed through from the source to the sink due to the saturated edges. From the Ford and Fulkerson theorem the minimum cut is therefore found at these saturated edges. Idea from [9].

## 7.2 The Min-Cut/Max-Flow Algorithm

In [7] Boykov *et al.* presents a min-cut/max-flow algorithm based on augmenting path, but optimized in speed due to the introduction of a new building technique for the search trees, *i.e.* a path between the source and sink. One search tree is built from the source and one is built from the sink, visualized in Figure 7.2 in red and blue, respectively. In the following Figure 7.2 will be the base on a short overview of the algorithm and its terminology.

The two search trees are treated systematically and consists of so called *active* and *passive* nodes, labeled by  $A$  and  $P$  in the figure. Nodes with no tree connection are called *free nodes*. Active nodes have the possibility to grow to a free node, whereas passive nodes cannot grow, since they are blocked by other nodes from the same tree. For the growth process the terminology is expanded with so called *parents* and *children*, where the naming is based on the saturation of the edges. Even though two search trees are used, the flow is still considered to flow from the source to the sink, and therefore the naming differs in the two trees: A parent in  $S$  has a non-saturated edge to its children, while the non-saturated edge in  $T$  goes from a child to its parent.



**Figure 7.2:** Search tree setup for a short overview of the min-cut/max-flow algorithm and its terminology. Red nodes,  $S$ , are connected to the search tree with root in the source, whereas blue nodes,  $T$ , belong to the sink. The labeling with  $A$  and  $P$  specifies the active and passive nodes, respectively. The free nodes are marked with black. The found path from the source,  $s$ , to the sink,  $t$ , at the end of a growth stage is marked with yellow. Edited from [7].

The iterative process of the algorithm consists of the following. Keep in mind that the stages are treated systemically for both trees.

**Growth stage** Active nodes grow by converting free nodes to active children of the tree. Meanwhile active nodes are converted to passive nodes if all their neighbors have been explored and no further growth is possible. If an active node, in its growth, encounters a node from the other tree the growth state is terminated, since a path from  $s$  to  $t$  has been formed (marked in yellow in Figure 7.2).

**Augmentation stage** The found path is now augmented by pushing the highest possible flow through the path. Naturally this entails a saturation of at least one of the edges in the path. If the saturation thereby splits a child from its parent, the child is said to be an *orphan*.

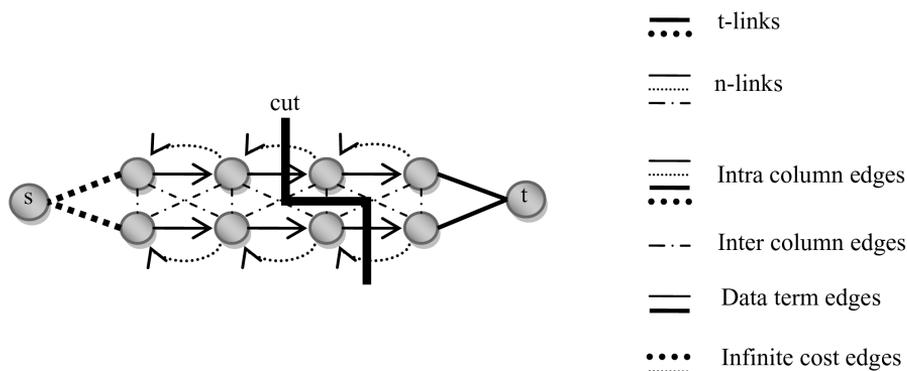
**Adoption stage** As indicated by the name, the adoption stage will assign any orphans to a new parent (through a new non-saturated edge) or assign the orphan as free node if no adoption is possible. When assigning an orphan as free, all its children are considered orphans as well and must be set up for the adoption procedure. The adoption stage is terminated when all orphans have been re-assigned.

The algorithm now returns to the growth stage and the trees are thereby expanded anew. This assures that the search trees never are built from their initial roots and entails the attractive cornerstone of the speed optimization. The algorithm terminates when no active nodes are available, the minimum cut is thereby found between the two search trees.

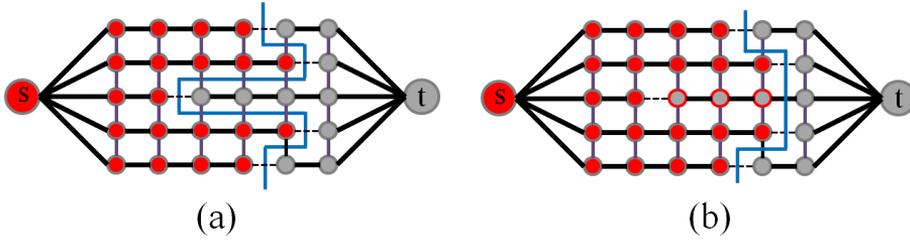
## 7.3 Edge weights

Returning to the graph construction, the edges can furthermore be divided into intra and inter column weights, which specifies the different edge costs in the neighborhood. The extent of the neighborhood depends on the number of links made to the nearest inter column nodes. The reader is again referred to Figure 7.3 for the definitions of the edges. A column, or flow line, is a direct path through intra column nodes from the source to the sink, where the sought surface must be cut at least once. The inter column edges specify connections to nodes in another nearby column. The flow (pipe-analogy) from node to node within a column is controlled by the data term edges,  $E_{data}$ , based on an image or *e.g.* gradient values, cf. Section 10.5. In order to ensure a unidirectional flow

in the column and to avoid multiple cuts within a column, the cost of the edges opposite the data edges are set to infinity,  $E_\infty$ . Considering the inter column edges,  $E_{inter}$ , these edge costs can be seen as separation penalties, enforcing a smoother surface solution. This effect is simply visualized in Figure 7.4, where (a) shows the cut (blue line) when the inter column weights are low and (b) the cut forced by a higher inter column edge cost. Thick black lines indicate high data term edges, whereas the dashed thin lines have a low edge cost. For the purpose of this simple illustration, a value of 10 and 1 is assigned to the high and low data term edges, respectively. In subfigure (a) the dark purple inter column weights are set to 1, whereas they in (b) are set to 2. The two minimum cuts that could be considered are superimposed on the two figures, and referred to as path 1 and path 2, respectively. With an inter column cost of 1, the maximum flow for path 1 is 13, whereas it for path 2 is 16. Path 1 is therefore the minimum cut for the graph in (a). If the inter column cost is increased to 2, the maximum flow for path 1 is 21, whereas it for path 2 is 18. A higher inter column edge cost has therefore enforced a smoother cut as seen in (b).



**Figure 7.3:** Nodes and edges in an  $s-t$  cut. The terminology of links and edges are explained with different line types. The t-links are edges to terminal nodes (thick lines), whereas n-links connect the nodes (thin lines). Intra column edges link the direct flow from the source to the sink (full and dotted lines), whereas the inter column edges connect to nodes in another column (dot-dash lines). Data term edges have costs based on the data set (full lines), and infinity cost edges facing backwards are applied to ensure a unidirectional flow from  $s$  to  $t$  (dotted lines). With these two opposite edges, the graph is said to be a directed graph.



**Figure 7.4:** The effect of the inter column weights. For the purpose of illustration, a value of 10 and 1 is assigned to the high (thick black lines) and low (black dashed lines) data term edges, respectively. In subfigure (a) the dark purple inter column weights are set to 1, whereas they in (b) are set to 2. The two optimal minimum cuts are superimposed on the two figures. A smoother cut is seen to be enforced in (b) due to the higher inter column cost. The nodes marked with a red contour are thereby assigned to the source.

To recap, the cost function for the edges consist of the energies specified in Equation 7.1. It should be noted that the construction of the edge cost and framework is inspired by Pedersen *et al.* [37]. For further consideration about the implementation of a graph cut segmentation algorithm, cf. Chapter 10.

$$E = E_{data} \cup E_{\infty} \cup E_{inter} \quad (7.1)$$



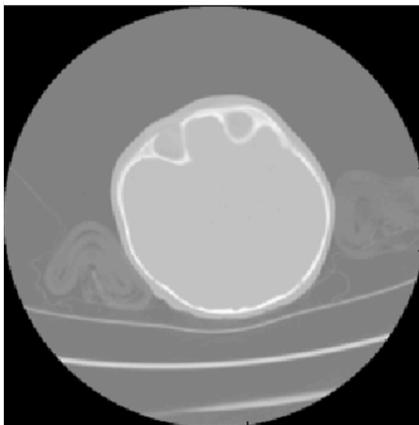
# Preprocessing

---

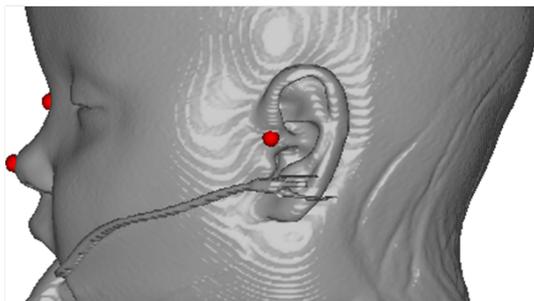
The images processed in this thesis were all made available by the 3D-Laboratory. All images were raw images, which means that they can be rather hard to analyze directly, due to different noise components present in the images, as seen in Figure 8.1. In order to solve this problem all images have been preprocessed by the 3D-Laboratory, in order to make them easier to interpret and process further by the two segmentation models. This has been done by performing two preprocessing steps, where the images have been oriented to a standard orientation and the noise components, *i.e.* everything else than the head region, have been removed or strongly reduced. The steps performed by the 3D-Laboratory are briefly described in Section 8.1 and Section 8.2. Furthermore, in an attempt to optimize the initial condition for the models, the images have been aligned as described in Section 8.3.

## 8.1 Standard Orientation

The orientation to a standard orientation is performed in the program *Landmarker*, where a rotation of the head based on four landmarks is performed. The four landmarks are placed on the following locations: right ear, left ear, the nasion and the tip of the nose, as illustrated in Figure 8.2.



**Figure 8.1:** An example of a transversal slice of the original data. The head is seen in the middle in light gray. The darker gray circle indicates the actual scan tube, wherein two slightly visible gray areas at each side of the head is stabilizing objects. The lines in the bottom indicate the scan bed.



**Figure 8.2:** An illustration of the landmark locations (red dots). The landmarks are superimposed on a polygonal surface rendering based on the HU value of skin.

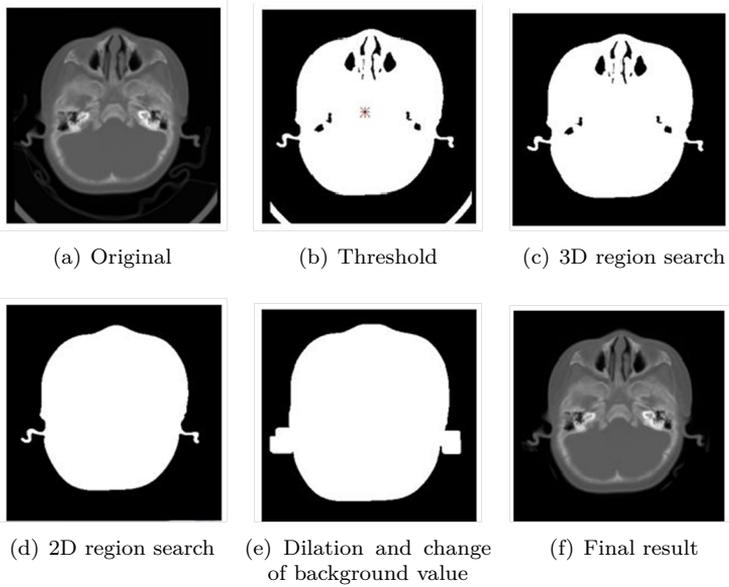
A transformation is calculated, which rotates the data to the standard orientation. This is conducted through a coordinate system incorporated in *Landmarker*. More profound information about the rotational calculation, can be found on the homepage for the 3D-Laboratory, [3].

## 8.2 Removal of Noise Components

A lot of components, *e.g.* scan bed, scan tube and headrest, as seen in Figure 8.1, can distract the analyzer from the actual area of interest. These components

are therefore removed best possible through a procedure performed by the 3D-Laboratory. The following steps, including the illustration in Figure 8.3(f), are based on a mail correspondence with Tron Darvann, from the 3D-Laboratory.

- A threshold, based on the original image, Figure 8.3(a), corresponding approximately to the value of the skin, is applied to the image. This gives a binary image where the background, corresponding to air, has the value zero, and everything else is represented with the value 1, Figure 8.3(b).
- A seed is placed in the center of the head, and a region search in 3D is applied, which collects all voxels having the same value as the seed. In this way the head is found, except a few holes, which are air filled regions in the head, Figure 8.3(c).
- A seed is then placed in the corner of each 2D slice and a similar search is conducted. Since the seed is not connected to the air filled holes in the head, the head is now given as one interconnected area, Figure 8.3(d).
- The head region is dilated a bit, to ensure that the entire area of interest is presented in the final image, Figure 8.3(e).
- Every value outside the filled area are given a constant value - here the minimum value from the original image. This value is in fact found outside the actual scan area (black area in Figure 8.1). It could be argued, that a more correct value would have been the value corresponding to air, which is found in the dark gray area in the same figure.
- Finally, the areas within the dilated head, are filled with image values gained from the spatially corresponding voxels in the original image. Small noise components can still be seen present in the image, due to the dilation.



**Figure 8.3:** Preprocessing steps in removal of background noise. The steps are further elaborated in the text.

## 8.3 Optimizing the Initial Condition

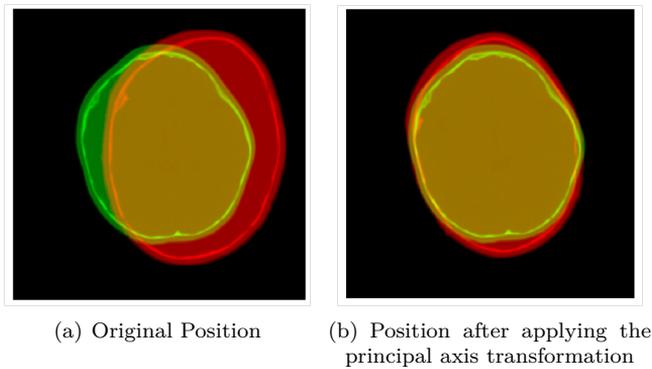
For both segmentation models, a good initial condition is important. For the image registration, a good initial condition is for the two images to lie around the same center, since a certain overlap is necessary. This will give the transformation algorithm the best possible conditions, and the chance of obtaining a satisfactory result within reasonable time frame, increases. For the graph cut model, a similar initial condition for all images will be an advantage, since this will simplify the placement of the sink and source. In order to obtain such a condition, the principal axis transformation, described in Section 8.3.1 was implemented and evaluated. This transformation was however replaced by a more simple linear transformation described in Section 8.3.2.

### 8.3.1 Principal Axis Transformation

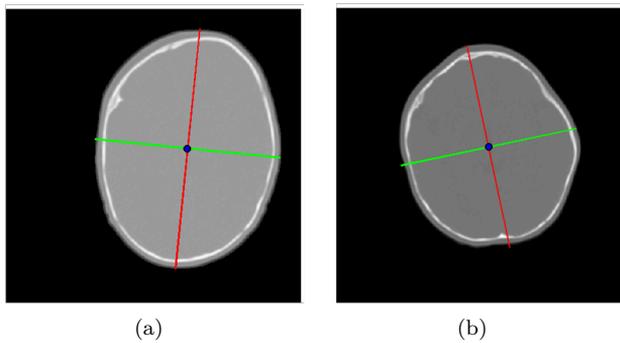
The principal axis transformation (PAT) is a similarity transformation, which uses translation, rotation and anisotropic scaling to align two images, a template and a reference, respectively. These parameters are based on the center of mass and the co-variance of the images, which are calculated from voxel intensities. The difference between the two centers of mass gives the translation parameters. From the co-variance matrix the eigenvalues and their corresponding eigenvectors are found. The eigenvectors compose the principal axes of the images, by finding the difference between the two corresponding matrices of eigenvectors, the rotation matrix, hence the rotation parameters can be found. Lastly, by comparing the size of the coupled eigenvectors, the scaling factors can be established.

A 2D example is illustrated in 8.4(a), where the reference image (green) is shown together with the template image (red). It is clear to see that a better alignment of the two images is possible. PAT is therefore applied, the principal axes and the center of masses are found for the two images, as seen on Figure 8.5. Based on the found information, the translation, rotation and scaling parameters are applied to the template image and the result can be seen in Figure 8.4(b), where an improved alignment is achieved.

Unfortunately, a problem however arises in the alignment process, due to the direction of the found eigenvectors. The eigenvectors defining the principal axes can be given as a positive or a negative direction. This entails that if the eigenvectors for the template image are found to be in the positive directions and one or both eigenvectors for the reference image are found to be in the negative direction, a rotation. A simple illustration is given in Figure 8.6, where (a) and

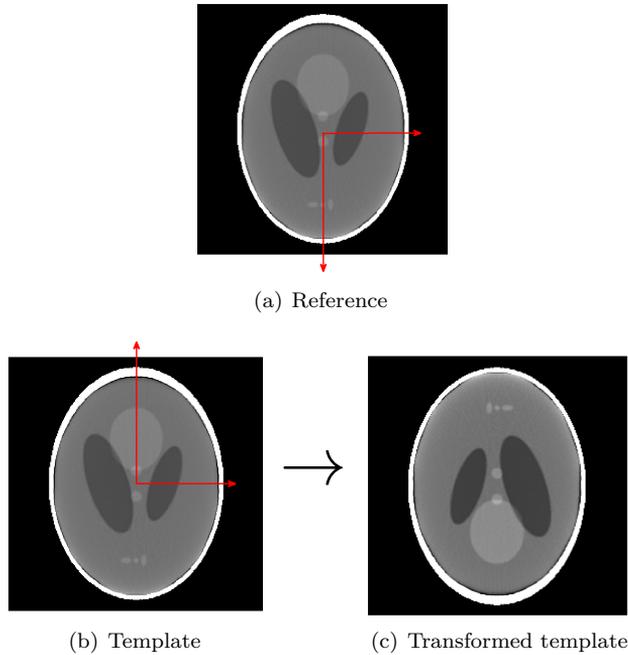


**Figure 8.4:** Two transversal CT slices from two different datasets. The reference (green) and template image (red) seen before and after application of the parameters from the principal axis transformation. The RGB image space is used in order to visualize both images concurrently.



**Figure 8.5:** Reference and template image, with principal axis and center of mass (blue dot). The first principal axis is visualized in red and the second in green.

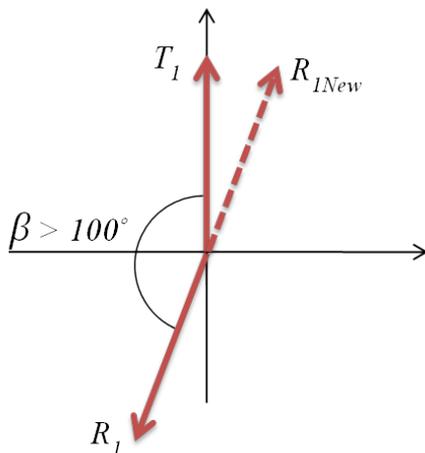
(b) show the reference and template image, respectively, together with its found principal axes. The corresponding vectors resulting from the first principal axis are seen to be found with opposite operation sign, which entails the 180 degree rotation around seen in (c).



**Figure 8.6:** Simple illustration of a reference (a) and template (b) image, where the first principal axis for the reference image is found in the negative direction, hence opposite its corresponding vector from the template image. This will give a 180 degree rotation as seen in (c).

The problem can be solved by evaluating the angle between the corresponding vectors. Based on a priori knowledge concerning object placement in the scanner, and due to the preprocessing part concerning standard orientation, it is presumed that an angle above 100 degrees between corresponding vectors, only will be seen in cases where opposite operation signs are present. Therefore, if an angle is found above 100 degrees, one of the eigenvectors is multiplied by minus one, which gives a change in direction, as seen in Figure 8.7, and thereby prevents a rotation of the image.

Unfortunately, when expanding the PAT to the 3D problem of this thesis, more difficulties arise. Due to the abnormal head shapes of these children it is no longer given that the principal axes are found in the same order for all images. If example the principal axes for two images are found as stated in Table 8.1, a transformation with these underlying eigenvectors will induce a erroneous rotation around the  $z$ -axis. In addition to the vector direction problem addressed in the 2D case, a problem concerning the order of the principal axes is now also introduced. The transformation has been applied on several images, in order to



**Figure 8.7:**  $T_1$  and  $R_1$  defines the principal axis of a template and reference, respectively. The angle  $\beta$  between the two vectors is seen to be above 100 degrees, and the  $R_1$  vector is therefore multiplied by minus one, which gives the vector  $R_{1New}$  illustrated with the dotted arrow. The rotation angle between  $T_1$  and  $R_{1New}$  will now provide a correct alignment.

create an overview regarding the extent of the problem. However, due to the variation seen among these children, a simple solution to address this problem has not been established. Instead the principal axis transformation is discarded and a new, more simple approach, described in Section 8.3.2, is used.

**Table 8.1:** Example of the directions principal axes found in two 3D images.

	Image 1	Image 2
1. principal axis	$z$	$z$
2. principal axis	$y$	$x$
3. principal axis	$x$	$y$

### 8.3.2 Linear Translation

Instead of aligning one image to the other, all images are now moved to a common center, here chosen as the geometrical center of the image. The region of interest is then centered by applying a simple linear transformation, where only translation is present. The center of mass is found, but only the center of masses for the first two dimension,  $x$  and  $y$  are used. The datasets are seen

---

to differ significantly in their  $z$  dimension, not only in the number of slices and resolution, but more importantly in their scan area, hence *e.g.* the amount of shoulder present in the image. As a consequence the center of mass in that direction cannot be used directly, the translation in  $z$  is therefore neglected, and addressed as zero. The center of mass in the other two dimensions can also be slightly biased by the shoulder region, but assuming symmetry in that area, this is left out of account. The translation parameters for the two needed dimensions are then found as the difference between the center of mass and the geometrical center of the image. By applying the found translation to the image, the region of interest is moved to the geometrical center in the  $x$ - $y$  direction. The images are now ready for the segmentations described in Chapter 9 and Chapter 10.



# Implementation and Optimization - 3D Image Registration Based Model

---

The image registration was performed by use of a software package, called RegLab4matlab, developed by Martin Vester-Christensen [44], which takes point of origin in a similar software program called FAIR,<sup>1</sup> produced by Jan Modersitzki, [33]. Both programs are built on the theory described in Chapter 6. The software package has furthermore been used by Hildur Ólafsdóttir, who has added a couple of files, which have been helpful during the work of this thesis. In order to have the software running on this data, modifications of some of the files were though necessary. The software is enclosed in the attached CD. The main program files were extended with explanatory comments, which were added through the acquaintance and review of the program. Since the software is based on object based programming, no overview of the program files is supplied.

In Section 9.1 the implementation process is briefly outlined. Since an already existing software package was utilized for this model, the implementation description is kept on a basic level, providing the reader with a general introduction

---

<sup>1</sup>Flexible Algorithms for Image Registration

to the model. In Section 9.2, the results and considerations regarding the different optimization attempts, concerning levels and regularization parameters, are described.

## 9.1 Implementation

### 9.1.1 Background Correction

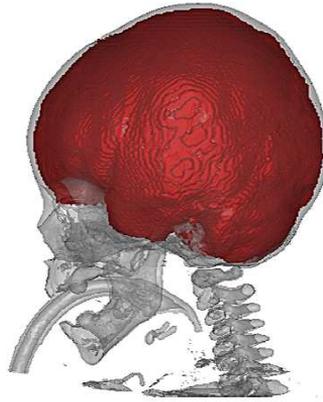
Before the images could run with the available software, a correction of the intensity of the background voxels, was necessary. In the preprocessing the noise components were removed from all images, and in that process the background values were set to the minimum value found in the original image, which was seen to lie between -7000 and -1500 HU in between the datasets. This induced problems in the image registration process, as the large difference between the background voxels for the two images, was seen as registration errors, due to a high squared errors for these calculations. The background values are therefore corrected to a value of -1000 HU, which approximately correspond to the air value in CT scans.

Difficulties were though still present in the first transformation process, where no proper convergence was seen. The extent of the problem differed dependent on the size and the transversal placement of the reference image, compared to the template. A correction of an equal background value to -1000 HU was seen insufficient, in order to obtain an alignment. The squared error in a boundary region could *e.g.* be calculated between a voxel containing air (-1000 HU) and a voxel containing soft tissue ( $\sim 30$  HU). This might be the most "correct" match, but the squared error would indicate differently. In order to solve the problem, all background values were further adjusted to a value of 0 HU, which would lower the difference between the background and object voxels, hence lowering the squared error. This adjustment was seen to yield significant better conditions for the registration process.

Another approach could have been to apply a mask to the registration problem, such that all values under a certain constant were neglected in the calculation of minimizing the cost function. In this way, adjusting background voxels could have been avoided.

### 9.1.2 Software Description

The main aim in image registration is to find the optimal transformation parameters, which can provide the best possible registration of the two images. However, it must be remembered, that in this thesis the main aim was to estimate the intracranial volume of the reference image. In order to reach that goal, a template image with a known intracranial volume must exist, cf. Chapter 5. A template with a reasonable manual segmentation, but more importantly with a high resolution, was selected among the 15 available segmented datasets. Dataset #14 was selected, and its intracranial volume segmentation, found from the manual segmentation, can be seen in Figure 9.1.



**Figure 9.1:** 3D surface rendering of the template image, dataset #14, with its intracranial volume found from manual segmentation.

The process for obtaining an estimation of the desired volume mask for the reference image is in a simple manner illustrated in Figure 9.2. Image registration was first performed between the preprocessed CT images for the template and reference image. The registration was performed by transforming the template image by finding a set of transformation parameters  $w$ , as described in Section 6.2.1 and Section 6.2.2, and illustrated in (b) and (c). The parameters were found by iteratively solving the minimization problem outlined in Equation 6.26. When the process reached an acceptable minimum, *e.g.* when the change in the cost function was under a certain level, the process was said to have found its solution. The parameters providing the final transformation, referred to as the optimal weights,  $w_{opt}$ , was then extracted and applied to the volume

mask of the template. In this way the volume mask from the template was equally transformed, which provided an estimation of the volume mask of the reference image. This last step is illustrated in (d) and (e). By multiplying the new volume mask with the voxel size from the reference image, an estimation of intracranial volume of the reference image was obtained.

## **9.2 Optimization**

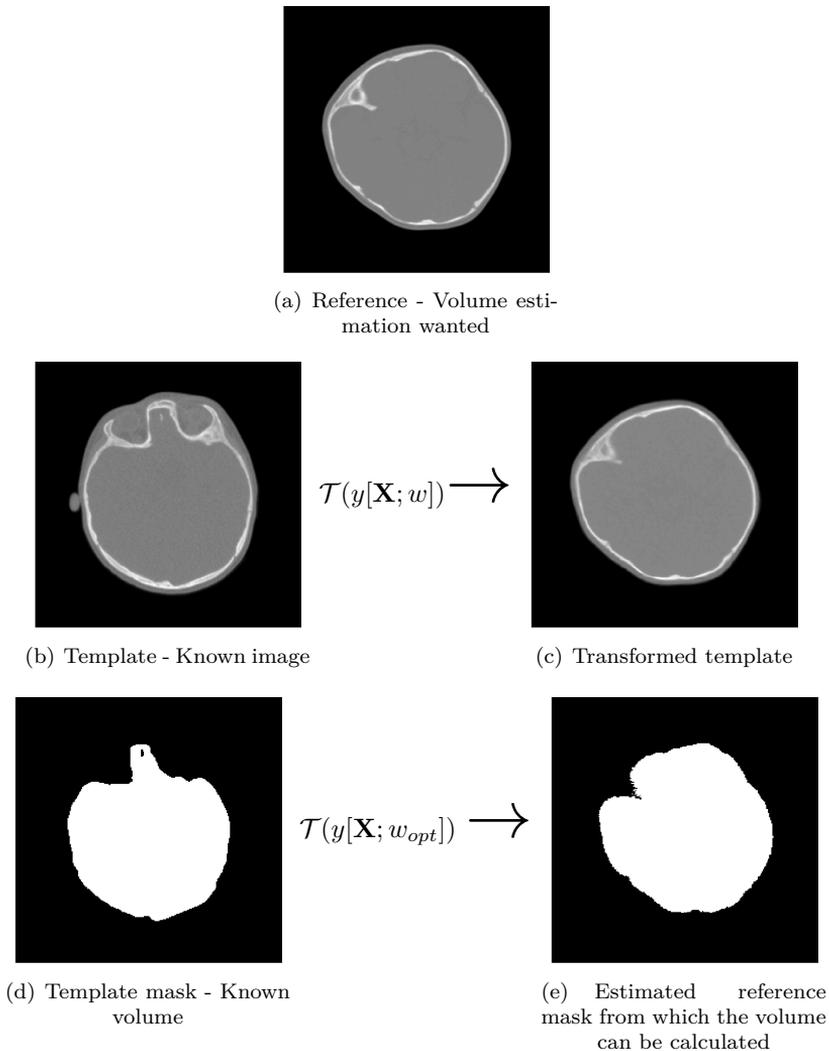
In image registration a number of parameters can be adjusted, both regarding the transformation forms, where the weights controlling the regularization might be the most significant parameter. However, also the initial condition for the line search, choice of line search method and stopping criteria can affect the result. In this situation the line search method and the stopping criteria were kept as in the original software package. The initial condition for the line search was seen to provide satisfactory results so no adjustment of that parameter was conducted. Instead, due to the size of the images, the calculation time was seen very high and relatively demanding, when the transformation was applied on all levels. A simple test was therefore conducted in order to see whether it would be possible to eliminate some of the level steps. Furthermore, adjustments of the weight parameters controlling the regularization of the B-spline transformation were performed. The optimization was evaluated against the resulted voxel deviation, found as the amount of voxel differences between the manual and model based segmentations.

### **9.2.1 Levels**

The transformation conducted in the image registration software, is built as a two step constellation. First, the affine transformation is conducted on the levels specified by the user. 12 parameters are obtained for each level, which all are saved in a matrix constellation. Subsequently, the B-spline transformation is conducted, again on the user specified levels for this transformation form. The parameters found from the affine transformation were given as an input to the B-spline, where they are used as initial conditions for the deformation grid.

#### **Affine Transformation**

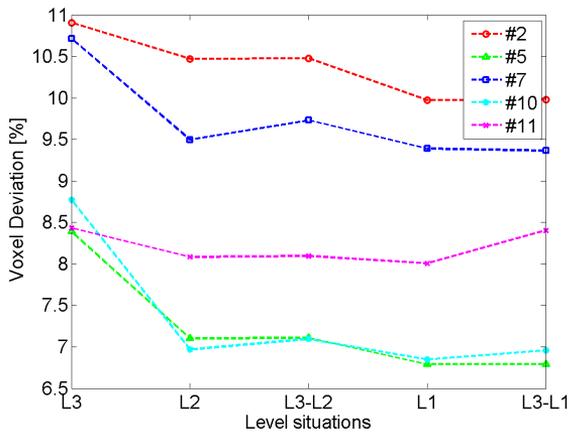
The purpose of the affine transformation was to align the template and reference image best possible by use of translation, rotation, shear and scale. This is a



**Figure 9.2:** Illustration of an image registration procedure. The top image shows the reference image, for which intracranial volume information is wanted. In order to obtain the wanted information, a known template image is registered to the reference. This is conducted by iteratively minimizing the sum-of-squared differences between the template and the reference image. For each iterative step a parameter set,  $w$ , is obtained, providing the information for the given transformation. When the process reaches its minimum, hence its optimal solution, defined on the basis of a number of stopping criteria, the parameters providing the final transformation are referred to as optimal,  $w_{opt}$ . These parameters are extracted, and by applying  $w_{opt}$  to the mask of the template, which is known, the intracranial volume for the reference image is found.

## 74 Implementation and Optimization - 3D Image Registration Based Model

relatively simple transformation, where reasonable results can be expected on relatively coarse images. A study of the effect of the affine transformation on the different levels was therefore performed, in order to examine whether it was possible to eliminate some of these transformations levels. The study was performed on five different level situations, where the results, defined by means of the percentage voxel deviation, are seen in Figure 9.3. When more levels are specified, *e.g.*  $L_3-L_1$  the transformation was applied according to the multilevel approach described in Section 6.5, starting with the coarsest level. Five randomly chosen datasets constitute to the stated voxel deviations, and the test results are therefore only an indication of a global effect. Furthermore, the calculation times in Table 9.1 are presented as a mean of the five datasets. It should be noted that the first three situations, hence  $L_3$ ,  $L_2$  and  $L_3-L_2$  were run on a 64-bit laptop with a 2GHz Intel Dual Core i7 processor and 6 GB RAM. This was seen inadequate when working with higher levels.  $L_1$  and  $L_3-L_1$ , are therefore run on the the Sun Fire X4600 M2 high memory server at IMM, consisting of eight 2.3 GHz CPUs (Quad-core AMD Opteron<sup>TM</sup> processor 8356) and a memory capacity of 256 GB RAM (64 blocks Samsung DDR2 Synchronous 333 MHz). The processing times are therefore not directly comparable.



**Figure 9.3:** Illustration of voxel deviation after affine transformation as a function of five level situations.

**Table 9.1:** Mean time consumption for the five level situations when applying the affine transformation.

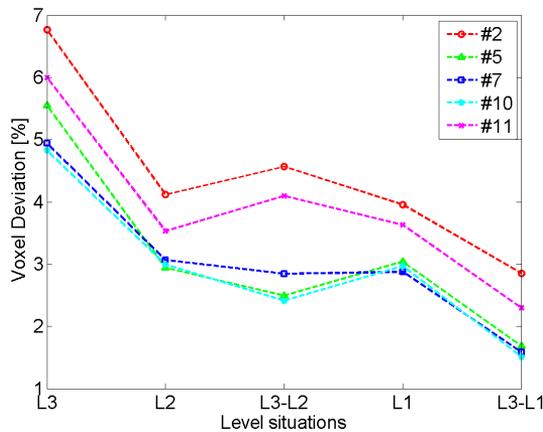
	$L_3$	$L_2$	$L_3 - L_2$	$L_1$	$L_3-L_1$
Mean time [s]	7.13	56.26	65.15	1595.16	1609.83

Based on the results from Figure 9.3 a decrease is seen for all five datasets when

going from  $L_3$  to  $L_2$ . However, the voxel deviation has no significant decrease when conducting the affine transformation on the finest level  $L_1$ . Similar, the voxel deviation is not seen to decrease significantly, when running on several levels. Another aspect which can be taken into consideration is the fact that conducting the affine transformation on a finer level will demand a powerful computer with a large amount of RAM. Moreover, the calculation time, presented in Table 9.1 increases significantly for finer levels. Based on these observations, the affine transformation was limited to be conducted only on  $L_2$ .

### 9.2.1.1 B-spline Transformation

The B-spline transformation is a transformation with the ability to correct for even small local dissimilarities between the template and reference image. Since the images on their coarser levels will contain fewer details, and thereby appear more "smooth", than their finer levels, cf. Section 6.5, a similar level limitation, as seen for the affine transformation, is not expected to be possible. However, a small study was conducted to investigate how the change in voxel deviation depends on the levels chosen for the B-spline transformation. The results are seen in Figure 9.4 together with mean calculation times in Table 9.2. It is noted that an affine transformation on  $L_2$  was applied prior to the B-spline transformation, and that the boundaries for the B-splines are free.



**Figure 9.4:** Illustration of voxel deviation after B-spline transformation as a function of five level situations. An affine transformation on  $L_2$  has been applied to all cases prior to the B-spline transformation.

Based on these results, it is evident, that in order to reach the best result, the B-

## 76 Implementation and Optimization - 3D Image Registration Based Model

**Table 9.2:** Mean time consumption for the five level situations run with the B-spline transformation.

	$L_3$	$L_2$	$L_3 - L_2$	$L_1$	$L_3-L_1$
Mean time [s]	6.82	32.57	41.47	694.00	613.37

spline transformation must be conducted on all three levels, hence  $L_3-L_1$ . This entails that all datasets are processed with an affine transformation conducted on  $L_2$  followed by the B-spline transformation conducted on the three levels,  $L_3-L_1$ , where the affine parameters from  $L_2$  are given as input to all three levels. It is noted that  $L_0$  was not included in the studies due very long processing times and no certain improvement in performance. Future studies could, however, assess this applicability.

### 9.2.2 Regularization Parameters

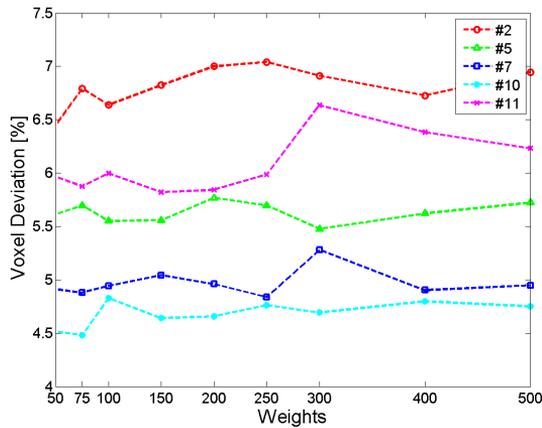
The B-spline transformation was applied to the registration problem by adding a number of B-splines equally distributed over the voxel grid. The density of the spline distribution differs depending on the levels, *i.e.* the sampling density. In order to have the same distribution in each image, the density distribution of B-splines was made independent of the voxel dimensions, by defining the distribution in millimeter. The millimeter distribution for the three levels was 20 mm, 10 mm and 3 mm from coarse to fine level, respectively<sup>2</sup>. A regularizer controlling the degree of flexibility of these deformation grids was added to the cost function. How much weight the regularizer on the given levels is assigned, can have an impact on the final result, *e.g.* if too much flexibility is allowed, foldings can appear in the image and introduce errors.

In order to have the B-splines work as effective as possible, an "optimal" weight for the regularizer for each level must be established. The two above studies were performed with a set of weight parameters for the three levels, at 100, 6000 and  $10^3$ , respectively, found from a quick study based on only two datasets. A more thorough study was therefore performed, involving the five dataset and the level setup established from the two above studies. The weight parameters were found one at a time, starting with the one controlling the coarsest level. Nine different parameters, [50 75 100 150 200 250 300 400 500] were tried and evaluated based on their resulted voxel deviation. The results for the nine parameters, for the five dataset, are outlined in Figure 9.5, where no unison minimum can

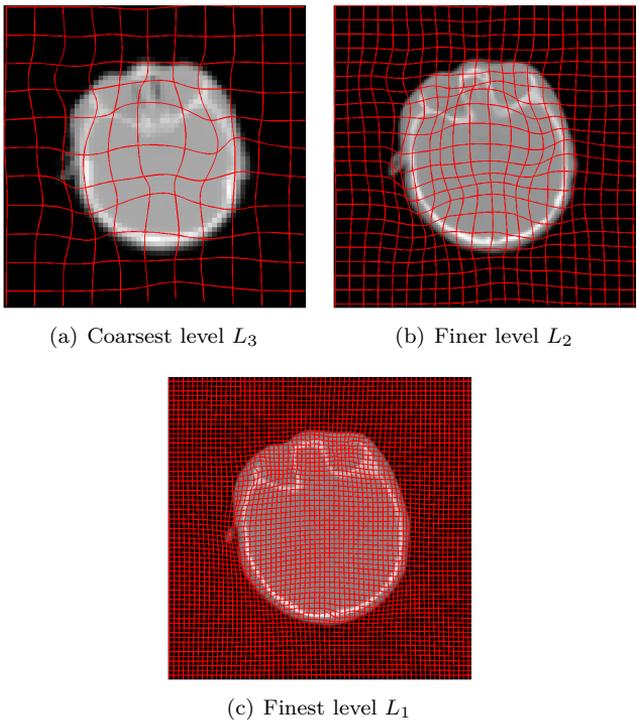
---

<sup>2</sup>Based upon inspiration and conversation with PhD student, Mark Lyksborg and MSc. Eng., PhD, Hildur Ólafsdóttír

be established. In order to examine the results closer, the deformation grids were evaluated, which showed no warp for the nine tried weights. A low weight, at 150, was therefore chosen in order to allow a relatively flexible grid on this coarse level. Similar studies were conducted for  $L_2$  and  $L_1$ . First step was to find the weight level, where no warp was present, and then examine values in that area. Similar to the results from the first level, minor changes in the voxel deviations were seen, but no unison minimum could be established. The weights for  $L_2$  and  $L_1$  were therefore chosen as low as possible, where no warp was detected, which resulted in the following regularizers;  $2.5 \cdot 10^3$  and  $5 \cdot 10^4$ . An illustration of the resulting deformation grids on the three levels, together with the transformed template, is seen in Figure 9.6. Whether these weight parameters should have been established before the level test, could be argued. However, since no significant changes were seen in the voxel deviations among the examined weights, this was not further investigated.



**Figure 9.5:** Voxel deviation after affine transformation on  $L_2$  followed by the B-spline transformation performed on  $L_3$ - $L_1$  as a function of nine different regularization parameters for five different datasets.



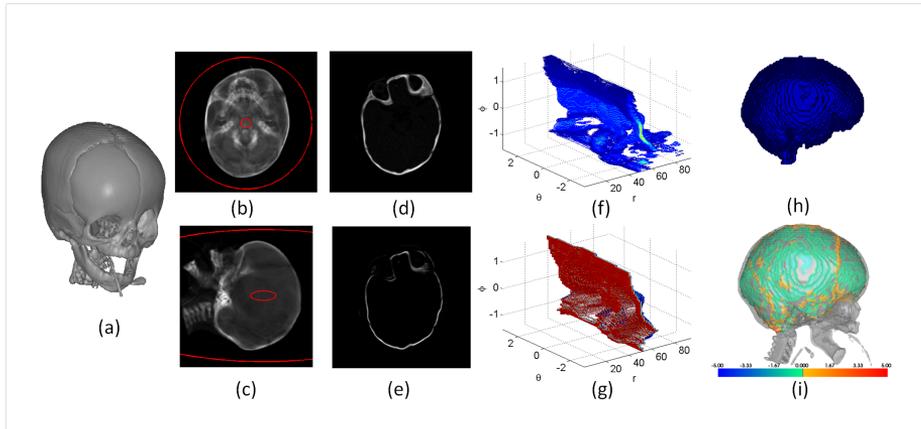
**Figure 9.6:** Deformation grids for the B-spline transformation on the three levels.

# Implementation and Optimization - 3D Graph Based Model

---

The following chapter will provide an insight to the implementation of the segmentation process built upon the graph cut algorithm described in Chapter 7. Section 10.1 describes the preprocessing specific for this graph cut procedure. Note that a simple walk through of the graph cut based segmentation algorithm is seen in Figure 10.1. The figure is used throughout this chapter for reference and overview. Section 10.2 states the construction of the source and sink, seen in Figure 10.1(b)-(c), and in Section 10.3 the radial gradient is defined (Figure 10.1(e)). The used spherical resampling is defined in Section 10.4, which results in the resampled gradient in Figure 10.1(f). The setup for the graph cut algorithm is defined in Section 10.5 and Section 10.6, where the weights and connections are investigated. This entails the graph cut in spherical coordinates as seen in Figure 10.1(g). Finally, the conversion back to Cartesian coordinates is defined in Section 10.7, allowing the final segmentation to be made, as in (h). The found segmentation can now be held up against the manual segmentation and evaluated as in Figure 10.1(i). The code written in MATLAB version 7.13 is provided in the enclosed CD. Note that in order to process 3D CT data with a resolution of  $256 \times 256 \times m^3$  (corresponding to  $L_1$  of the downsampled dataset, see Section 4.2), the algorithm must be executed on a high memory server. All

calculation were performed on the Sun Fire X4600 M2 high memory server at IMM, consisting of eight 2.3 GHz CPUs (Quad-core AMD Opteron<sup>TM</sup> processor 8356) and a memory capacity of 256 GB RAM (64 blocks Samsung DDR2 Synchronous 333 MHz).



**Figure 10.1:** Walk through of the graph cut implementation. (a) 3D surface visualization based on the HU bone value. (b)-(c) Collapsed image with the source and sink superimposed in red. (d) Transversal slice of the head. (e) Transversal slice from the 3D radial gradient. Only positive gradient values are of interest. (f) Resampled version of the radial gradient in spherical coordinates. (g) The cut found in the spherical coordinates. (h) Resulting graph-based segmentation. (i) Manual segmentation with superimposed closest point deviation [mm] found between the manual and graph-based segmentation.

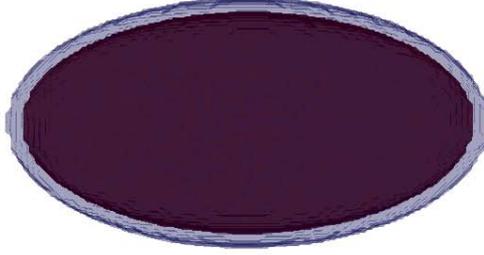
## 10.1 Preprocessing

As described in Section 8.3.2 the heads are linearly transformed in the  $x - y$  direction. For the purpose of graph cut, this translation is performed in order to facilitate an easy placement of the source, the sink, and the center of the sampling grid. The segmentation is performed on images downsampled once, *i.e.* with a dimension of  $256 \times 256 \times m^3$ , where  $m^3$  is the variable number of slices. The downsampling is necessary, in order to decrease the memory capacity required to create the extensive weight matrix. It should furthermore be noted that the images are windowed for HU values below 20, since no gradients in these regions are of any interest.

## 10.2 Construction of the Source and Sink

The source and the sink are chosen as ellipsoids in order to mimic the sought surface best possible, with the simplest possible shape. The source must be placed inside the skull and the sink on the outside, facilitating the minimum cut to encapsulate the entire skull. Since the preprocessing, *i.e.* the linear transformation, only provides a suitable center for the source and the sink in the  $x-y$  dimension, the center in the third dimension must be manually selected. The segmentation is no longer fully automatic, but a user-specified  $z$ -center is necessary due to the difference in scan area of the datasets. The user must moreover specify the radius of both the source and the sink, and verify that the placement is correct in all dimensions. By creating terminal weights with infinitely high weights outside these boundaries (and zero in between), the cut is forced to lie between the two boundaries. It should be noted that the radius specified by the user is set in an isotropic image, but when converting to the anisotropic voxel size, the source and sink are seen as ellipsoids. It was decided to place the source and the sink on the basis of a collapsed version of the CT image, to facilitate an easy interpretation of the anatomical placement of the boundaries, cf. Figure 10.1(b)-(c). Another way would be to place the source and the sink based on the spherical resampled visualizations.

Due to the incorporation of the manual input, the effect and placement variation dependency has been tested. This was done on a test set created as a simple 3D double ellipsoid, in order to simply mimic the craniofacial anatomy, see Figure 10.2. The test data consists of an inner ellipsoid with an assigned HU value at 30, and an outer ellipsoid band with a HU value of 900, which should be interpreted as the brain and the skull, respectively. The background values are furthermore set to the HU value for air. The voxel size is defined isotropic, which will lead to a spherical sink and source. Together with the elliptic shape of the test set, the algorithm will still be challenged, since the sought cut is a non-plane surface. The placement and spread of the source and the sink was varied and the effect on the segmentation was evaluated compared to the true known boundary of the ellipsoid. No significant change in the error value was seen when expanding the spread of the source. The location can however be placed too close to an edge, which will cause the gradient unit vectors to be misleading. It is thereby concluded, that as long as the  $z$ -location is placed around the center of the skull, no crucial error will be induced by the user. Alternatively, a future feature could be a quantitative evaluation of the correctness of the user specified center, or a complete automatic center placement based on extracted skull dimensions. Note that similar center dependent results were seen, when performing the test on a real dataset.



**Figure 10.2:** Surface representation of the 3D test set consisting of an inner (red) and an outer (blue) ellipsoid.

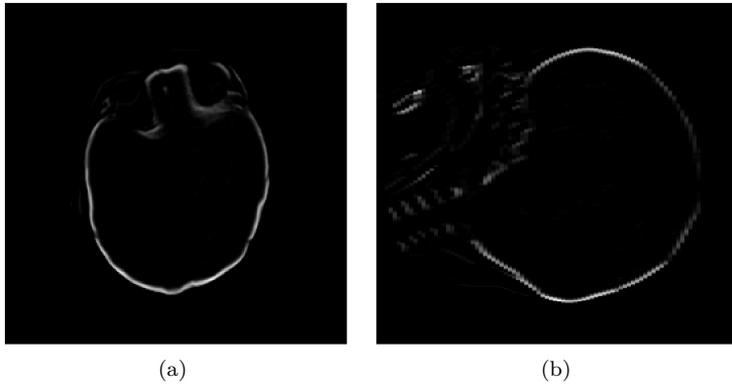
### 10.3 Radial Gradient

The edge weights,  $E_{data}$  are found on the basis of the radial gradient image, which in 3D is calculated from Equation 10.1, where  $\hat{\mathbf{r}}$  is the radial unit vector, with the geometrical image center in  $x$  and  $y$ , and the user specified center of  $z$ . By multiplying with normalized radial vectors the gradient image is weighted equally in all directions. The voxels of the dataset are an-isotropic, and the unit vectors in the  $z$ -direction is therefore scaled by the dimension ratio,  $sca$ , as seen in Equation 10.2.

$$\frac{\partial I}{\partial \mathbf{r}} = \begin{pmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial z} \end{pmatrix} \cdot \hat{\mathbf{r}} \quad (10.1)$$

$$\frac{\partial I}{\partial \mathbf{r}} = \frac{\partial I}{\partial x} \cdot \hat{r}_x + \frac{\partial I}{\partial y} \cdot \hat{r}_y + \frac{\partial I}{\partial z} \cdot \frac{\hat{r}_z}{sca} \quad (10.2)$$

Due to the a priori knowledge of the skull structure, only positive radial gradients are used in construction of the weights to find the inner surface of the skull. The 3D radial gradient, and the subsequent spherical resampling, are also chosen on the basis of the a priori knowledge of the head shape. A transversal and sagittal slice of the positive radial gradient are seen in Figure 10.3.

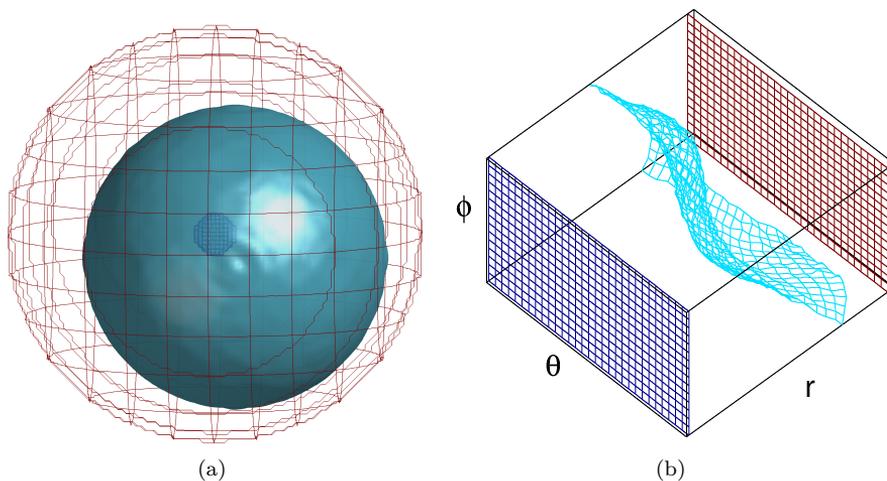


**Figure 10.3:** A transversal (a) and sagittal (b) slice of the positive 3D radial gradient.

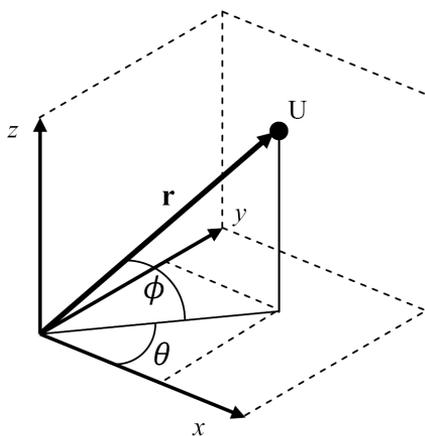
## 10.4 Spherical Resampling

When constructing flow lines for graph cutting, the nature of the sought surface must be considered in order to construct a multi-column set that is cut only once for each column. In practice this limits the method to terrain-like surfaces, but by deliberate resampling of the nodes, more complex surfaces can be transformed into simple terrain-like surfaces as well. In order to transform a volumetric closed surface, a spherical resampling is used, implying a sought surface with star-shaped behavior. As a consequence it is assumed that all edge points can be reached with a straight line from the center. This is the basis of the multi-column construction, since each column cuts the surface exactly once. A spherical resampling of a sphere is visualized in Figure 10.4. The source and sink are created as spheres as well and are shown as the dark blue and dark red grid. The resampling is based on a sampling grid with its center in the middle of the source, and the sink and source are therefore represented as a plane surface in the spherical coordinates. The sought surface is in contrast not plane due to its displacement with respect to the center of the sampling grid. Reference is furthermore made to Figure 10.1(g) where the spherical resampling of the 3D gradient image is visualized in spherical coordinates.

The spherical resampling is based on the conversion of spherical coordinates to Cartesian coordinates as seen in Equation 10.3, where  $r$  defines the radius from the center to a sampling point,  $\phi$  the rotation in the  $z$ -plane and  $\theta$  the rotation in the  $x$ - $y$ -plane, cf. Figure 10.5. The creation of the sampling grid is further explained in the following.



**Figure 10.4:** Spherical resampling of a sphere (a) and resulting terrain-like surface (b). The dark red and dark blue surfaces represent the sink and the source, respectively. Each point on the source has a corresponding point at the sink, and the line between represents a column which cuts the surface only once.



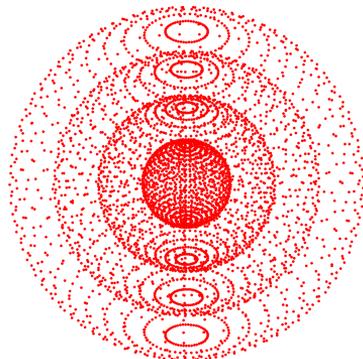
**Figure 10.5:** Visualization of the definitions for converting spherical coordinates to Cartesian coordinates and *vice versa*.

$$\begin{aligned}
 x &= r \cdot \cos(\phi) \cdot \cos(\theta) \\
 y &= r \cdot \cos(\phi) \cdot \sin(\theta) \\
 z &= r \cdot \cos(\phi)
 \end{aligned}
 \tag{10.3}$$

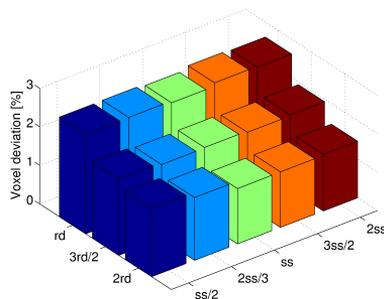
By varying  $r$ ,  $\phi$  and  $\theta$  as follows  $r = ]0 : R]$ ,  $\phi = ]-\pi/2 : \pi/2[$  and  $\theta = [-\pi : \pi[$ , the spherical coordinates will create the sampling scheme, visualized in Figure 10.6. Note that  $R$  specifies the radial spread of the sampling.

The number of spherical points is controlled by an estimated radius of the skull, meaning that the sampling density is adjusted to approximately one voxel around the equatorial band of the skull. The radius is found by estimating the volume of the intensity thresholded image in each slice, and by treating this area as a circle, the radius in each slice is estimated. The slice with the greatest radius is assumed as the equatorial band, and this radius is used to calculate the sampling density for  $\phi$  and  $\theta$ . The sampling density for  $r$  corresponds to the  $x$ - $y$  dimension of a voxel, *i.e.* it is not dependent on the slice thickness. By converting the spherical coordinates to Cartesian coordinates, the resampling of the gradient image is possible. The resampling is performed by a trilinear interpolation (3D interpolation) of the image grid, in order to estimate the values for the spherical sampling points. Note that all processing of the image is performed in physical coordinates, in order to accommodate all variable images sizes.

Even though a sampling density of one in the equatorial band intuitively is most correct, other sampling densities have been considered. Both the amount of sampling points in a sampling sphere,  $ss$ , (cf. Figure 10.6) and the radial density,  $rd$ , are varied. Again the test set, introduced in Section 10.2, is used as a phantom and the results are assembled in Figure 10.7. From the figure it is noted, that a variation of the sampling points in a sampling sphere has no crucial effect on the voxel deviation, whereas a higher sampling density in  $r$ , is seen to improve the performance. The voxel deviation is defined as the percentage-wise difference of all erroneous voxels. No test was performed when lowering  $rd$ , since preceding tests only gave results with higher voxel deviations. However, it should be noted that the processing time is drastically increased with an increased number of sampling points, so a trade-off is crucial. Even though the same effect were seen for the real datasets, it was not possible to test the performance with a doubling of the sampling spheres due to limitations in server memory capacity. It was therefore decided to proceed with a sampling density of one in the equatorial band, *i.e.*  $ss$  and  $rd$  equal one. Future studies, together with the incorporation of a more memory and time efficient implementation, could process with this information for inspiration and further analysis. It should furthermore be noted that no masking with respect to the source and sink is performed, which would be the first step in future time and memory optimization (due to the high sampling rate around the center, as seen in Figure 10.6). This is further discussed in Chapter 12 and Chapter 13.



**Figure 10.6:** The spherical scheme for resampling. Note that the sampling density is exaggerated in order to visualize the sampling spheres created by varying  $\phi$  and  $\theta$ , and their radius controlled by  $r$ . With this spherical sampling grid, the number of sampling points is constant in each sampling sphere.



**Figure 10.7:** Investigation of the effect of the sampling density.  $ss$  controls the sampling points in each sampling sphere, whereas  $rd$  controls the density of the spheres. The performance is evaluated based upon the voxel deviation given in percentage.

### 10.4.1 Considerations Concerning the Creation of the Sampling Grid

When a sampling grid, like the one specified above, is used, the sampling density will not be uniform. As mentioned earlier the sampling density does only correspond to a "uniform" voxel resolution near the equatorial band of the sphere. The sampling density is drastically increased near the poles and in the middle of the sampling grid, but decreases in the periphery. The advantage in this construction is the easy and intuitive data handling in spherical coordinates. The data matrix has direct flow lines from the center and out by varying  $r$ , and thereby an easy construction and approach of neighboring nodes.

An alternative, but more comprehensive procedure, would be to triangulate the surface of the sphere with the estimated head radius, and thereby achieve uniformly distributed sampling points on the surface. The radii, or flow lines, from the centre of the sphere and through one triangulated point, would then define the sampling points on the other sampling spheres, which still would be spaced one voxel apart. The sampling density would thereby be more uniform around the periphery of the whole sphere, but still higher in the center and decreasing with higher radii. This constellation is still necessary in order to construct straight flow lines for the purpose of graph cutting.

One could also consider a cylindrical resampling, *i.e.* a slice-wise polar resampling, as in [24]. This could speed up the algorithm due to the decreased sampling rate at the poles of the sampling spheres, but it would not be in accordance with the shape of the head. Intuitively this could lead to problems especially in the top and bottom of the skull, and the spherical resampling is therefore chosen instead.

## 10.5 Edge Weights

The data term of the vertex cost, described in Section 7.3, is based on the radial gradient. For graph cutting the minimum cut is sought, and the data term in the edge cost must therefore facilitate a low edge capacity in regions with high positive gradients, *i.e.* the border line between brain matter and bone. The edge cost,  $E_{data_l}$ , for the edge  $l$ , is therefore defined as in Equation 10.4, where  $u$  defines the node from which the edge is linked, [24]. The radial gradient  $\frac{\partial I}{\partial \mathbf{r}}$  is defined in Equation 10.2.

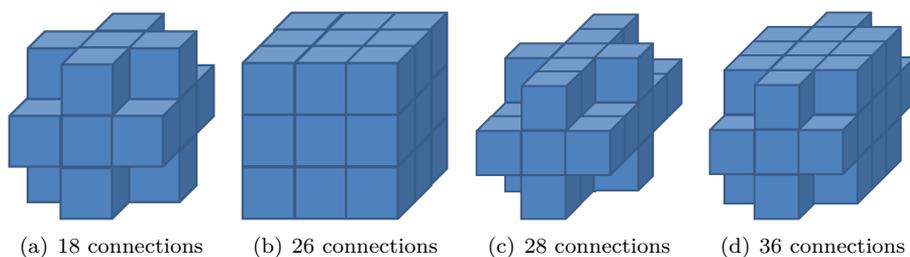
$$E_{data_l} = \max \left( \frac{\partial I}{\partial \mathbf{r}} \right) - \frac{\partial I(u)}{\partial \mathbf{r}} \quad (10.4)$$

In order to remedy the effect of bias induced by the gradient filter, the placement of the vertex cost from the data term is moved one sampling point back in the direction of the radials. Even though a radial gradient is used, the gradient filter can still be understood as a patch with  $[-1 \mid 1]$ , and the placement of the real gradient boundary will thereby be shifted a half voxel (in the radial direction).

As mentioned in Section 7.3, the opposite directed edges are set to infinity in order to ensure a unidirectional flow. Hereby all intra column weights are defined. The algorithm by Boykov [7] is applied to find the minimum cut, *i.e.* the spherical segmentation.

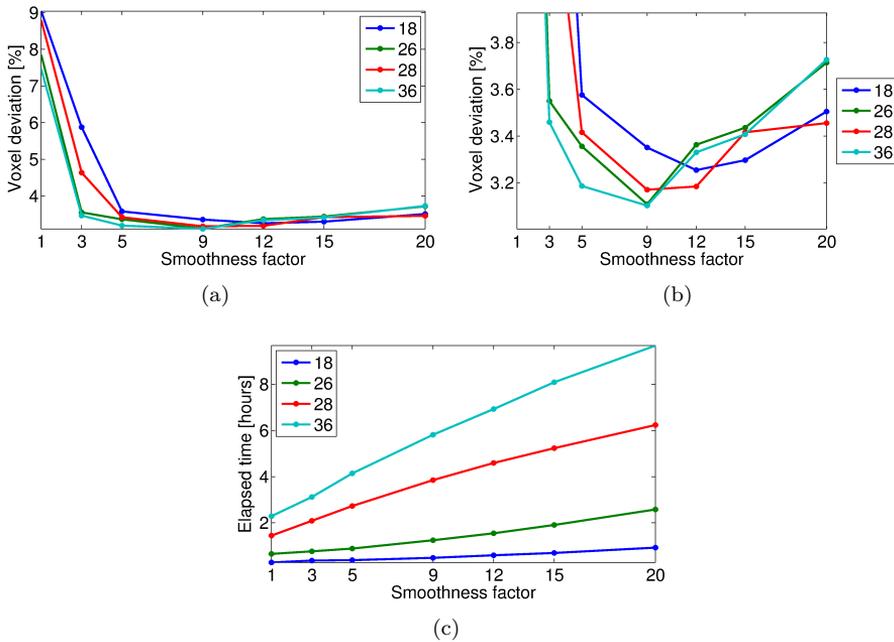
## 10.6 Node Connections and Inter Column Weights

The neighborhood of a node needs to be defined as well. The edge connection to the next sampling point in the radial direction has, as mentioned, the data term edge cost associated and infinity cost the opposite way. The other structural closest 3D points are defined as the rest of the neighborhood, and have inter column weights associated. Four kinds of neighborhoods are considered, and visualized in Figure 10.8. The node in the middle of the structures are in all cases the node of concern, and the formation refers to the matrix of the spherical sampling grid, where  $\phi$  is increasing from right to left,  $\theta$  from top to bottom and  $r$  increases from front to back. It should be noted that the structures in Figure 10.8(c) and 10.8(d) are not symmetric, but have a wider spread in the direction of the radial. This is done in order to facilitate possibility of a higher dependency of variation in the direction of the radial gradient. It should be noted that a thorough interpretation of the neighborhood connections is not facilitated in this study, and an advanced understanding and investigation of the effect and influence is necessary in future work. Different neighborhoods could have different effects for different applications, but for now the performance of the algorithm is only seen to depend slightly on the extent of the neighborhood. By testing the performance of the different neighborhoods, a reasonable neighborhood setup for segmentation of craniofacial anatomy is chosen. The results are seen in Figure 10.9 and discussed after the introduction of the inter column weights.



**Figure 10.8:** Number of connections associated with the sampling point in the center of the structures. The structures are extracted from the spherical sampling matrix, so the orientation is the same as in Figure 10.4.  $\phi$  is increasing from right to left,  $\theta$  from top to bottom and  $r$  from front to back.

The inter column weights are set to be constant in the neighborhood. This is done to facilitate an easier implementation, and due to the fact that no  $x$ - $y$  correspondence is intuitive in a spherical sampling grid. Note that even a connection to a node in the radial direction has the inter column weight associated, as long as it is not the one closest to the node of concern. The inter



**Figure 10.9:** Test of the influence of the smoothness factor (the inter column weight) and the number of connection in the neighborhood. In (a) the performance is defined by means of the mean voxel deviation in percentages compared to the manual segmentation. (b) is a zoomed version of (a). (c) summarizes the mean processing time in hours for the weight construction and graph cutting. Seven datasets contribute to the results, and their individual results can be inspected in Appendix A.

column weights are to be tuned to restrict the smoothness of the surface. The higher the weight, the fewer edges are cut, hence a smoother surface. The inter column weight is therefore also referred to as a smoothness factor.

In Figure 10.9 the influence of the smoothness factor is investigated together with the effect of the number of neighboring connections. The performance is defined by means of the voxel deviation in percentage (Figure 10.9(b)), compared to the manual segmentation. The results summarize the effect from a seven dataset study, where the tests are run on level  $L_1$  in order to correspond to the final implementation. The limited amount of data in the parameter test is due to time limitation since one parameter test has a duration of approximately three days, even on the high memory server. Note that all individual parameter tests can be found in Appendix A. The parameter test was performed for a smoothness factor of 1, 3, 5, 9, 12, 15 and 20, and the four neighborhoods defined in Figure 10.8. The voxel deviation is seen to decrease for higher smoothness

factors, but increases again for the highest factors. This increase could be due to that the smoothness is too severe and information, compared to the manual segmentation, could be lost. The effect of the neighborhood connections has only a slight effect, but generally the neighborliness with 26 and 36 connections outperforms results with 28 and 18 connections. The effect of the neighborhood is furthermore considered in Chapter 12. Figure 10.9(c) summarizes the mean processing time for the weight construction and graph cutting, and a clear tendency is seen. The higher the smoothness factor and the higher the number of neighborhood connections, the higher the processing time. A trade-off between the level of smoothness and neighborliness versus the processing time must therefore be enforced. Based on these results, it was first decided to process with a smoothness factor of 5 and a neighborliness with 18 connections for all datasets. The decision was primary based upon the time factor, but more optimal parameters could be chosen. In fact, results with a smoothness factor of 9 and a neighborliness with 26 connections has one of the lowest mean voxel deviations, and still a reasonable time trade-off. It was later in the process seen necessary to change the parameters to these values, cf. Chapter 11, where both results are investigated.

## 10.7 Conversion Back to Cartesian Coordinates

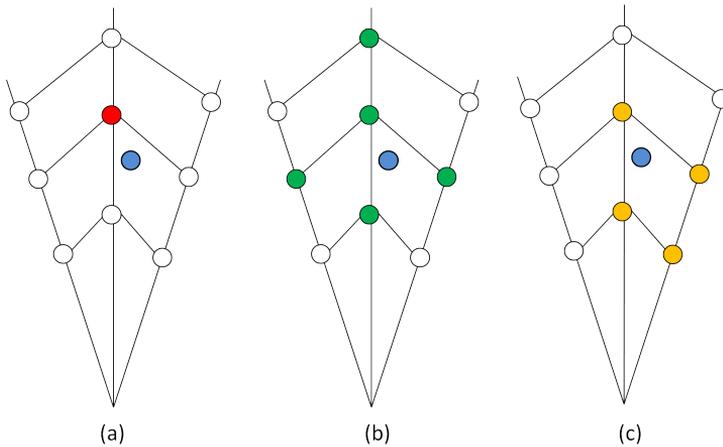
The binary mask is now obtained by graph cutting in spherical coordinates. In order to validate the segmentation the spherical coordinates must be transferred back to image coordinates. Each Cartesian grid point of the image is converted to its corresponding spherical coordinate by Equation 10.5. Equation 10.5 uses the same notation as in Figure 10.5 and Equation 10.3.

$$\begin{aligned} \theta &= \text{atan2}(y, x) \\ \phi &= \text{atan2}\left(z, \sqrt{x^2 + y^2}\right) \\ r &= \sqrt{x^2 + y^2 + z^2} \end{aligned} \tag{10.5}$$

The notation,  $\text{atan2}$ , is the four-quadrant inverse tangent (arctangent) of the real parts of the two inputs.

A majority count of the eight grid points, from the spherical samplings scheme, encapsulating the converted voxel, is then used to determine the mask value of the Cartesian grid point, Figure 10.10(c). This is done for all Cartesian grid points, by which the desired 3D mask is found, as seen in Figure 10.1(h). It should be noted, that a variety of majority counts have been considered, see

Figure 10.10, and through that different results could be obtained. In fact, a slight decrease in error rates was seen when using a "majority count" only based upon the nearest sampling point, Figure 10.10(a). Also (b) entails a decrease compared to the majority count specified in (c). Why this effect is seen, is not understood but could be due to an implementation error. It was however decided to proceed with method (c) since it intuitively is the most correct, but a further investigation would be prudent. Furthermore, it should be noted that the implemented majority count was not distance weighted, which naturally would be the most precise approach.



**Figure 10.10:** 2D visualization of the spherical sampling points on which the different majority counts are performed. Note that the 2D visualization entails no sketch of the sampling points when varying  $\phi$ . The blue circles mark the transformed image voxel of interest. (a) is a "majority count" only based on the nearest sampling point (red dot). (b) is a majority count based on seven sampling points, where the closest sampling point again is chosen together with its six neighbors. (c) is the majority count in use, where the eight points that encapsulate the converted Cartesian grid points constitute to the majority.



All results have been validated against the manual segmentation performed with the programs *Landmarker* and *Analyze*, cf. Chapter 5. Since these manual segmentations were performed by different individuals, a small study was conducted in order to evaluate the deviation in these performances, and the results are presented in Section 11.1. Subsequently, results from the two segmentation models are presented in Section 11.2, where they are evaluated on the basis of the manual segmentations. In Section 11.3 the results from a small comparison study is presented, where the performance of the two models are compared to each other. Furthermore the obtained volume estimations are compared to normal data obtained from unaffected children.

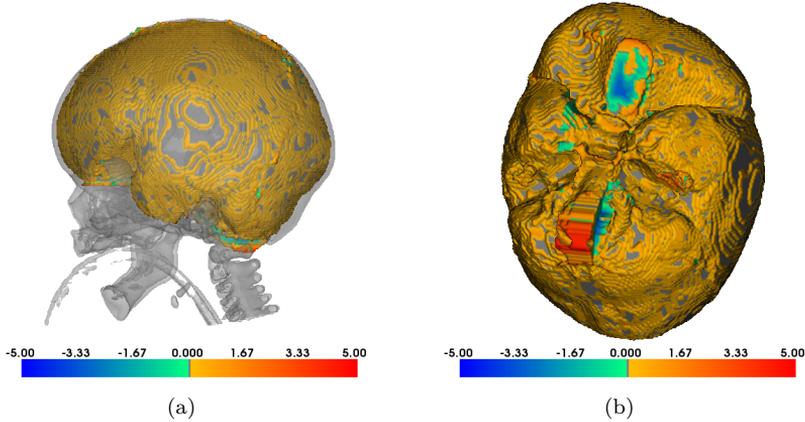
## 11.1 Manual Segmentation

In order to get an impression of the error induced by the individual performing the segmentation, three datasets were segmented twice. Each of the three dataset was segmented by a skilled user, hereafter referred to as U1, who was familiar with the program, and an untrained user, hereafter referred to as U2, for whom the program was new. The results are outlined in Table 11.1, where the volume for the two segmentations is presented for each dataset, together with the calculated voxel deviation given in percent. The maximum manual

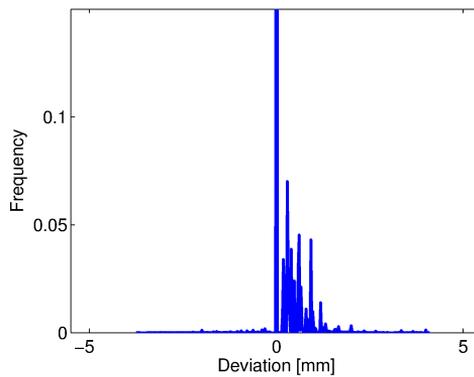
error is seen to be found at 1.74 %. An illustration of the extent and location of the differences between the two segmentations for data #1 is seen in Figure 11.1, where the closest point difference is calculated between the surface created by U1 and U2, where U1 is given as source. The extent of the voxel deviation is presented in millimeter. The blue color indicates that U2 lies on the outside of the source, whereas red indicates that U2 lies on the inside of the source segmentation. Small deviations are seen around the entire skull arch, whereas larger errors are seen in the cranial floor, around the foramen magnum and the region above the eyes. It should be noted, that the calculated volumes and the images seen in Figure 11.1 are from the original images, hence no downsampling has been performed. In Figure 11.2 a histogram of the actual voxel deviations in millimeter is presented. A slight bias of the voxel deviation is seen towards the right in the histogram, indicating that the volume segmented by U1 is slighter larger than the volume found by U2. It should also be noted that the histogram is shown in a cropped version, in order to achieve a better visualization of the variations in the histogram.

**Table 11.1:** Volume estimations from two individual segmentations and their appurtenant voxel deviation in percentage.

	Volume		Voxel deviation [%]
	U1 [ $cm^3$ ]	U2 [ $cm^3$ ]	
#1	859.6	846.7	1.74
#5	1039.8	1046.8	0.79
#8	1076.9	1075.7	0.45



**Figure 11.1:** Error maps, illustrating the voxel deviations between the two users for dataset #1, calculated as closest point difference with U1 as source. The deviations are given in millimeter, where a green/blue color indicates that U2 lies on the outside of the source, whereas a yellow/red color indicates that U2 lies on the inside of the source segmentation. Deviations above 5 mm and under -5 mm are truncated to 5 mm and -5 mm, respectively. Further, it should be noted that (a) depicts the bone surface, whereas (b) facilitate a view of the bottom of the segmentation mask (foramen magnum is found in the bottom of the image.)



**Figure 11.2:** Histogram showing the frequency of point-to-point calculations as a function of deviation between two manual segmentations performed on data #1. The histogram is presented in a cropped version to facilitate a better visualization.

## 11.2 Model Based Segmentations

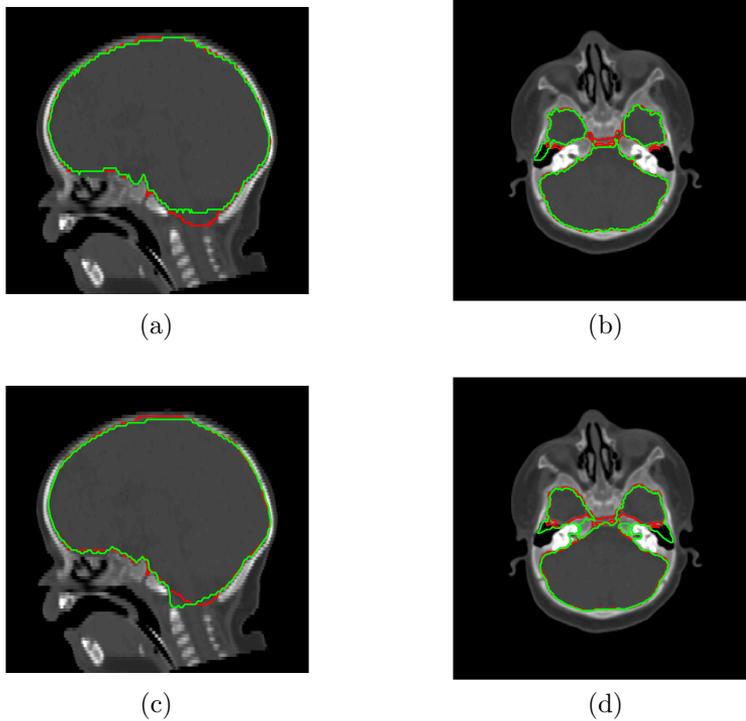
All images were segmented with both models, which give 15 results for the graph cut approach, but only 14 for the image registration approach, where one dataset was used as a template, and thereby left out as test data. Results for all datasets are collected in Table 11.2, where the volume estimations for the manual and the two model based segmentations are outlined, together with the corresponding voxel deviations. The presented image registration results are found based on the setup defined in Chapter 9, and the graph results with a setup consisting of a neighborliness of 18 and a smoothness factor of 5. The volumes for the image registrations are seen to range from lying slightly above to slightly under the manual segmented values, while the volumes found based on the graph cut approach all are seen to lie a bit above the manual values.

Furthermore, the voxel deviation between the manual segmentation and the two models, are calculated and presented in percent. The mean voxel deviation for the image registration model with the 14 datasets, was found to be 1.88 %, while it for the graph cut model, with the 15 datasets, was found to be slightly higher at 4.42 %. It is though noted that three outliers are seen for the graph cut deviations, dataset #7, #14 and #15, respectively. By further examination, the high deviations were seen caused by high gradients outside the area of interest inducing significant segmentation deviations. An example is seen in Figure 11.4, where the segmentation boundary for dataset #7 is presented. Similar segmentation deviations were seen for #14 and #15. If these three datasets are disregarded, the mean voxel deviation for the graph cut method falls to 3.74 %.

**Table 11.2:** Result overview, listing the volume estimations from the manual and the two model based segmentations. The voxel deviation between the manual and the two model based segmentations are furthermore presented.

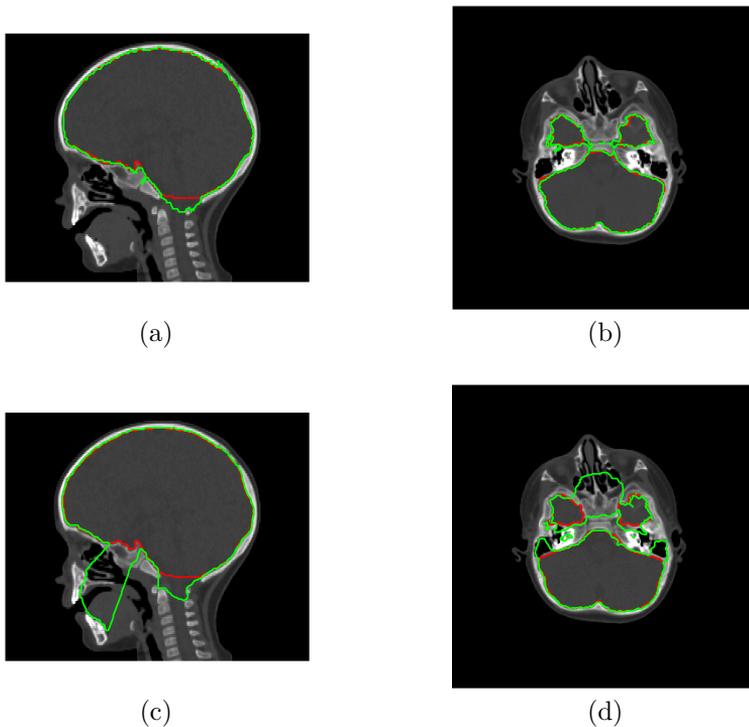
#	Image Registration			Graph Cut	
	Manual Volume [ $cm^3$ ]	Volume [ $cm^3$ ]	Model Dev. [%]	Volume [ $cm^3$ ]	Model Dev. [%]
1	859.6	852.3	2.32	868.0	2.71
2	824.8	826.5	2.80	840.0	3.17
3	921.2	912.5	2.39	947.2	3.81
4	938.6	953.5	2.34	977.3	4.45
5	1039.8	1040.9	1.69	1079.3	4.32
6	1066.4	1068.4	1.86	1094.0	3.54
7	1047.5	1065.7	2.40	1142.4	9.48
8	1076.9	1081.6	1.75	1100.2	2.97
9	888.7	899.6	2.01	920.3	4.07
10	1081.4	1083.2	1.51	1119.2	3.97
11	695.3	706.4	2.40	719.4	4.15
12	902.5	908.4	1.51	923.6	3.06
13	1101.3	1105.4	1.29	1149.4	4.68
14	1240.9	-	-	1302.3	5.23
15	982.1	970.8	1.97	1042.6	6.64

The voxel deviations are seen insufficient to provide a proper evaluation of the model based segmentations. The estimated segmentation boundary is therefore viewed together with the manual segmented boundary, both superimposed on the corresponding CT image, in order to locate the regions of errors. In Figure 11.3 the segmentation boundaries for the two models from dataset #1, are viewed together with their corresponding manual segmentation, and shown in both a sagittal and transversal view. In all illustrations the manual segmentation is presented in red, and the model based in green. In (a) and (b) the results obtained from the image registration are seen, whereas (c) and (d) show the graph cut results. Both models are seen to provide reasonable segmentations, only showing small deviation from the manual segmentation, in the area of the foramen magnum and around the sphenoid sinuses, which are seen in the transversal slices as the black areas in both side of the skull.



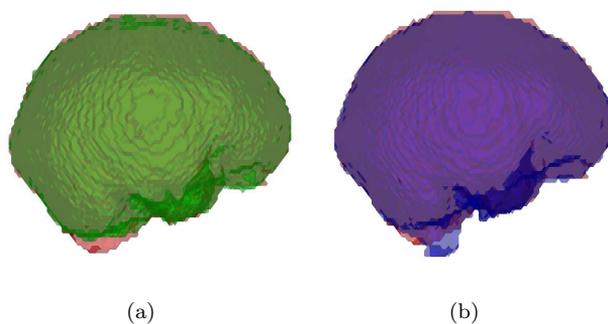
**Figure 11.3:** Segmentation boundary for the manual (red line) and the model based segmentation (green line) for data #1, superimposed on the corresponding CT image. (a) and (b) present the results from the image registration, and (c) and (d) the graph cut.

Another example is seen in Figure 11.4, which provides illustrations of the segmentation results from dataset #7. The graph cut segmentation is seen to result in large errors in the cranial floor, which correlates with the high voxel deviation seen in Table 11.2. These large deviations are seen to be forced by large gradients at the teeth and spinal bones. However, ignoring these two deviation areas, the remaining segmentation has a very high correlation with the manual segmentation. The graph cut based segmentation is actually seen to provide more precise, ignoring the two prior mentioned areas, than the segmentation obtained from the image registration, which is seen to be fluctuating, especially in the cranial arch.

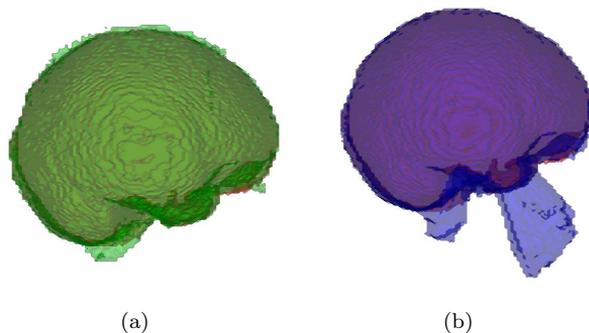


**Figure 11.4:** Segmentation boundary for the manual (red line) and the model based segmentation (green line) for data #7, superimposed on the corresponding CT image. (a) and (b) present the results from the image registration, and (c) and (d) the graph cut.

In order to evaluate the results more profound, a 3D rendering of the segmentations is performed. The results are seen in Figure 11.5 and Figure 11.6 for dataset #1 and #7, respectively. The manual segmentation shown as the red surface, the image registration green, and the graph cut based segmentation in blue. The extent of the fluctuations referred to for dataset #7 for the image registration is in this illustration clearly seen.



**Figure 11.5:** Surface rendering illustrating the difference between the manual segmented mask (underlying red), and the image registration segmented mask (green) in (a) and the graph cut segmented mask (blue) in (b) for dataset #1.



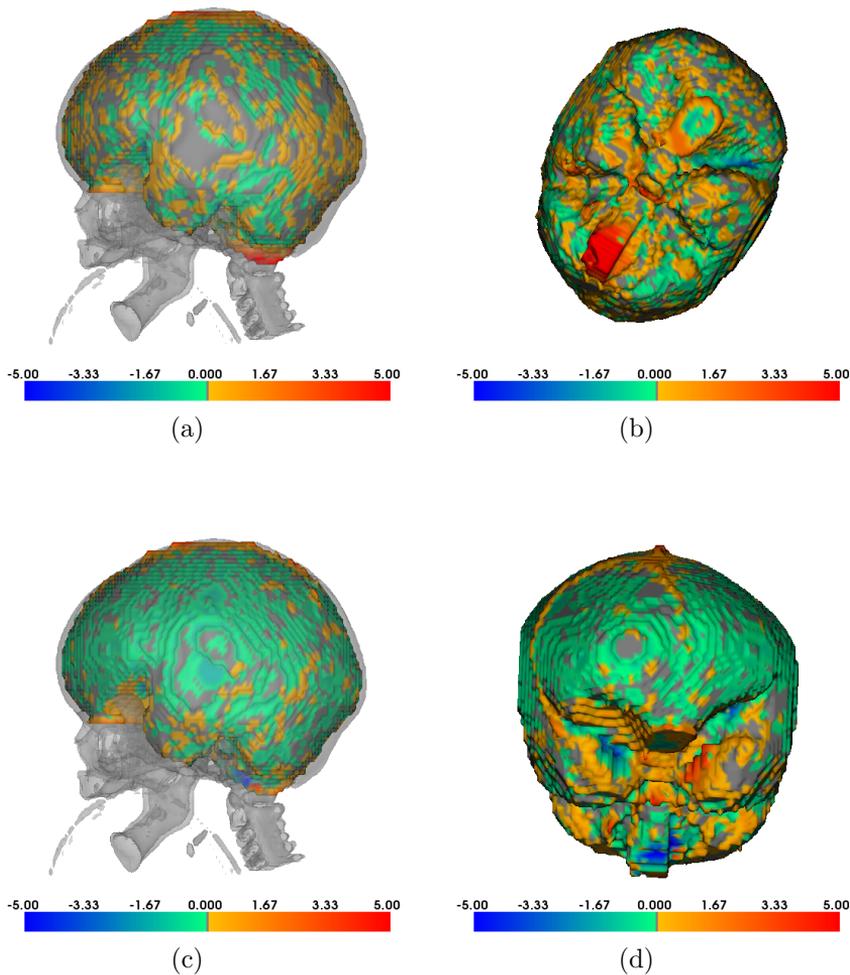
**Figure 11.6:** Surface rendering illustrating the difference between the manual segmented mask (underlying red), and the image registration segmented mask (green) in (a) and the graph cut segmented mask (blue) in (b) for dataset #7

Another evaluation method utilized in this thesis, was to generate movies of all datasets, illustrating the segmentation in a semi-3D space. The movies were generated based on a series of slice based illustrations, as the ones seen in Figure 11.3 and Figure 11.4. Two movies were generated for each dataset, showing the segmentation in a transversal and sagittal view. These are available for the all datasets on the enclosed CD.

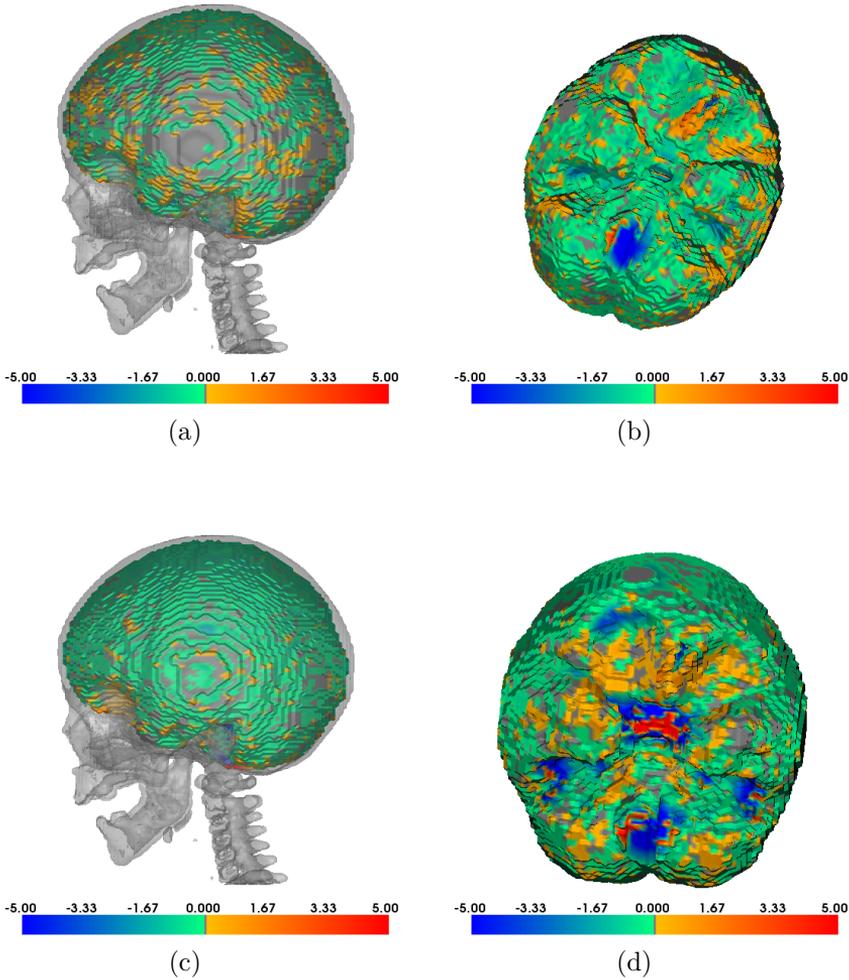
In order to evaluate the extent of the voxel deviation, the closest point differences found between the surface of the manual and model based segmentations are superimposed on a 3D rendering of the manual segmentation, similar to the illustration shown in Figure 11.1. The extent of the deviations are shown in millimeter, where a green/blue color indicate a model based segmentation lying on the outside, whereas a yellow/red color indicate a model segmentation located on the inside of the manual segmentation. Furthermore, it should be noted that deviations above 5 mm and under -5 mm are truncated to 5 mm and -5 mm, respectively.

The illustration of the location and the extent of the voxel deviation for the two datasets examples are seen in Figure 11.7 and Figure 11.8, where (a) and (b) is results from the image registration, and (c) and (d) from the graph cut. For dataset #1, in Figure 11.7, the image based segmentation is seen equally distributed between lying on the inside and the outside of the manual segmentation. Significant deviations are only seen in the top and bottom of the cranium, where the image registration based segmentation in some places is seen to lie more than 2-3 mm from the manual based segmentation. In the area of the foramen magnum the deviation is as high as 5 mm or more. The graph cut based segmentation is primarily seen to lie on the outside of the manual segmentation, but in general only with a small deviation. Similar to the image registration based segmentation, larger deviations are seen in the top and bottom, where the graph cut segmentation is seen to lie on the inside of the manually segmented boundary, except in the region of the foramen magnum.

For dataset #7, illustrated in Figure 11.8, both models shown good performance, by providing very nice segmentations. The image registration based segmentation in the skull base seen slightly biased towards a segmentation lying on the outside of the manual, which is opposite compared to dataset #1. The graph cut based segmentation is seen to show significant deviations at 5 mm or more in the bottom of the cranium, corresponding to the large boundary movements caused by the teeth and spine gradients, seen in Figure 11.4(c). Disregarding this deviation, the remaining segmentation is located slightly on the outside of the manual segmentation, but equally distributed.

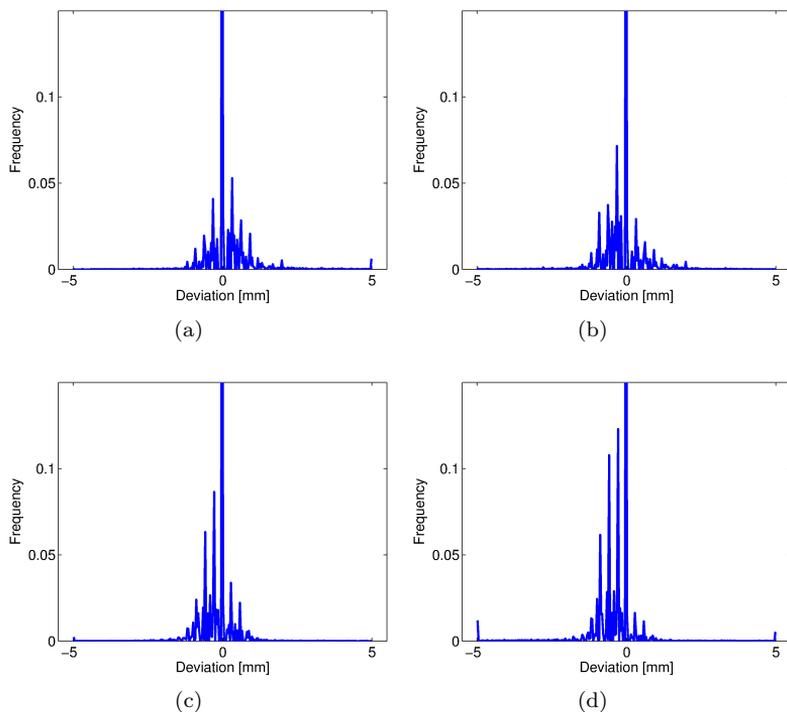


**Figure 11.7:** Illustration of the extent and location of the voxel deviation for dataset #1, given as the closest point differences found between the manual and model based segmentation, superimposed on a 3D rendering of the manual segmentation. (a) and (b) present the results from the image registration, and (c) and (d) the graph cut. The extent of the deviations are shown in millimeter, where a green/blue color indicates a model based segmentation lying on the outside, whereas a yellow/red color indicates a model segmentation located on the inside of the manual segmentation. Furthermore, it should be noted that deviation above -5 mm and below -5 mm are truncated to 5 mm and -5 mm, respectively.



**Figure 11.8:** Illustration of the extent and location of the voxel deviation for dataset #7, given as the closest point differences found between the manual and model based segmentation, superimposed on a 3D rendering of the manual segmentation. (a) and (b) present the results from the image registration, and (c) and (d) the graph cut. The extent of the deviations are shown in millimeter, where a green/blue color indicates a model based segmentation lying on the outside, whereas a yellow/red color indicates a model segmentation located on the inside of the manual segmentation. Furthermore, it should be noted that deviation above 5 mm and below -5 mm are truncated to 5 mm and -5 mm, respectively.

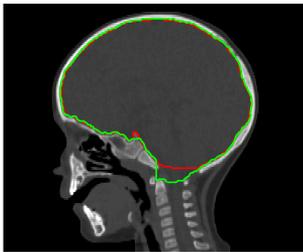
Moreover, the frequency of point to point calculation for the two examples are seen in the histograms plotted in Figure 11.9. (a) and (b) present the results for dataset #1 for image registration and graph, respectively, and (c) and (d) results from dataset #7. The majority of the deviations are seen to lie within the interval of -2 to 2 mm. Furthermore, it is evident, that the closets point differences found for the graph cut has a slight tendency towards the negative values, corresponding to the previous results, where the main part of the segmentation was seen to be found lying on the outside of the manual segmentation. The closets point differences for the image registration is seen to be more equally distributed around zero, which especially the histogram for dataset #1 indicates.



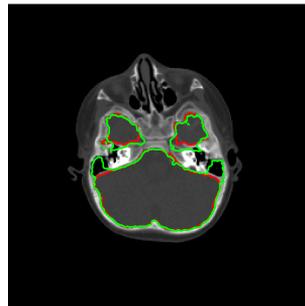
**Figure 11.9:** Histograms showing the frequency amount of point-to-point calculations as a function of deviation between the manual and the model based segmentations. (a) and (b) show the frequency distribution for #1 for image registration and graph cut, respectively, while (c) and (d) similarly show the distribution for #7. All histograms are shown in a cropped version to facilitate a better visual interpretation.

Considering the results in Figure 11.4(c) and (d), where the large gradients from teeth and spine bone induce large segmentation errors, the graph cut setup is reconsidered. The parameters for the graph cut, consisting of a neighborliness of 18 and a smoothness factor of 5, was chosen on the basis of a time trade-off. These parameters are for dataset #7 proven insufficient, wherefore a new setup is evaluated. Based on the 10.9, the constellation of a neighborliness of 26 and a smoothness factor of 9 provided the optimal setup, *i.e.* the minimal voxel deviation. A graph cut model with this new setup is applied on dataset #7, and the results are presented in 11.10 - Figure 11.12.

The large deviation in the jaw region, which was seen previously, is avoided, while the deviation in the region of the foramen magnum is present. This parameter change is seen to force a smoother segmentation surface, which results in a decrease in voxel deviation for dataset # 7 from 9.48 % to 4.07 %, but a significant increase in processing time is observed. As a consequence of this result, an additional segmentation of all datasets is performed. The results of volume estimations and voxel deviations are presented in Table 11.3, where the mean voxel deviation is found of 3.46 %. Unfortunately, an increase in mean processing time from 25 minutes and 7 seconds to 1 hour and 17 minutes is seen. A summary of the time and voxel deviation for the two models are provided in Table 11.4, where the results for both graph cut setups are provided. Noted that IR refers to image registration, and CG to graph cut.



(a)

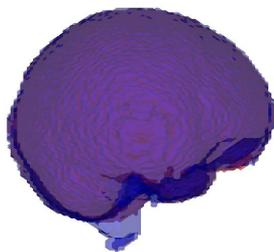


(b)

**Figure 11.10:** Segmentation boundary for the manual (red line) and the graph cut based segmentation (green line) for data #7, found with a neighborliness of 26 and a smoothness factor of 9.

**Table 11.3:** Result overview, listing the volume estimations from the graph cut segmentation with a parameter setup consisting of a 26 neighborliness and a smoothness factor of 9.

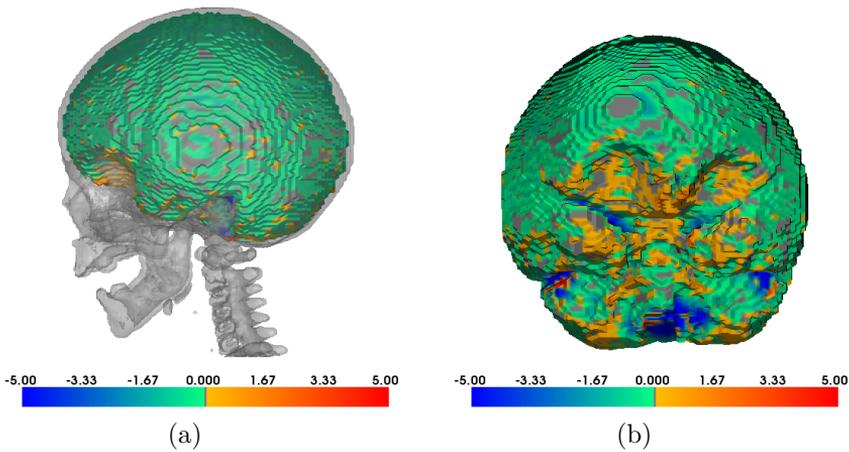
#	Manual Volume [ $cm^3$ ]	Model Volume [ $cm^3$ ]	Dev. [%]
1	859.6	866.4	2.61
2	824.8	834.4	3.43
3	921.2	942.3	3.32
4	938.6	971.4	3.82
5	1039.8	1072.0	3.70
6	1066.4	1084.1	2.79
7	1047.5	1085.5	4.07
8	1076.9	1091.6	2.81
9	888.7	914.9	3.64
10	1081.4	1135.5	3.52
11	695.3	711.5	3.18
12	902.5	918.4	2.52
13	1101.3	1146.8	4.44
14	1240.9	1297.1	4.82
15	982.1	1008.6	3.28



**Figure 11.11:** Surface rendering illustrating the difference between the manually segmented mask (red) and the graph cut segmented mask (blue) for #7 with the new setup.

**Table 11.4:** Summary of the performance of the two models, image registration (IR) and graph cut (GC), expressed in mean processing time and voxel deviation, respectively. Results for both graph cut setups are presented, where the parameters are given by smoothness factor/neighborliness.

	IR	GC 5/18	GC 9/26
Voxel Deviation [%]	1.88	4.22	3.46
Mean time	9 min	25 min	1 h 17 min

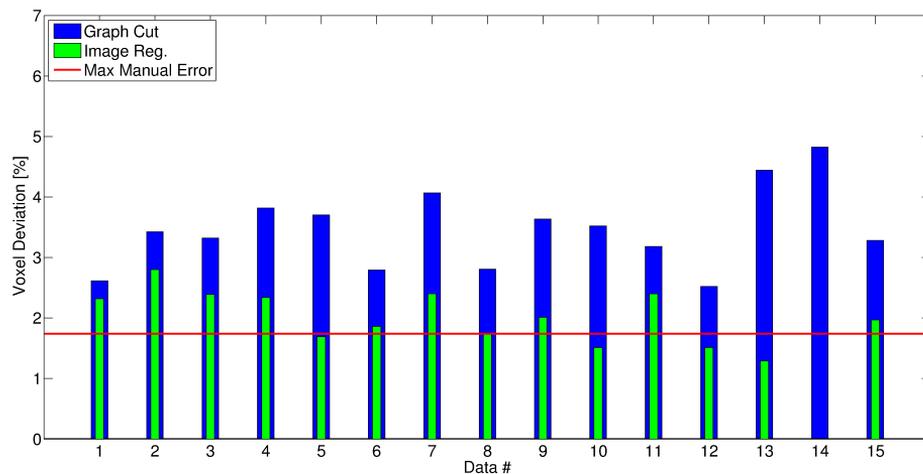


**Figure 11.12:** Illustration of the extent and location of the voxel deviation for dataset #7 with the new graph cut setup, given as the closest point differences found between the manual and model based segmentation, superimposed on a 3D rendering of the manual segmentation. The extent of the deviations are shown in millimeter, where a green/blue color indicates a model based segmentation lying on the outside, whereas a yellow/red color indicates a model segmentation located on the inside of the manual segmentation. Furthermore, it should be noted that deviation above 5 mm and below -5 mm are truncated to 5 mm and -5 mm, respectively.

### 11.3 Model Comparison

In the following section the two models are examined in a small comparison study, both according to each other, but also in correlation to normal volume estimations. All results shown for the graph cut model is based on the setup with a neighborliness of 26 and a smoothness factor of 9, *i.e.* all voxel deviations are taken from Table 11.3.

A comparison of the voxel deviations obtained from the two model segmentations, is shown in Figure 11.13, where the image registration result are shown in green and the graph cut results in blue. Additionally, the maximum manual segmentation error found at 1.74 % based on the small study in Section 11.1, is presented as the red line. The maximum manual segmentation error is arbitrary set, due the sparsity of data in the preliminary study in Section 11.1. The image registration based segmentations are seen to provide smaller voxel deviations compared to the graph cut based segmentations, and in some situations the deviations from the image registration are observed lower than the maximum segmentation error. However, the graph cut deviations are seen to lie within a reasonable range from the image registration, only dataset #13 is seen to deviate significantly. Dataset #14 is likewise seen to result in a high voxel deviation, but for this dataset comparison is not possible, due to its figuration as a template in the image registration process.



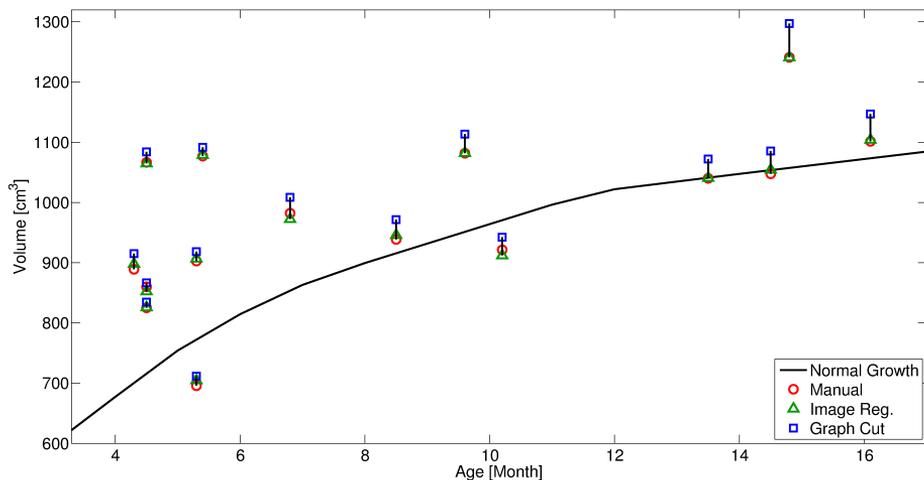
**Figure 11.13:** Voxel deviations in percent for all datasets processed with image registration and graph cut, represented with green and blue bars, respectively. The red line defines the maximum manual segmentation error found at 1.74 % based on the small study in Section 11.1.

In Figure 11.14 a scatter plot showing the volume results for the manual and the two model based segmentations is shown together with a curve, illustrating the normal cranial growth pattern for children up to 17 months. The normal curve is based on data from [41], and produced as in the paper, by use of the Lowess smoothing technique<sup>1</sup>, with a 50 % smoothing window and three iterations. The results from the two model based segmentations; image registration and graph cut, are illustrated with green triangles and blue squares, respectively, are seen to lie close to the manual segmentations, illustrated by the red circles. It should be noted, that for the dataset with a value above 1200 cm<sup>3</sup>, no value exist for the image registration model, as this dataset figures as a template. The image registration value is therefore set equal to the manually segmented volume. In general the graph cut results are seen to have a tendency of lying slightly higher than both the image registration and manual results. This is further visualized in Figure 11.15, where the volumes from the two segmentation models are plotted against each other, and the referred tendency easily is seen, as all marks are placed above the mid-line.

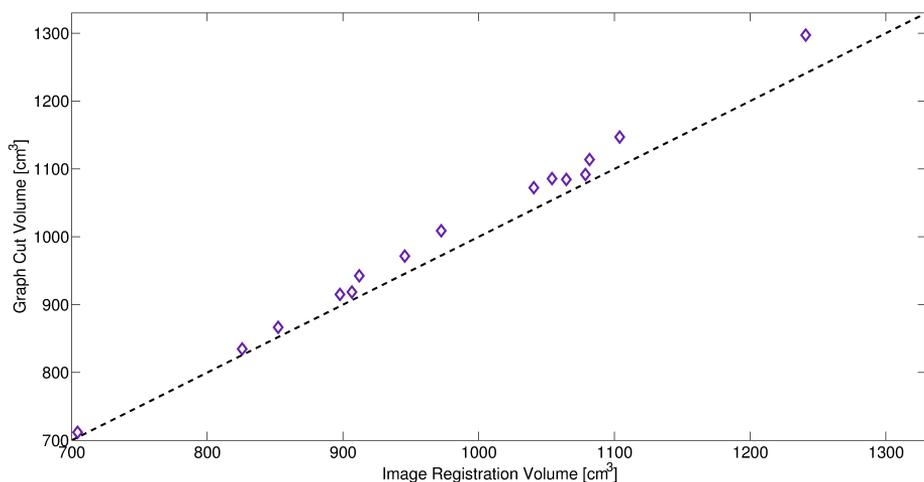
The volume estimations are until the age of 6-7 months seen to be a bit increased compared to the normal curve, where after they tend to move closer around the curve. However, based on the number of data examined in this thesis, it is statistically difficult to say, whether this is a general tendency. Even though no unambiguous statement can be based on the results gained here, both models show possibilities for further population studies.

---

<sup>1</sup>Uses piecewise 1D polynomial, cf. MATLAB



**Figure 11.14:** Scatter plot and curve showing the intracranial volume. The black curve is based on 18 data points from [41] and is produced by use of the Lowess smoothing technique, with a 50 % smoothing window and three iterations. The volume estimations from the manual, image registration and graph cut segmentations are represented with a red circle, green triangle and a blue square, respectively. The vertical black lines connect the three results associated with the same dataset, and are only present to ease the interpretation. For the dataset with a value above 1200  $\text{cm}^3$  it should be noted that no value exist for the image registration model, as this dataset figures as a template. The image registration value is set equal to the manually segmented volume.



**Figure 11.15:** Visualization of the correspondence in estimated volumes between the two model based segmentations. The dashed line illustrates the mid-line *i.e.* equal estimation of the volume.

# Discussion

---

For the purpose of clarity, the discussion is divided into four sections, each describing one of the main topics of concern. Section [12.1](#) discusses the veracity and applicability of the manual segmentation as frame of reference. Section [12.2](#) treats the aspects of the image registration model, while Section [12.3](#) deals with cornerstones of the graph cut model. Lastly, the possibility of the two models serving as a foundation for population studies is assessed in Section [12.4](#).

## 12.1 Manual Segmentation

The manual segmentations, which throughout this thesis have been used as a comparison foundation, and for validating the success of the two model based segmentations, have to some extent been found inadequate. Different factors are in evidence for this assessment. The user defined intensity threshold was based on a simple visual interpretation, as seen in [Figure 5.1](#), where it is hard to establish the exact bone demarcation. An acceptable value for the intensity threshold can therefore easily be chosen within a 10 HU interval. Choosing a too high value will add the risk of removing useful bone information, which will be seen in the form of a thinner bone demarcation, starting further from the brain center than the actual truth. Choosing a too low value will induce the

risk of creating a segmentation lying on the inside of the actual bone demarcation. Looking further into the segmentation process, additional sources of errors were introduced due to the manual editing. As described in Chapter 5 manual editing was necessary in order to obtain proper bounding, both due to cranial breaches caused by the craniosynostotic condition, but also due to the natural foramina in the cranial base. First of all, the editing was performed as freehand drawing by means off the computer mouse, which alone was a difficult working tool. Subsequently, the estimation of the correct anatomical outlining, especially in the cranial base, can be rather difficult to establish. Among the 15 segmentations performed for this thesis, great variation was seen, especially at the outlining of the foramen magnum. Another aspect, which should be emphasized, is the aspect of human fatigue during the two hour processing time. Several manual segmentations were seen to have small ridges in the midline of the top of the cranium. These are most likely caused by small breaches in the bone line, which were of such a size, that the user has found them insignificant, and not performed any adjustments. Furthermore, the editing above the eye cavity, were, in some cases, seen performed rather simple, with just a straight line, easily seen on Figure 11.3. Summarized, these factors can contribute to a significant error.

In order to obtain a higher similarity between the mask segmentations, general guidelines must be established, especially concerning the anatomy in the cranial base, but also in the choice of intensity threshold. First, one could question the variation in the chosen intensity thresholds in between datasets. Since all images were CT images presented in HU values, it would be assumed that a global intensity threshold could be established and used on all images. However, this was not seen possible. It has also been considered whether the intensity threshold was based on age, hence the bone density, which is seen to increase through age. Such a correlation was not found significant among the datasets processed in this thesis. Furthermore, a study, [10], has shown, that a global threshold for a single dataset might be inadequate.

In Section 11.1, Table 11.1, a difference in voxel deviation up to 1.74 % was seen between two segmentations of the same dataset performed by two individuals. This result was based on a very sparse study, which could be of interest to examine further, including a study of the individual variations. Consider Figure 11.1, which shows the closest point differences for the dataset with the highest voxel deviation, dataset #1, it was seen that the deviation mainly was due to a difference in the intensity threshold chosen for the two segmentations. Furthermore, voxel deviations were especially seen in the critical areas of the cranial base, around the foramen magnum and the region above the eyes, caused by the different handling of the manual editing. This was manifested in the histograms in Figure 11.2, where the deviations were biased to the right in the histogram. Taking all these considerations into account, it could be argued whether these

manual segmentations can be used as proper validation foundation, or if they in some cases serve more as an error contribution.

## 12.2 Image Registration

The image registration based segmentations were conducted by means of a software package passed along to this project. The general registration setup was therefore already established, and modifications were performed in order to adjust the program to this thesis. In general, the image registration process was seen to perform very satisfactory and provided relatively low voxel deviations for almost all data. The variation of voxel deviations around the cranial arch, were unbiased towards neither side of the manual segmentation boundaries, which were further manifested by the uniform distributed histograms. The main errors were primarily seen in the cranial base, and some minor errors were located in the cranial top, as seen in Figure 11.4(a), (b) and 11.6(a). Considering the sphenoid sinuses, which were seen to be critical areas for the graph cut model, the image registration model was seen to have a higher success rate around these, although not infallible, as seen in Figure 11.3(b).

One of the main difficulties seen for the image registration model was caused by the lack of anatomical consistency between the manual mask segmentations, which induced complications in the mask transformations. The actual transformation performed on the intensity based CT images was performed with a high success, but the obtained transformation parameters were not adjustable towards the variation of anatomical interpretation induced in the mask construction. This was especially pronounced in the region of the foramen magnum, and can be seen in *e.g.* Figure B.3 in Appendix B. Another error of concern is the actual registration process, where possible errors can be induced, due to inadequate transformations. This was expressed as a slight tendency towards a fluctuating segmentation surface, especially when larger differences were present between the template and reference image. This could *e.g.* occur when gaps exist between bone plates or in the cranial base, where high anatomical fluctuations in general were seen.

Another aspect is the choice of template. Normally, when working with image registration, the template image is often constructed as an atlas, *i.e.* a mean of many images, instead of just a single image, as used in this thesis. It was noted that the choice of template, especially its resolution, had a slight influence on the results. The use of an atlas could therefore eliminate bias against the template. However, depending on how the atlas is constructed, a slight bias against the chosen images will still be present. In this case, where the processed

images were obtained from craniosynostotic children, the head shapes can differ significantly from child to child, and an atlas construction will therefore not necessarily provide the best possible template. In the worst case scenario, an obtained atlas would contain a small amount of all the different deformities and thereby result in an image far from a useful template. However, it must be noted, that since all data processed in this thesis, were gathered from children with UCS, although both left and right sided, the deformities induced will not be as varied, as if data was obtained from children belonging to different a craniosynostotic type. Another approach could have been a construction of an atlas based on images obtained from unaffected children. In this way the introduction of different deformities to the template would be avoided. The template will, however, has to align to the different UCS cases, but with the choice of transformation, this is thought feasible for most cases.

### 12.3 Graph cut

In general the graph cut procedure produced excellent segmentations, especially around the cranial arch. It was, however, noted that the segmentation boundary often was found slightly on the outside of the manual (reflected as green color at the closest point difference maps). This could be caused by the various intensity thresholds used for the manual segmentations, and in particular if a threshold was chosen too low. In many of the manual segmentation processes, the intensity thresholded boundary was seen to lie on the inside of the visually highest radial image gradient, as seen on Figure 5.3. Since the graph cut is gradient based, the model based segmentation could therefore be expected to lie slightly on the outside of the manual segmentation, which also manifests itself in the more right-side weighted histograms. The visualization of small errors in the closest point difference map is therefore not always an indication of an erroneous model based segmentation, but could be due to a defective manual segmentation. If the intensity threshold is the source of error, the image registration model will not be affected to the same extent, as the registration is intensity based as well.

The slight difference around the cranial arch could possibly be regularized further by adjusting the setup of the graph cut model. First of all, the segmentation placement is dependent on the position of the sampling spheres, a doubling of the sampling density of the spheres, as referred to in Section 10.4, could entail a more precise cut, *i.e.* closer to the level of the highest gradient. Another aspect is to consider on which edge side of a node, the corresponding weight is added. In this thesis, a result improvement was seen when placing the edge cost on the edge flowing towards the node. Moreover, the bias of the gradient could also contribute to a minor displacement.

One of the main reasons for choosing the graph cut algorithm for segmenting the cranial volume in children with craniosynostosis was the possibility to overcome breaches between cranial bones, without any prior arrangement. Gaps between cranial bones were easily overcome due to the spherical smoothness restriction incorporated in the graph cut model. Not many of the processed datasets revealed major gaps, but a high breach in the cranial arch was seen for *e.g.* dataset #6, cf. Figure 3.3. Even then, the graph cut algorithm followed the curvature of the cranium, in fact even better than the manual segmentation, see Figure B.1 in Appendix B, where the manual segmentation was seen to exceed the natural non-existing bone demarcation. Minor fluctuations can, however, be present in all surface areas, since the smoothness of the surface is dependent on the sampling density and location in the given area; hence the success of the majority count in the conversion back to Cartesian coordinates. These minor fluctuations were, however, not presented in the resulting visualizations.

In the cranial base, the graph cut model has its limitations. The many different curvatures, which can create obstructing folds, contradict the assumption regarding a star-shaped object, cf. Figure 3.3(b). The algorithm was in these areas insufficient, and will result in erroneous segmentations. Another difficulty arose around the sphenoid sinuses. Since only a thin bone demarcation exists between the sinus and the brain, the gradient of the thin demarcation was easily exceeded by the high gradient arose from the transition from the air filled sinus to the bone. One could consider gradient modifications for improved performance, as described in Chapter 13. Another gradient based error that could contribute to critical areas in the cranial base and in the region above the eyes, is the direction of the gradient. Since the radial gradient uses the gradient component in the direction of the radial, only transitions perpendicular to the radial were highly emphasized. This is again conflicting with the high curvature of the cranial base and around the eyes, where very low gradients were seen, see Figure 10.3.

Considering the cranial base, the area around the foramen magnum was an area of great concern, *i.a.* due to gradient influence of spinal bones. The extent differed in between patients, since slight variations of head posture were present. This was manifested as different boundaries, restricting how much of the spinal cord that was included in the segmentations. This variation was a further source to the higher error rates for the graph cut model than for the image registration; the image registration model was biased by the manual segmentation, and it thereby holds a higher correlation towards the manual decision of the boundary for foramen magnum. Unfortunately, the variation around the foramen magnum is also high within the manual segmentations, and a general agreement of its anatomical location should be established. In any case, it is necessary to implement a more fixed boundary for the graph cut model in order to oblige the variations that can arise both in between patients

(for populations study), and within the same patient (for surgery evaluation).

Outliers, both seen as high error rates in Table 11.2 and Table 11.3, and as large non-anatomical deviations on the scan slices, are caused by higher gradient areas outside the skull. Cf. Figure 11.4(c), Figure B.36(c) and Figure B.33(c) in Appendix B, where both the teeth, intubation pipe and bones of the spinal column were seen to cause the graph cut model to deviate from its smooth surface. As described in Section 7.3 a smoother surface can be restricted by a higher inter column weight, *i.e.* a higher smoothness factor. These, and the neighborliness of a node, were investigated in Figure 10.9. When considering the results from dataset #7, a smoothness factor at 5 and a neighborliness of 18 were insufficient to restrict a smooth surface. The more optimal parameter set, chosen based on Figure 10.9 (a smoothness factor of 9 and a neighborliness of 26), was on the contrary seen to generate the desired smooth surface. This was, however, at the expense of a significant increase in processing time, as noted in Table 11.4. As deduced from the figures in Appendix A, different datasets have different optimal parameters. It should be noted that the optimal setup could be found with a set of parameters not tested in this study, and a set of parameters could therefore exist, for which all outliers would have a smoother surface. In fact, for dataset #14, the parameter tests is seen in A.2 in Appendix A, from which an optimum is a combination with a smoothness factor of 15 and a neighborliness of 36. The higher smoothness factor was necessary in order to oppose the high gradient of the intubation tube. The very high gradient was present due to a transition from air (-1000 HU) to the tube edge (1000 HU) relative perpendicular to the radials. Since all weights were based on Equation 10.4, this maximum positive gradient will furthermore cause the other gradients around the skull to have less influence.

In further consideration of the parameter tests one could contemplate the effect of the neighborhood. From the mean of seven dataset, Figure A.1 in Appendix A, a neighborliness of 26 and 36 connections were seen to give very alike results when varying the smoothness factor. This could be due to the fact, that these connection types have links to nodes in the "corner" around the node, cf. Figure 10.8. An addition of these nodes might therefore have more effect than an extension in the direction of the radial (as with 28 and 36 connections). Another noteworthy effect is the higher the neighborliness, the higher smoothness factor was necessary to obtain the minimum result. Again it should be emphasized that a more thorough understanding and investigation of the neighborliness of a node must be established, in order to understand all aspects of its influence.

## 12.4 Model Comparison

In Section 11.3 the results of a small comparison study of the models were presented. Figure 11.13 summarizes the performance of the two algorithms, evaluated based on the voxel deviation. As discussed above, the general tendency was that the graph cut model provides slightly higher voxel deviations compared to the image registration. This is among others, due to the fact, that the graph cut model is not biased towards the manual segmentations. In some cases, the image registration was seen to outperform the manual segmentation, by providing voxel deviations lower than the maximum manual segmentation error. As summarized in Table 11.4 the processing time was another aspect of evaluating the usability of the models. It was noted that the graph cut model in general has a high processing time, which furthermore increased when optimizing the parameters. Time optimization is, therefore an important future aspect for the graph cut model, whereas the image registration might have reached its optimal efficiency. It should be noted, that the graph cut model must be processed on a high memory server, in order to apply the algorithm to an image with a resolution of  $256 \times 256$  times the number of slices. For now, with the available possibilities, it was not possible to apply the model to a higher resolution image.

Up until now, the performance of the models was evaluated based on visual interpretation and voxel deviations. However, one of the main objectives was estimation of the intracranial volume, including an evaluation of the prospective of the models potential for population studies. The estimated volumes, for both models, were therefore compared against a normal curve obtained from unaffected children, depicted in Figure 11.14. Both models exhibited fine prospects for volume estimation; the graph cut model was though seen to estimate slightly higher volumes than those found based on the manual segmentations. Comparing the UCS volumes to the normal growth curve, a slight tendency for increased volume was seen in the early ages. Whether an increase or decrease in volumes should be seen, is an aspect where no common agreement, among scientist in the craniofacial field, yet is established. Several studies concerning children with UCS show disparate results and some even shows that the volumes do not differ from normal volumes, [8, 40]. However, the results presented in this thesis might have a slight tendency to shown increased volume until an age of 6-7 months, which is in contradiction with the results presented in [40]. It should be emphasized that the study in this thesis only includes 15 subjects, which is insufficient in order to serve as a proper statistical foundation. However, the prospect for further population studies and statistical analysis is generated, both regarding comparison to normal data, but also to children with other types of craniosynostosis.

The higher volumes for the graph cut model is further illustrated in Figure [11.15](#). As described previously, the increased volumes are caused by a slight tendency of lying on the outside of the segmented surface and the difficulties in the cranial base, especially the region of the foramen magnum and the sphenoid sinuses.

All things considered, the models capability has been established, and even though that the graph cut model has slightly higher voxel deviations, its potential for optimization and applicability must be ascertained.

## Future Work and Perspectives

---

As stated in the discussion, different approaches can be taken in order to improve the process of volume estimation in children with craniosynostosis. First of all, the manual segmentations, on which both models were validated, were seen to have significant errors of concern. The intensity thresholds were established based on a simple visual interpretation, and were seen to differ with a relative large variation between datasets. It could therefore be of interest to see how sensitivity the segmentation process, hence the volume estimation, is to the choice of threshold, *e.g.* when segmenting a simple sphere with a diameter of 12 cm. A 1 mm displacement of the "cranial" boundary will cause in a 2.5 % change in volume. Depending on the sensitivity to the threshold, it could be argued whether the intensity threshold method should be replaced with a model based on intensity gradients, *e.g.* as discussed in [10].

Considering the two model based segmentations, the image registration process, was seen to provide fine results within reasonable time. This model, however, has the disadvantage of being biased against the template image and the aspects of future work is limiting. The graph cut model, which is gradient based, is on the contrary unbiased, and has the advantage of providing a smooth segmentation boundary, especially in the cranial arch. However, difficulties lie within establishing the proper framework setup. To accommodate these difficulties a number of initiatives can be taken, both regarding performance and time.

The graph cut model shows great potential for volume estimation, but as described in the discussion it has some problem areas, where further optimization studies could be of interest in the hope of increasing its performance. One of the main issues was the critical areas in the cranial base, where the radial gradient was seen too low, and the segmentation boundary was thereby influenced by the stronger gradient outside the area of interest. By moving the center for the radial gradient closer to the cranial floor, hence forcing the radial to cross the section in a more perpendicular angle, a stronger gradient might be achieved. Another approach is to modify the gradient contribution, *e.g.* by applying a gradient filter, such that closer lying gradients are emphasized. In this way, the strong influence from *e.g.* teeth and intubation gradients might be softened. A final attempt could be placing several gradient centers, and calculate the final gradient image as a sum of these.

The long processing time and the large amount of memory needed for the graph construction and cutting is also an area of great concern when considering the graph cut model for volume estimation. A speed optimization is therefore needed to compete against the image registration model. Since the graph was forced to lie in between the source and the sink, it was a waste of time to construct the graph, hence the node connections, in these areas. The first attempt would therefore be to construct a masking by the source and the sink. A more extensive masking by the source and the sink will furthermore remedy the high sampling density in the center.

Inspired by the multilevel performance used for the image registration, and the small test performed on varying the samplings density, a scale space analogy could be considered. An initial sparse sampling density would result in coarse representation of the desired transition. By subsequent performing a denser sampling in the area of the transition, the boundary could be found more accurately. A scale space approach could furthermore drastically decrease the processing time without compromising the accuracy of the algorithm.

As mentioned in Section 10.4.1 another aspect of the speed optimization could be to use a triangulation of a sphere in order to avoid the high sampling density around the poles in the defined spherical sampling grid. All the above mentioned optimization initiatives in both speed and performance could entail an algorithm that could perform on images with a resolution of  $512 \times 512$  times the number of slices, within a reasonable time frame, and hopefully on a computer with a standard capacity.

Both models were seen to have difficulties in the cranial base; the graph cut model mainly due to low gradients and obstructing curves, and the image registration mainly due to variations in anatomical interpretations in the manual segmentations. In order to remove the factor of the cranial base, it has been

considered whether it would be possible to add an intersection in the cranial base *e.g.* based upon a small number of anatomical landmarks. Unfortunately, a landmark notation could entail some difficulties, due to the fact that craniofacial anatomy can vary significantly. However, since both models have great success in the segmentation outside these critical areas, a further study of the cost-benefits of such an implementation might be of interest.

Another aspect in volume estimation for these children is the prospect of evaluating the effect of surgery. Information of this can of course be obtained based on the entire volume before and after surgery, but in some cases it might be of higher interest to focus on a specific area, *e.g.* in cases of metopic synostosis, an evaluation of the volume in the frontal part would be indicative. However, as with the intersection in the cranial floor, an anatomical landmark based division of the intracranial volume will be a demanding task, even within the same patient.

The initial idea behind the segmentation was the assessment of the intracranial volume, but another noteworthy application of the segmentations could be a tool for 3D shape analysis of the intracranial volume. This is a medical aspect of almost greater concern than the intracranial volume, especially regarding children with craniosynostosis. As in [26] shape analysis could facilitate a comparison of various parameters for these children, and facilitates comprehensive population studies.

After conversation with Professor, PhD, dr.odont. Sven Kreiborg, the medical aspect of the copper beaten skull sign could be of high interest. The copper beaten sign is the creation of an imprint of the gyri on the inside of the skull, and is a result of a increased intracranial pressure, [30]. If one quantitatively could estimate the level of the cobber beatenness, a potential tool for a higher understanding of the effect could be established. Throughout this study it has, however, not been possible to ascertain whether the models could facilitate such a measure. Further considerations of such an application should therefore be supplied. A quantitative measure would first of all imply that the models can perform on the highest possible resolution, and that the imprint can be detected by the algorithm. The quantitative measure should also involve an estimation of the skull volume "without" the imprinting, which could be hard to obtain. A thought could be to increase the smoothness factor of the graph cut model, which could entail a smooth and non copper beaten surface.

As outlined throughout this chapter, many different prospects lie within optimization of the graph cut model. It is believed, that by implementing some of these consideration, the graph cut model has the potential to perform on equal terms, if not better, as the image registration model.



## Conclusion

---

Throughout this thesis two deformable models have been examined regarding their prospects for intracranial volume estimation in children with unicoronal synostosis. The two deformable models investigated, includes an image registration and a graph cut based algorithm. Both models were chosen on the hypothesis of their ability to overcome the lack of cranial bone information, which can be an issue in these cases of craniosynostosis. Throughout the work, the capability of the two models was established, based on satisfactory segmentation results, compared to manual based segmentations.

In previous craniofacial applications, the image registration model has been used in order to access similar features of skull information. The same model was therefore been utilized in this thesis. The disadvantage of this model is its bias towards a template image, holding a manual segmentation, and as a consequence the graph cut approach was investigated. This model has the advantage of being unbiased, and holds the possibility of a gradient based approach to the problem. An existing graph cut algorithm has been applied and customized to oblige the problematic in this thesis.

The performance of the algorithms were validated by the means of a semi-automatic segmentation including manual editing. An important aspect which must be emphasized is that this model in itself induced a significant error into the problematic. This entails that the foundation for comparison should be

taken with some caution, especially due to the fact that the image registration model is biased against the exact same segmentations. Based on the 15 datasets (14 for the image registration) the mean voxel deviation for the two models was found to be 1.88 % and 3.46 % for the image registration and graph cut model, respectively. Another important aspect was the processing time for each algorithm, found at 9 minutes and 1 hour and 17 minutes, respectively. Based on these results the image registration approach appears to be the most favorable method, however it should be remembered that this model holds a bias towards the template. Since the graph cut application is a relatively new approach for craniofacial segmentations, optimization in both performance and time duration is therefore possible to a great extent.

To conclude, both models have the ability to serve as a tool for further population studies concerning craniosynostosis. Thus the considered study of comparing intracranial volumes in UCS affected children with normal data can easily be extended, to comment on the possible deviations which might be between these two groups. Intracranial volume estimations alone are, however, not always sufficient in order to quantify differences, but the obtained segmentations can furthermore serve as a basis of a possible shape analysis.

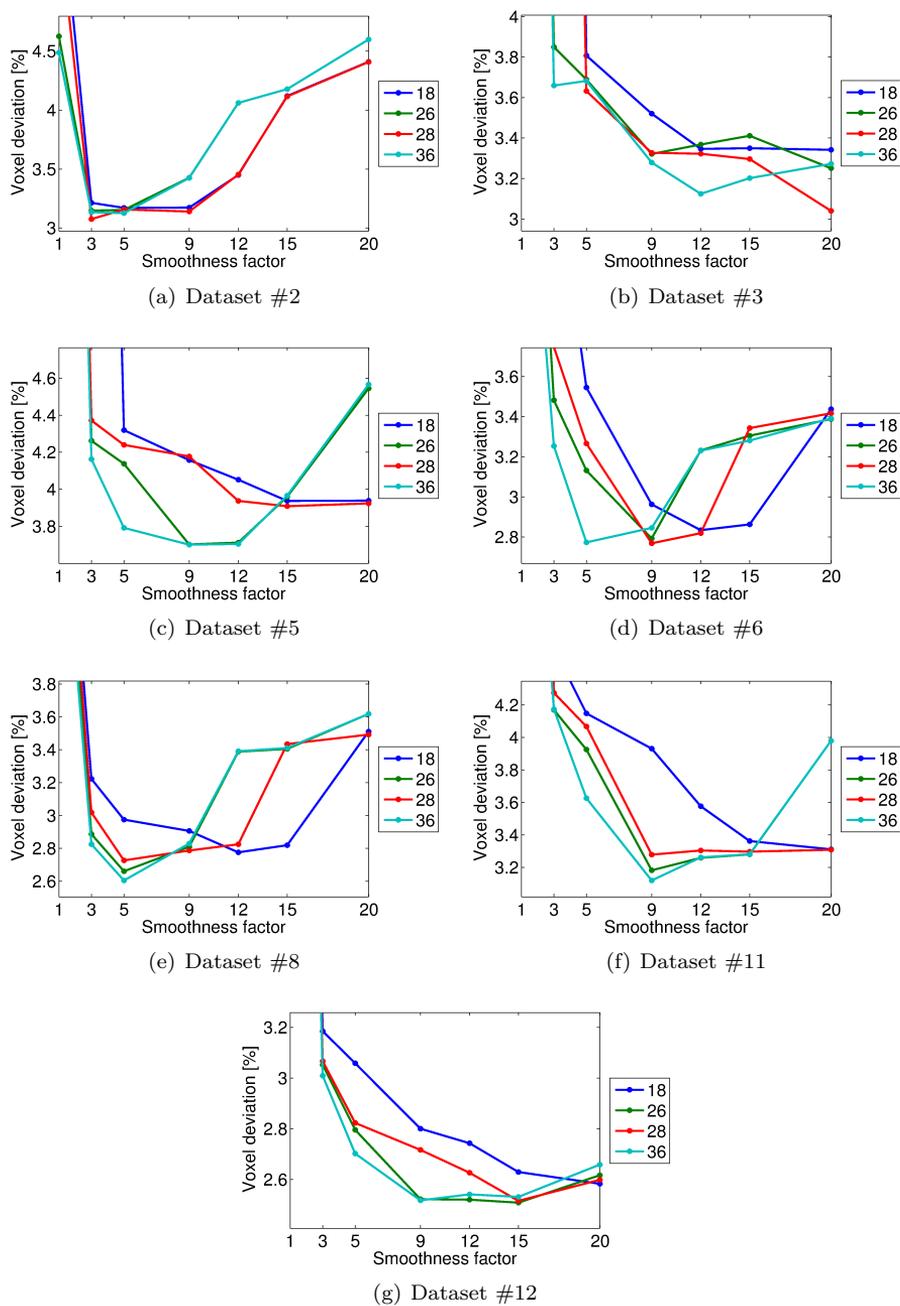
## APPENDIX A

# Parameter test for optimizing the GC model

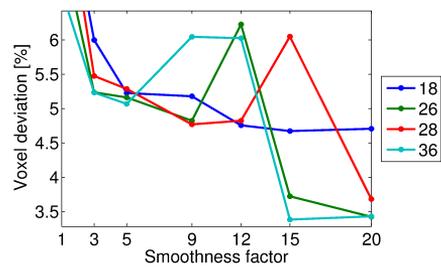
---

Since Figure 10.9 only provides an overview of the mean effect of seven datasets, the following will reveal all the individual parameter tests.

Subsequently, the parameters for dataset #14 were investigated, and the results are seen in Figure A.2. It should be noted that the results from this dataset are not included in the above-mentioned parameter test.



**Figure A.1:** Test of the influence of the smoothness factor (the inter column weight) and the number of connection in the neighborhood for seven datasets. The performance is defined by means of the voxel deviation in percentages compared to the manual segmentation. Note that the graphs are zoomed version.



**Figure A.2:** Test of the influence of the smoothness factor (the inter column weight) and the number of connection in the neighborhood for dataset #14. The performance is defined by means of the voxel deviation in percentages compared to the manual segmentation. Note that the graphs are zoomed version.



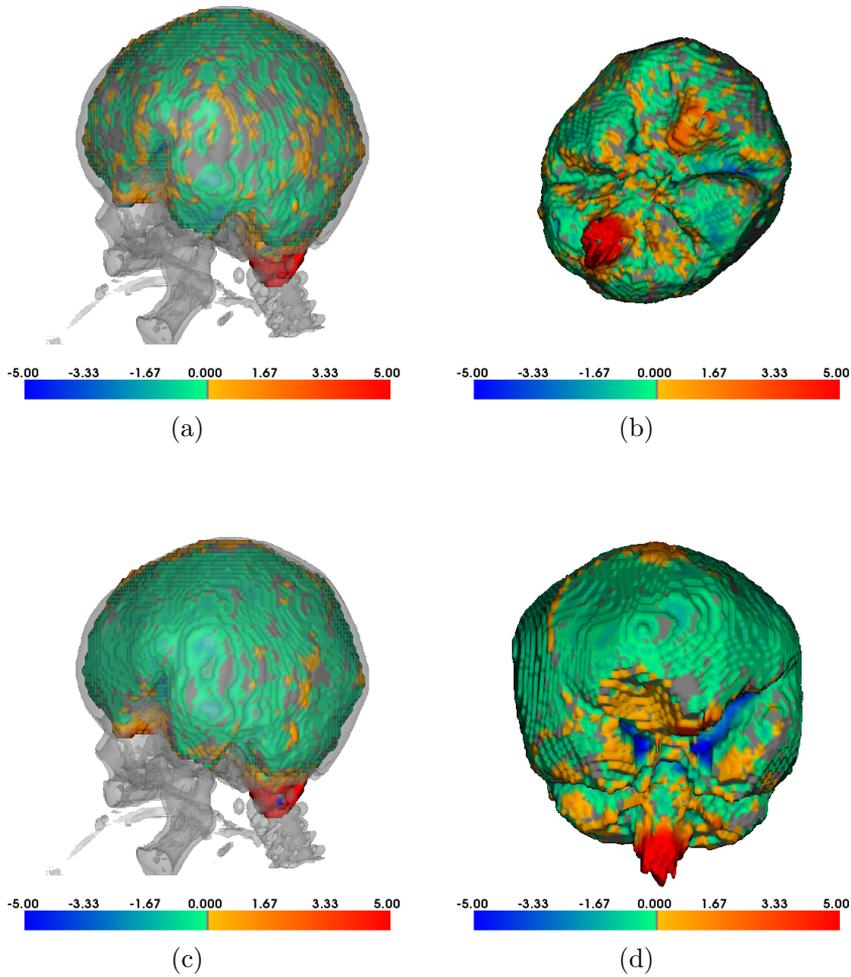
## APPENDIX B

# Results - Illustrations

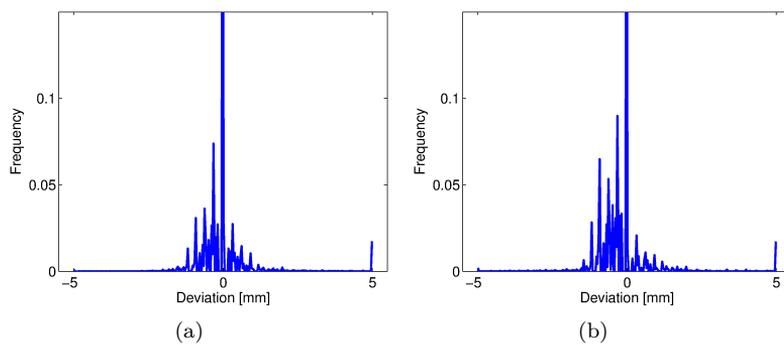
---

In the following pages, the results from both segmentation models are shown for all datasets, except the two, which have been shown in the result chapter in this thesis. First 3D illustrations are presented in two view angles, showing the voxel deviation found as the closest point distance (CPD) between the manual and model based segmentation. These are throughout the appendix referred to as error maps. The extent of the voxel deviations are shown in mm, where green/blue colors indicate a model based segmentation lying on the outside, whereas yellow/red colors indicate a model segmentation located on the inside of the manual segmentation. Further, it must be noted that deviations above 5 mm and below -5 mm are truncated to 5 mm and -5 mm, respectively. It should be noted, that these are presented with slightly different scaling, due to a manual saving process. These are followed by a histogram showing the frequency amount of point-to-point calculations as a function of the extent of the deviation found in the CPD maps. These are all showed in a cropped version, in order to get the best possible visual interpretation. Finally, the segmentation boundaries superimposed on the corresponding CT image are shown in sagittal and transversal plane, respectively.

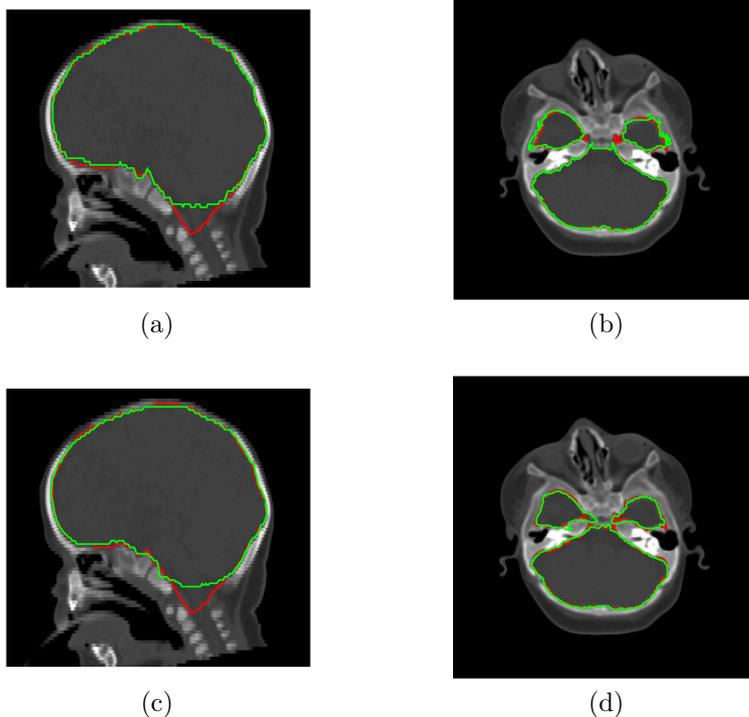
## Dataset #2



**Figure B.1:** Error maps illustrating the extent and location of the voxel deviation for dataset #2. (a) and (b) represent the results from the image registration model, whereas (c) and (d) hold the results from the graph cut model.

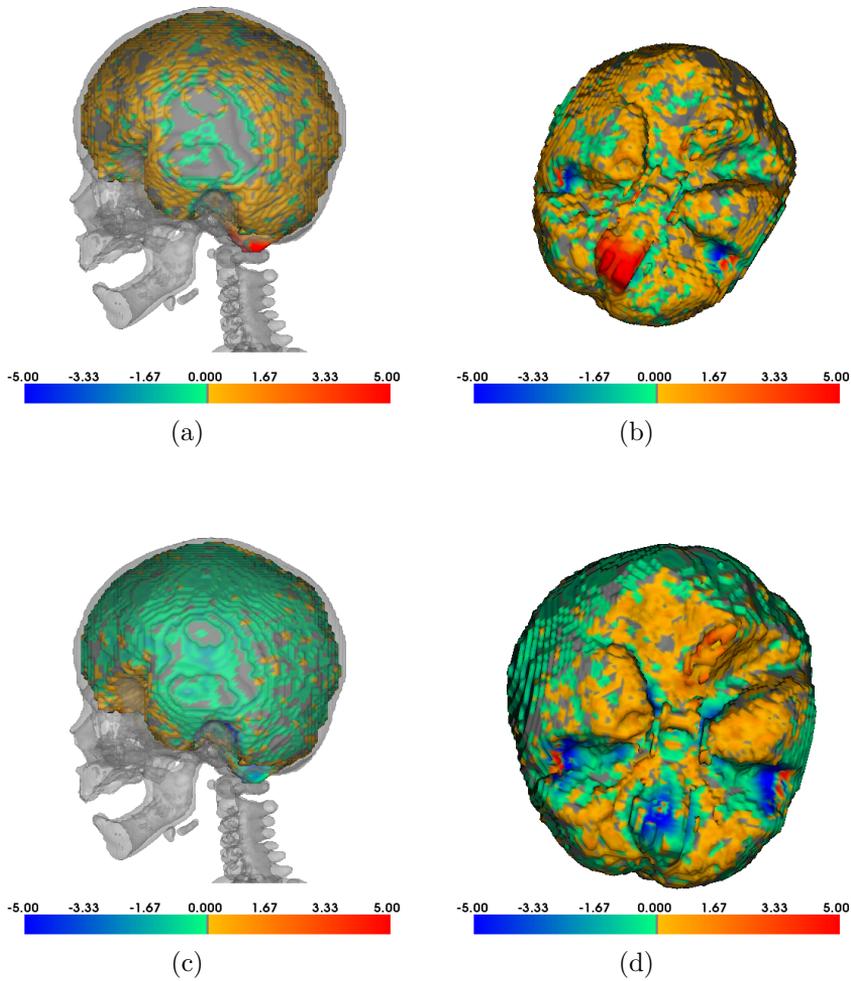


**Figure B.2:** Histograms showing the frequency of point-to-point distances [mm] between the manual and a model based segmentation. (a) holds the image registration result, and (b) the graph cut result, for dataset #2

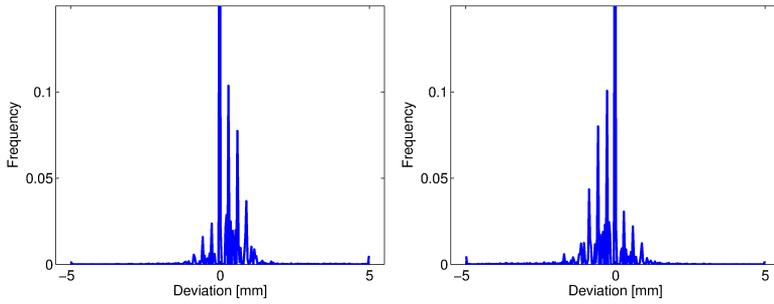


**Figure B.3:** Segmentation boundaries for the manual (red line) and the model based segmentation (green line). (a) and (b) presents the boundary obtained from the image registration, while (c) and (d) holds the boundary for the graph cut, for data #2

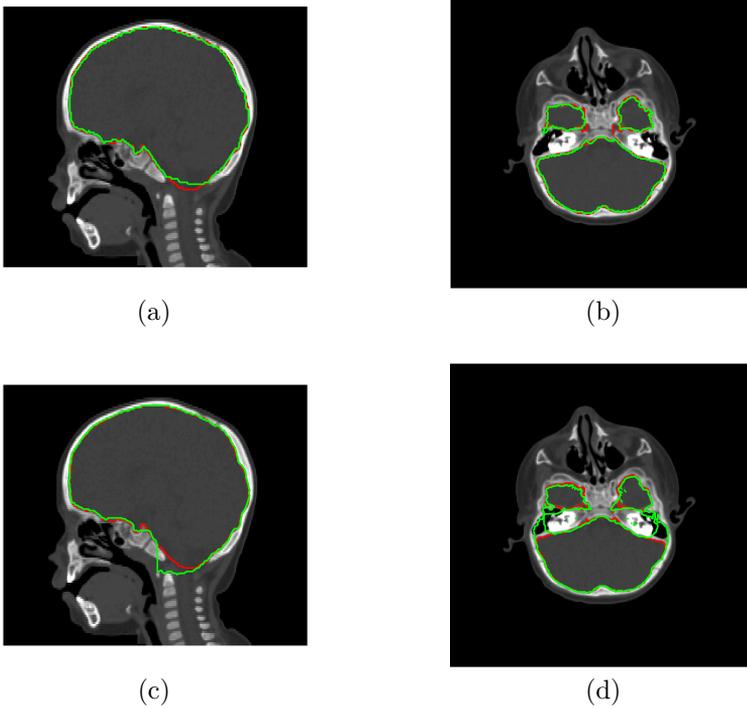
## Dataset #3



**Figure B.4:** Error maps illustrating the extent and location of the voxel deviation for dataset #3. (a) and (b) represent the results from the image registration model, whereas (c) and (d) hold the results from the graph cut model.

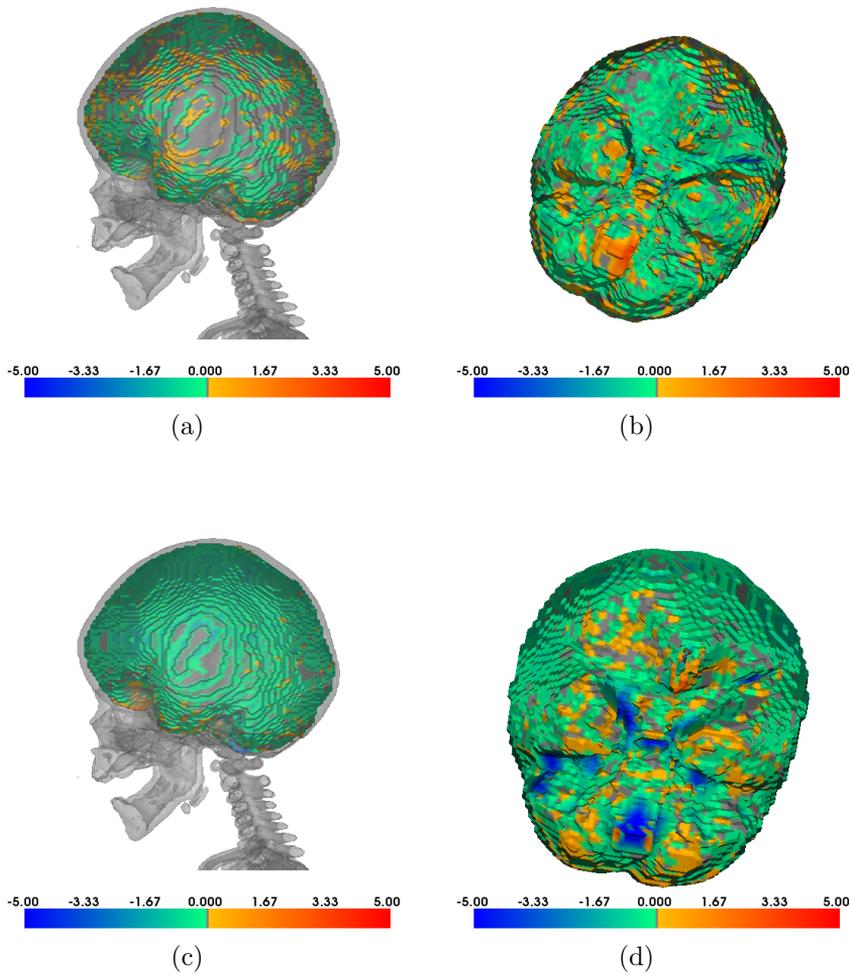


**Figure B.5:** Histograms showing the frequency of point-to-point distances [mm] between the manual and a model based segmentation. (a) holds the image registration result, and (b) the graph cut result, for dataset #3

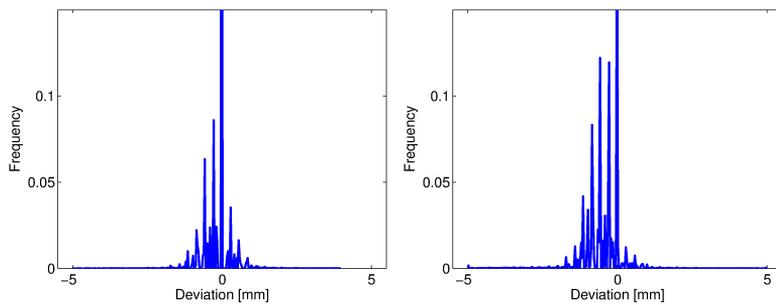


**Figure B.6:** Segmentation boundaries for the manual (red line) and the model based segmentation (green line). (a) and (b) presents the boundary obtained from the image registration, while (c) and (d) holds the boundary for the graph cut, for data #3

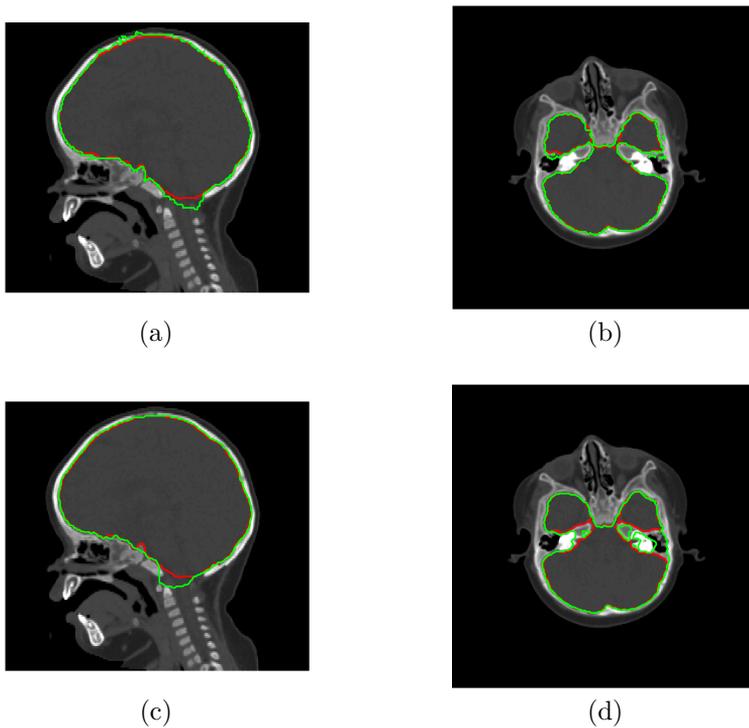
## Dataset #4



**Figure B.7:** Error maps illustrating the extent and location of the voxel deviation for dataset #4. (a) and (b) represent the results from the image registration model, whereas (c) and (d) hold the results from the graph cut model.

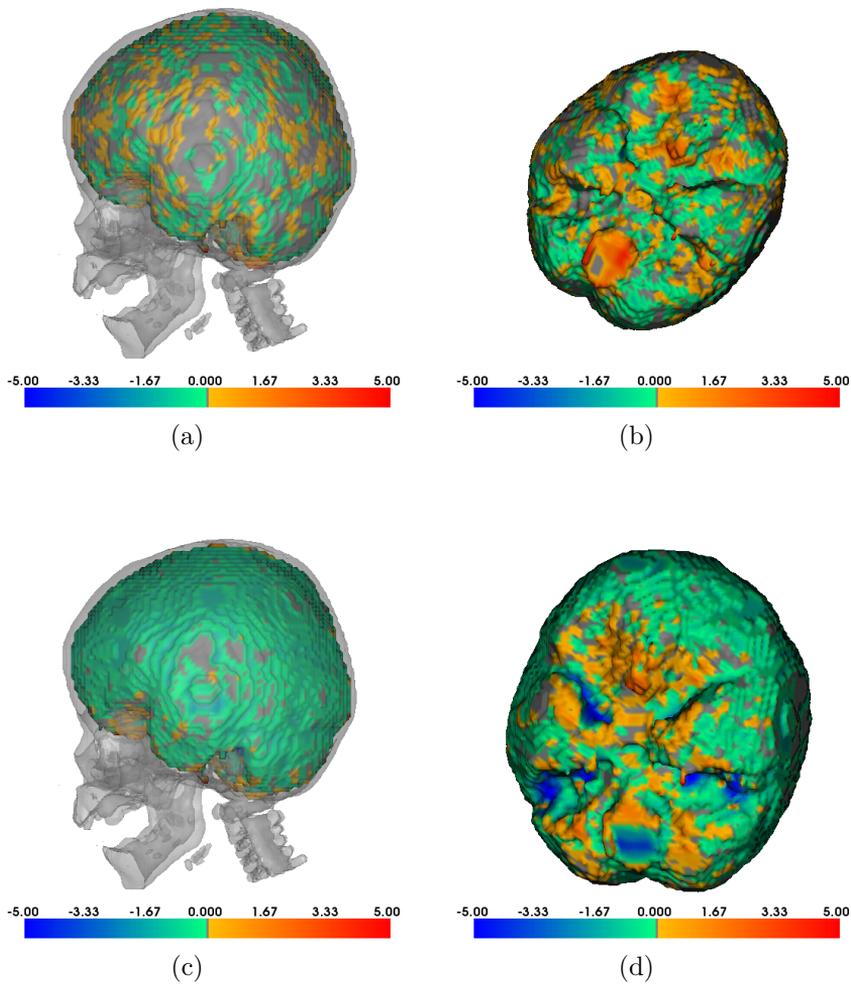


**Figure B.8:** Histograms showing the frequency of point-to-point distances [mm] between the manual and a model based segmentation. (a) holds the image registration result, and (b) the graph cut result, for dataset #4

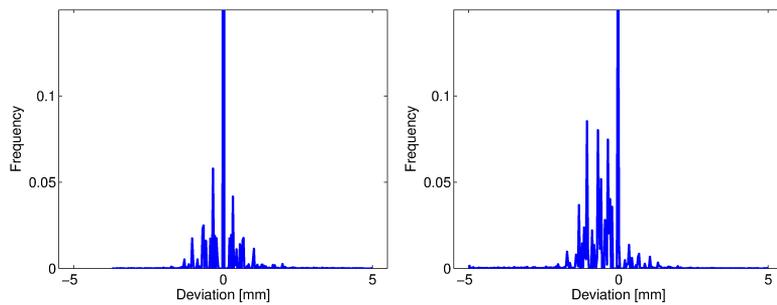


**Figure B.9:** Segmentation boundaries for the manual (red line) and the model based segmentation (green line). (a) and (b) presents the boundary obtained from the image registration, while (c) and (d) holds the boundary for the graph cut, for data #4

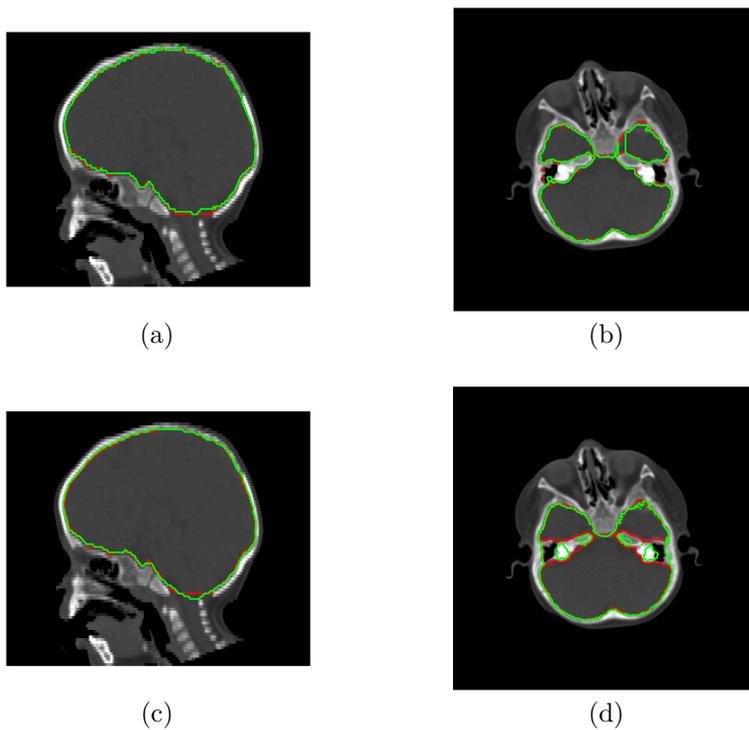
## Dataset #5



**Figure B.10:** Error maps illustrating the extent and location of the voxel deviation for dataset #5. (a) and (b) represent the results from the image registration model, whereas (c) and (d) hold the results from the graph cut model.

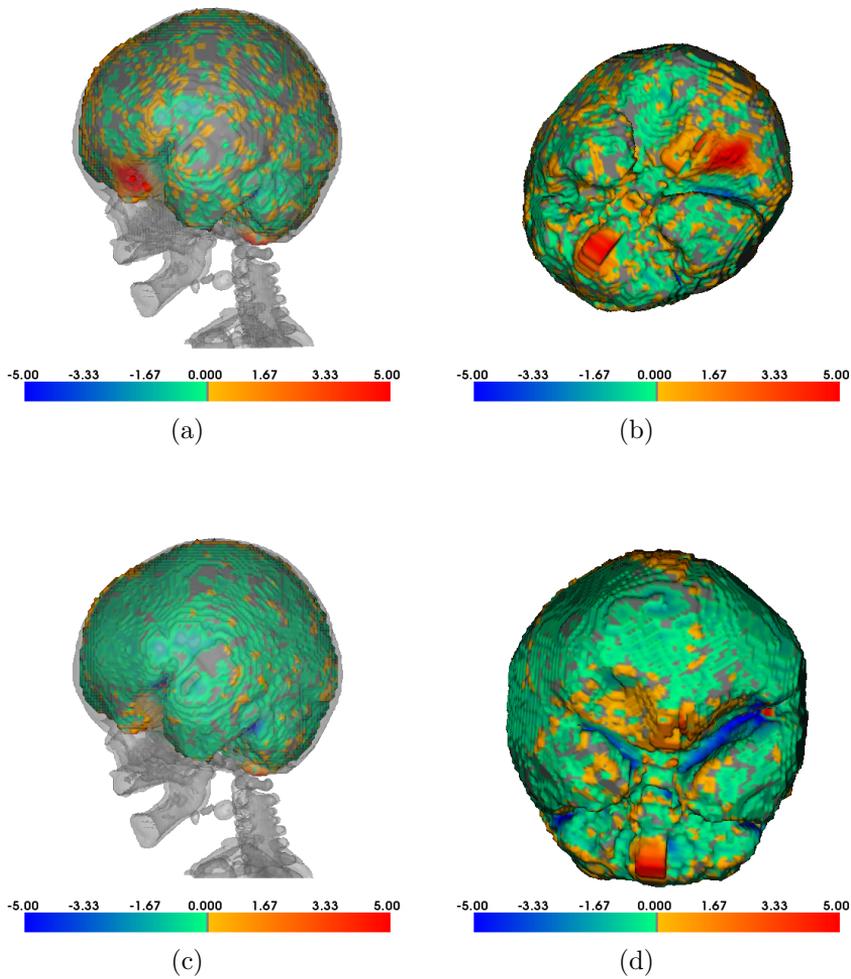


**Figure B.11:** Histograms showing the frequency of point-to-point distances [mm] between the manual and a model based segmentation. (a) holds the image registration result, and (b) the graph cut result, for dataset #5

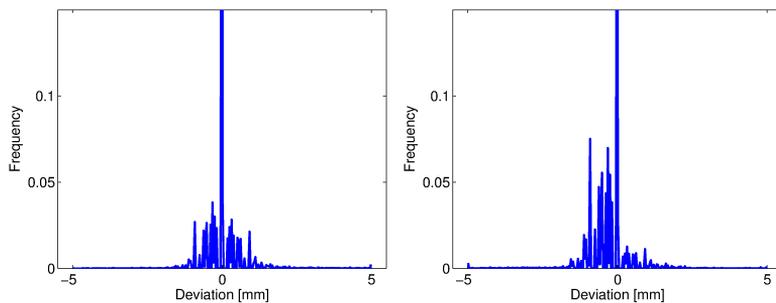


**Figure B.12:** Segmentation boundaries for the manual (red line) and the model based segmentation (green line). (a) and (b) presents the boundary obtained from the image registration, while (c) and (d) holds the boundary for the graph cut, for data #5

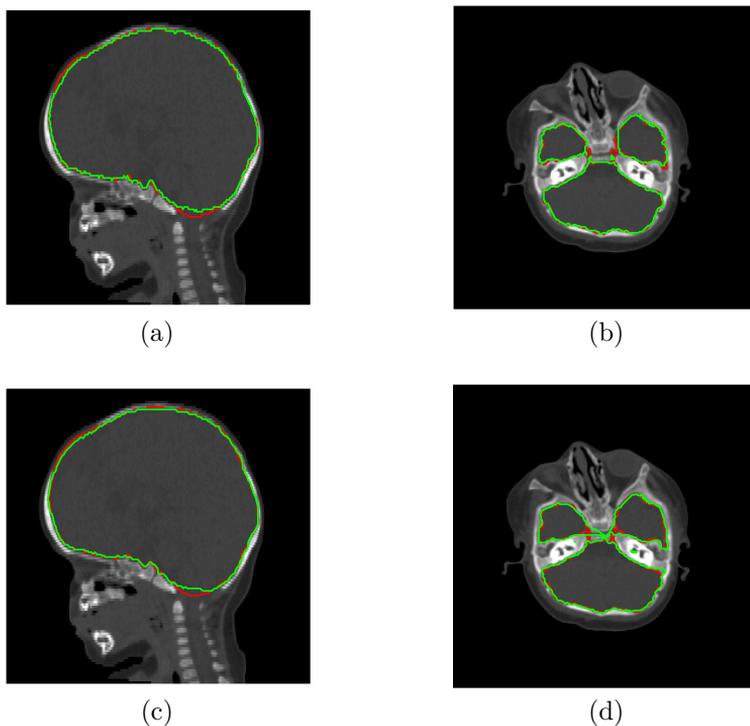
## Dataset #6



**Figure B.13:** Error maps illustrating the extent and location of the voxel deviation for dataset #6. (a) and (b) represent the results from the image registration model, whereas (c) and (d) hold the results from the graph cut model.

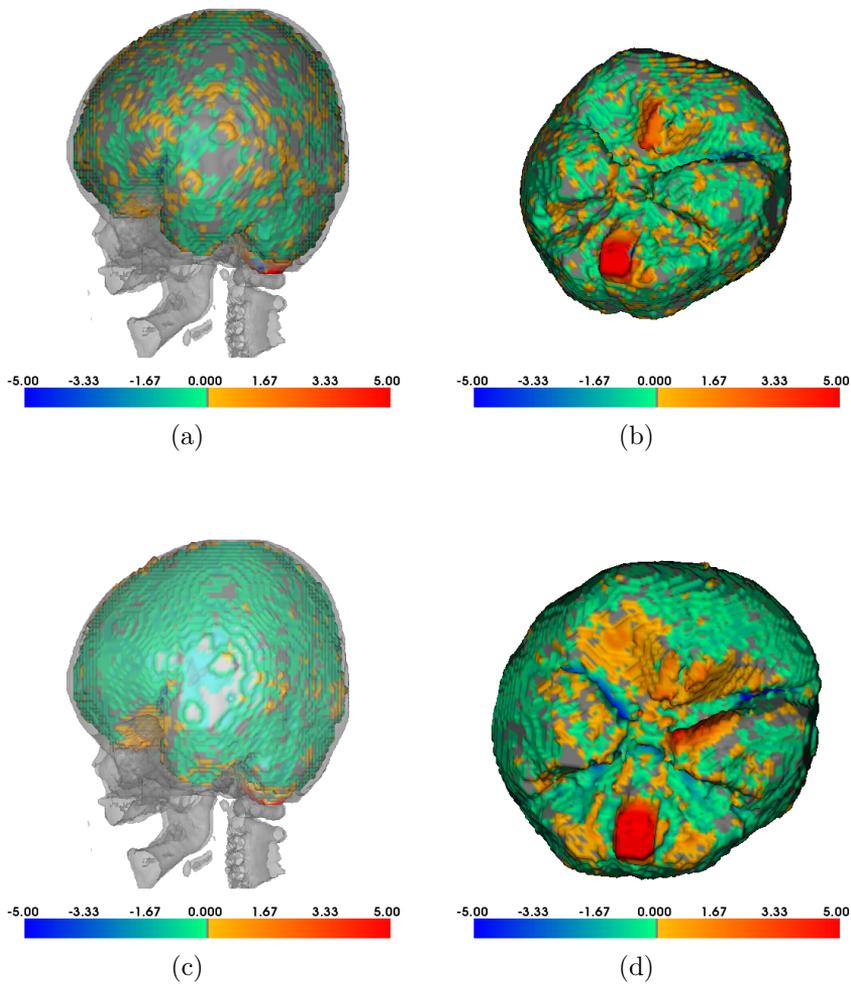


**Figure B.14:** Histograms showing the frequency of point-to-point distances [mm] between the manual and a model based segmentation. (a) holds the image registration result, and (b) the graph cut result, for dataset #6

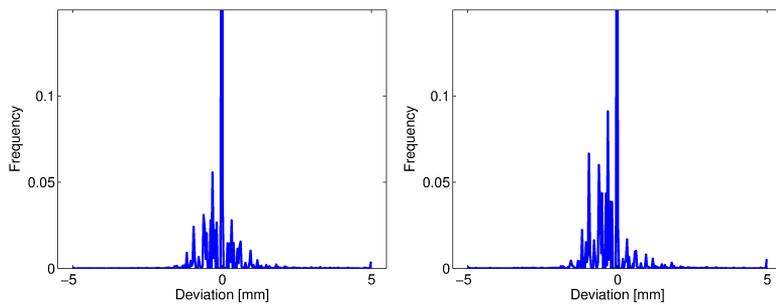


**Figure B.15:** Segmentation boundaries for the manual (red line) and the model based segmentation (green line). (a) and (b) presents the boundary obtained from the image registration, while (c) and (d) holds the boundary for the graph cut, for data #6

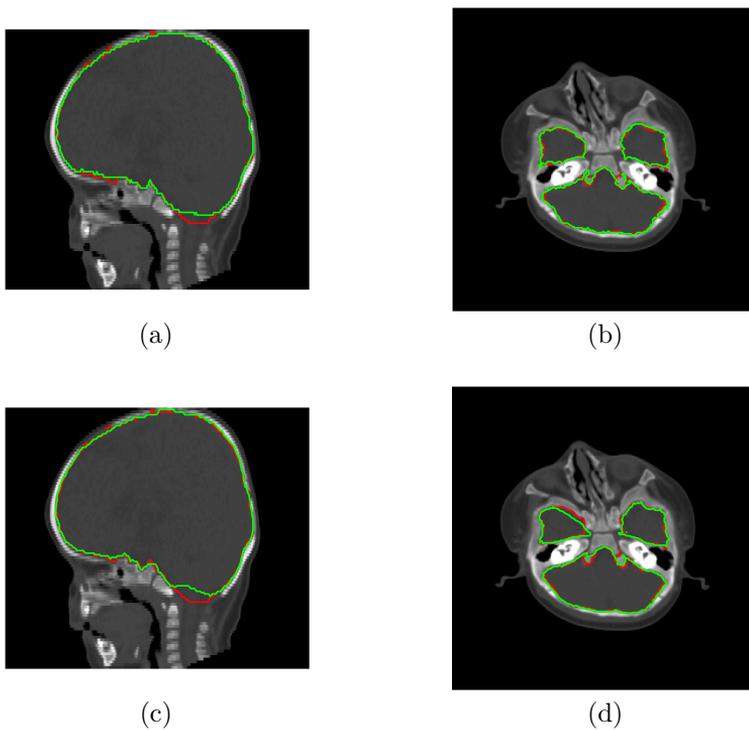
## Dataset #8



**Figure B.16:** Error maps illustrating the extent and location of the voxel deviation for dataset #8. (a) and (b) represent the results from the image registration model, whereas (c) and (d) hold the results from the graph cut model.

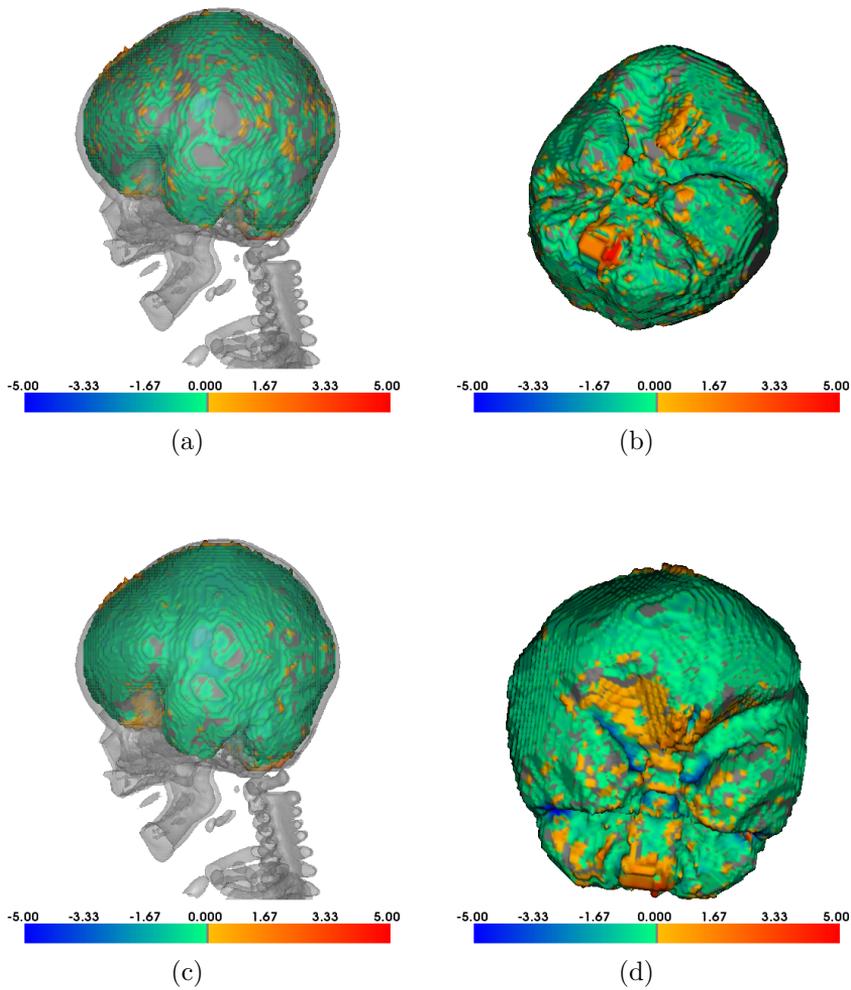


**Figure B.17:** Histograms showing the frequency of point-to-point distances [mm] between the manual and a model based segmentation. (a) holds the image registration result, and (b) the graph cut result, for dataset #8

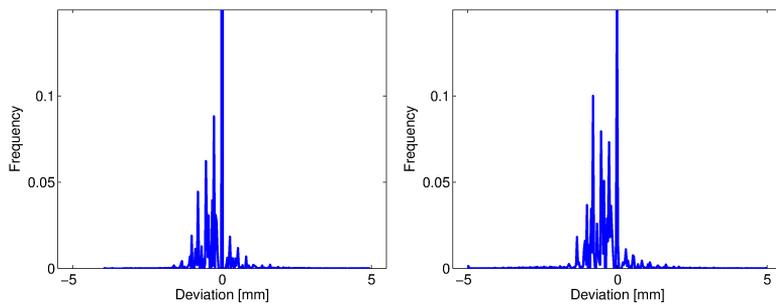


**Figure B.18:** Segmentation boundaries for the manual (red line) and the model based segmentation (green line). (a) and (b) presents the boundary obtained from the image registration, while (c) and (d) holds the boundary for the graph cut, for data #8

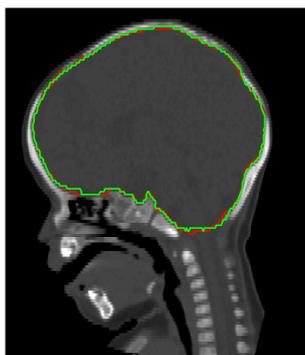
## Dataset #9



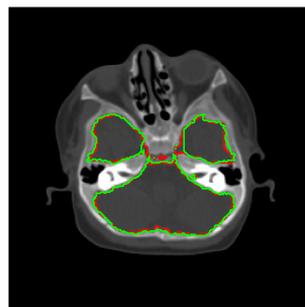
**Figure B.19:** Error maps illustrating the extent and location of the voxel deviation for dataset #9. (a) and (b) represent the results from the image registration model, whereas (c) and (d) hold the results from the graph cut model.



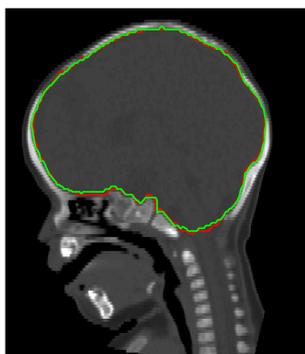
**Figure B.20:** Histograms showing the frequency of point-to-point distances [mm] between the manual and a model based segmentation. (a) holds the image registration result, and (b) the graph cut result, for dataset #9



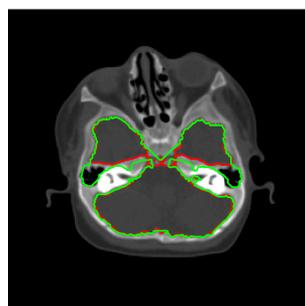
(a)



(b)



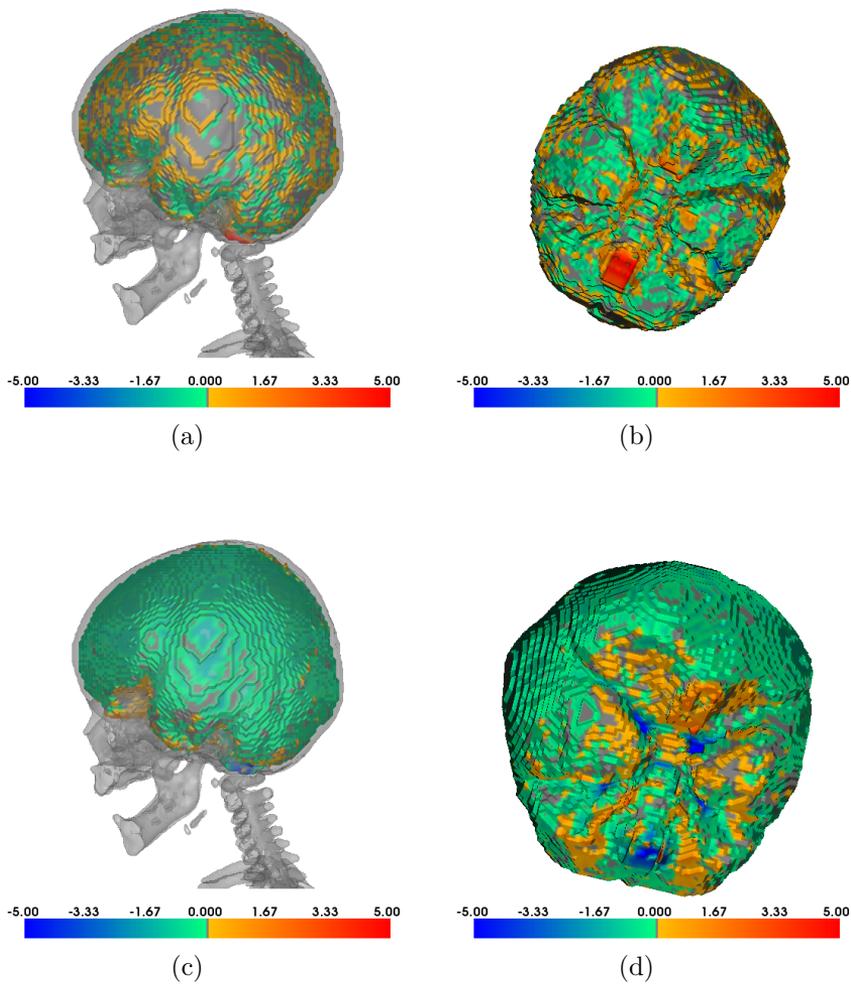
(c)



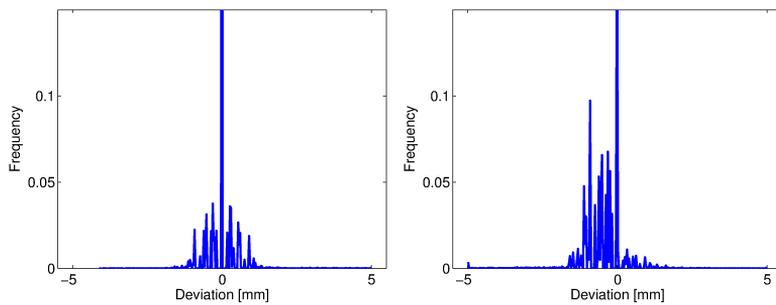
(d)

**Figure B.21:** Segmentation boundaries for the manual (red line) and the model based segmentation (green line). (a) and (b) presents the boundary obtained from the image registration, while (c) and (d) holds the boundary for the graph cut, for data #9

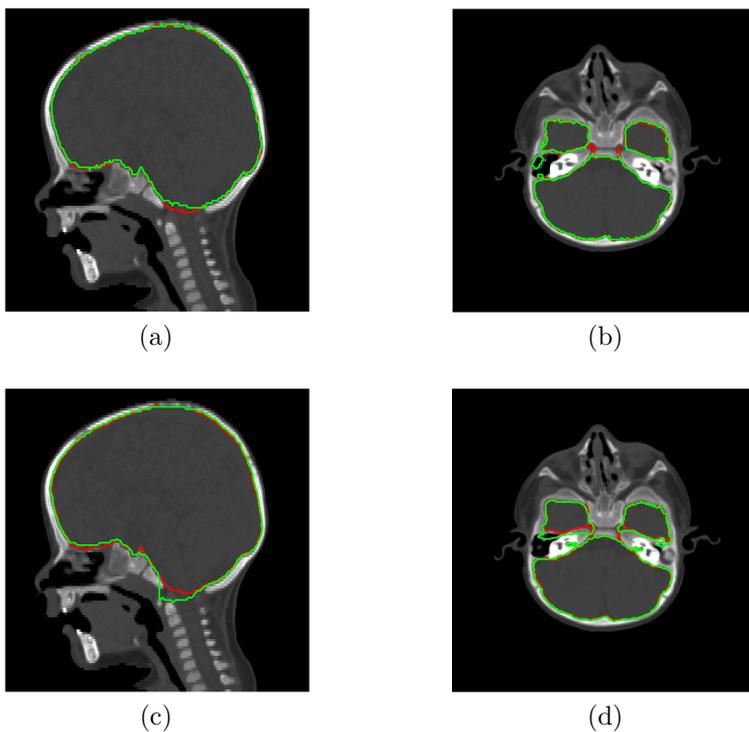
## Dataset #10



**Figure B.22:** Error maps illustrating the extent and location of the voxel deviation for dataset #10. (a) and (b) represent the results from the image registration model, whereas (c) and (d) hold the results from the graph cut model.

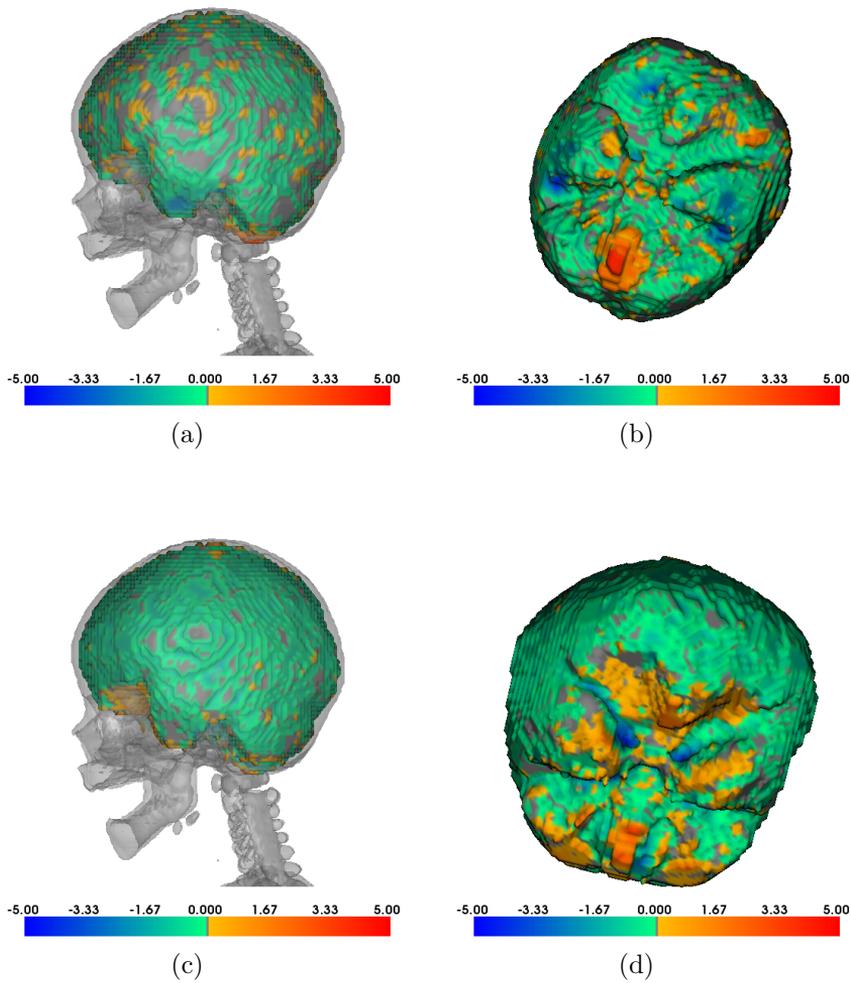


**Figure B.23:** Histograms showing the frequency of point-to-point distances [mm] between the manual and a model based segmentation. (a) holds the image registration result, and (b) the graph cut result, for dataset #10

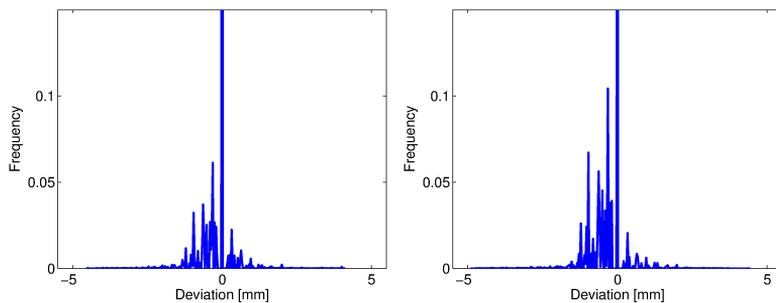


**Figure B.24:** Segmentation boundaries for the manual (red line) and the model based segmentation (green line). (a) and (b) presents the boundary obtained from the image registration, while (c) and (d) holds the boundary for the graph cut, for data #10

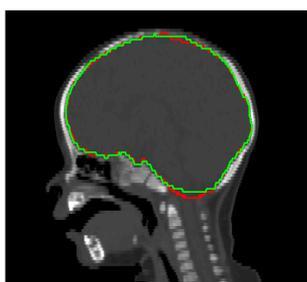
## Dataset #11



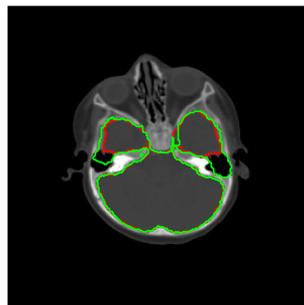
**Figure B.25:** Error maps illustrating the extent and location of the voxel deviation for dataset #11. (a) and (b) represent the results from the image registration model, whereas (c) and (d) hold the results from the graph cut model.



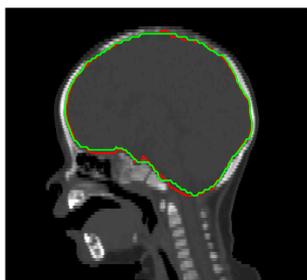
**Figure B.26:** Histograms showing the frequency of point-to-point distances [mm] between the manual and a model based segmentation. (a) holds the image registration result, and (b) the graph cut result, for dataset #11



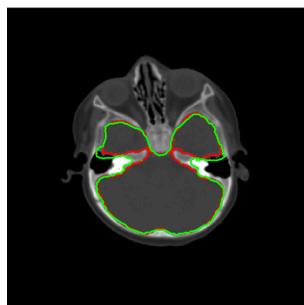
(a)



(b)



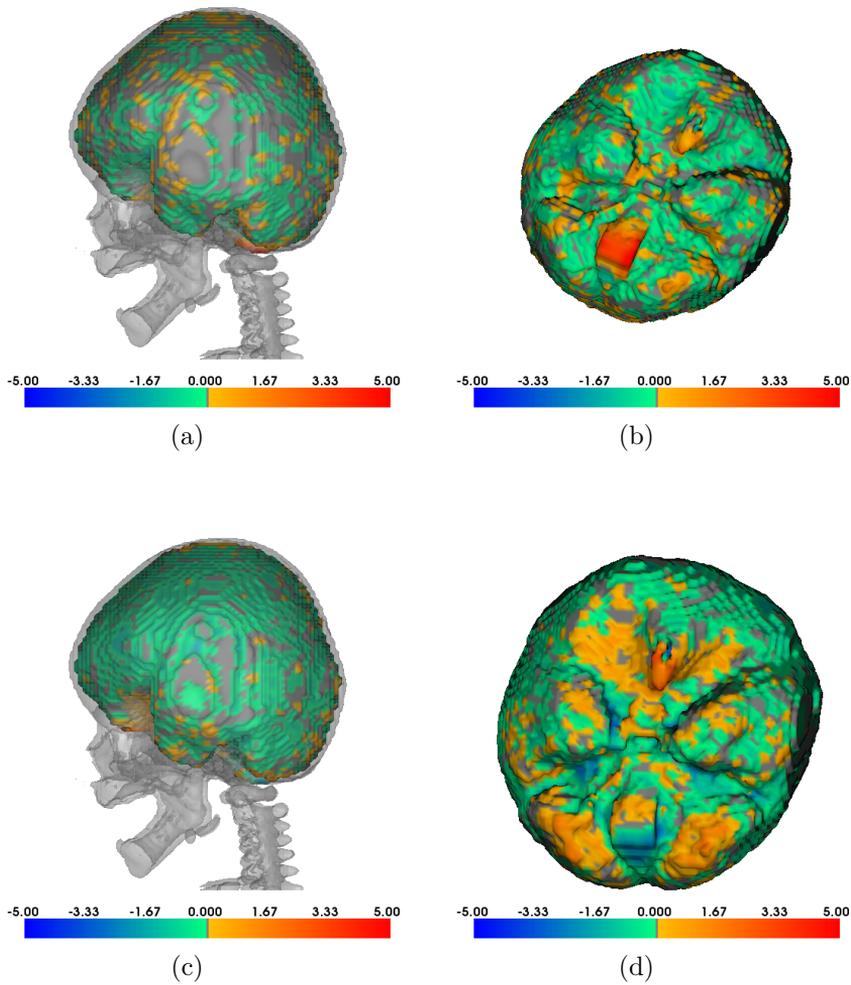
(c)



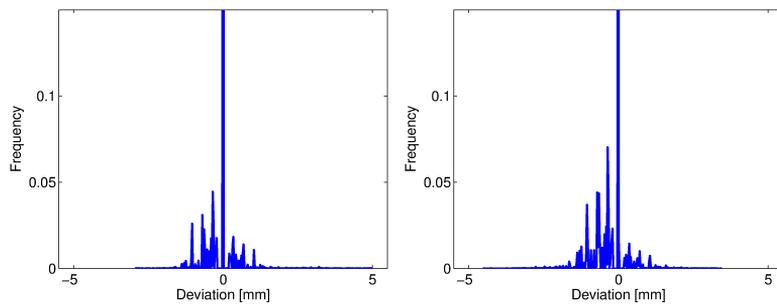
(d)

**Figure B.27:** Segmentation boundaries for the manual (red line) and the model based segmentation (green line). (a) and (b) presents the boundary obtained from the image registration, while (c) and (d) holds the boundary for the graph cut, for data #11

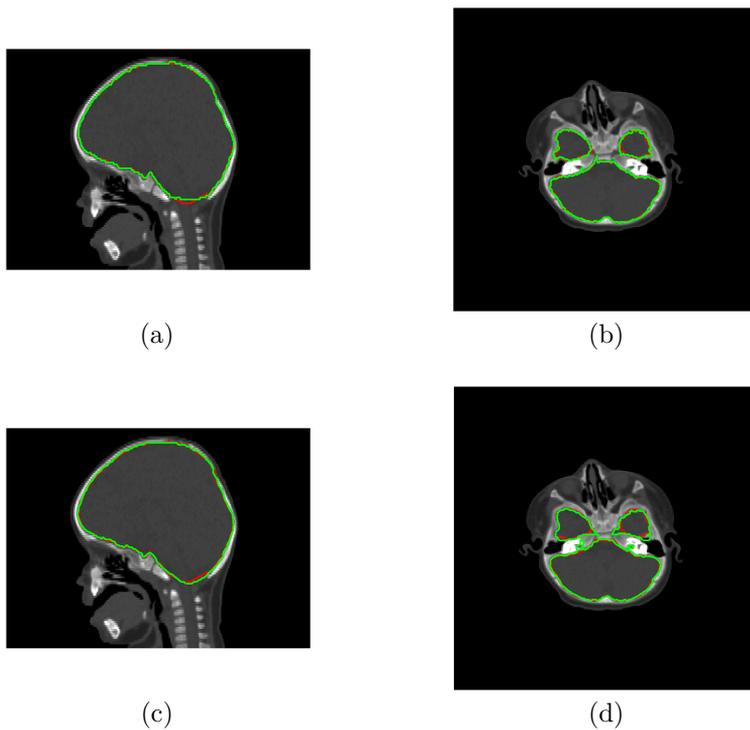
## Dataset #12



**Figure B.28:** Error maps illustrating the extent and location of the voxel deviation for dataset #12. (a) and (b) represent the results from the image registration model, whereas (c) and (d) hold the results from the graph cut model.

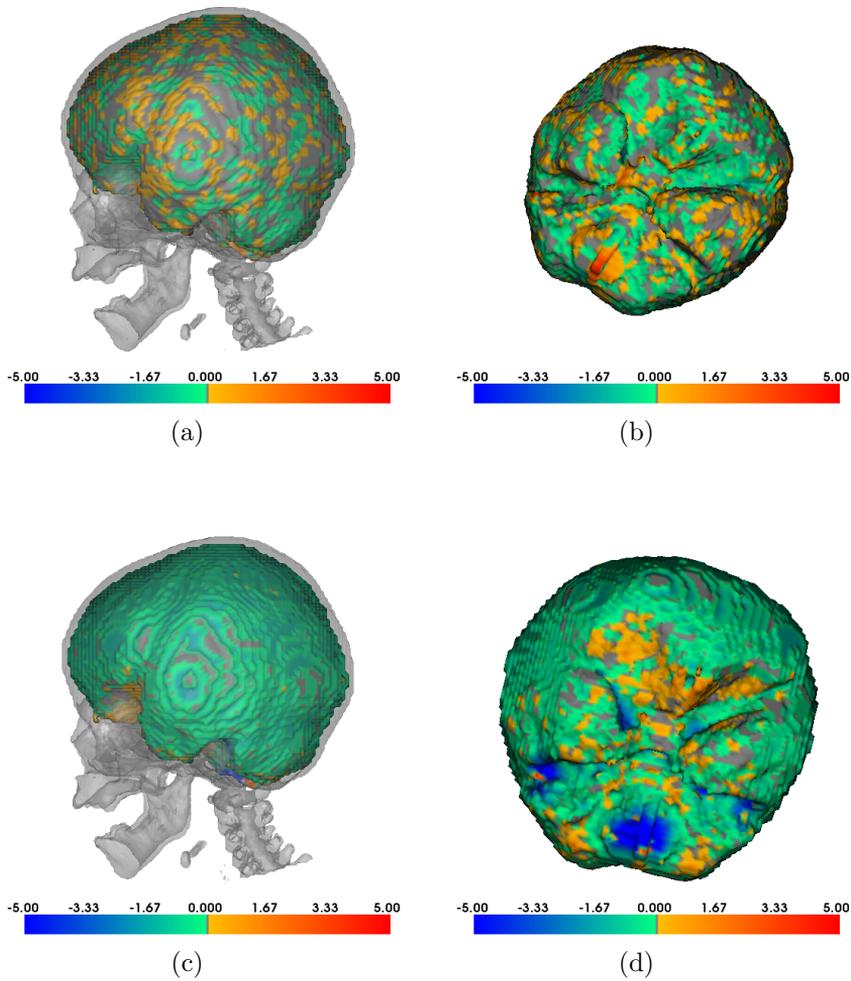


**Figure B.29:** Histograms showing the frequency of point-to-point distances [mm] between the manual and a model based segmentation. (a) holds the image registration result, and (b) the graph cut result, for dataset #12

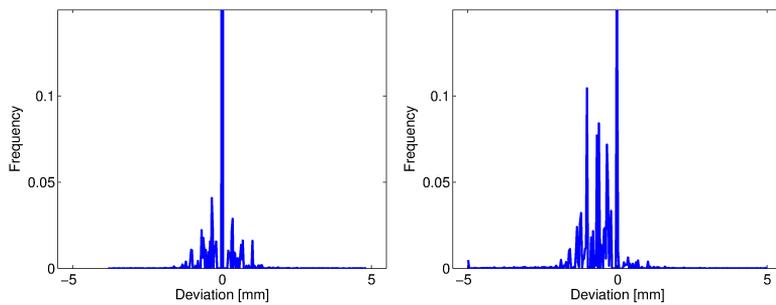


**Figure B.30:** Segmentation boundaries for the manual (red line) and the model based segmentation (green line). (a) and (b) presents the boundary obtained from the image registration, while (c) and (d) holds the boundary for the graph cut, for data #12

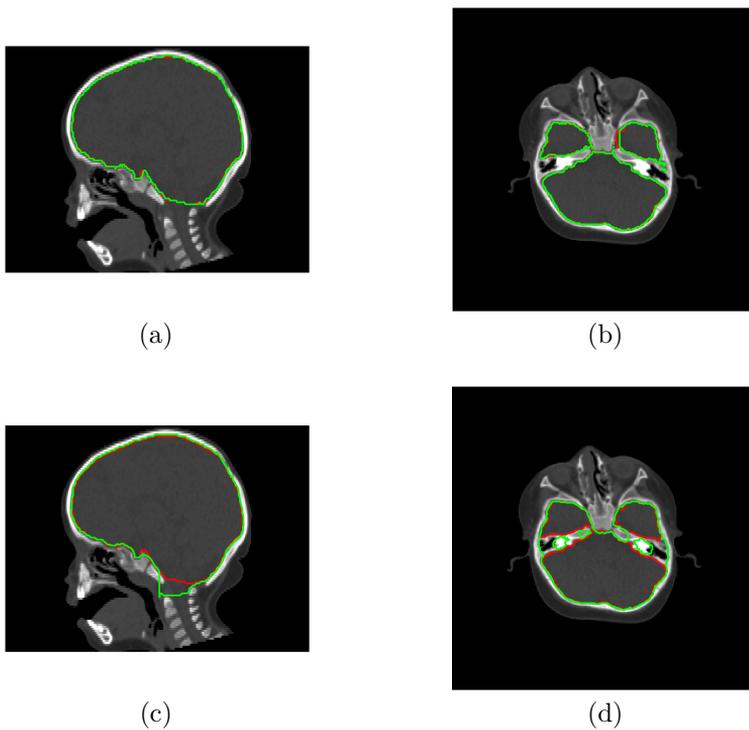
## Dataset #13



**Figure B.31:** Error maps illustrating the extent and location of the voxel deviation for dataset #13. (a) and (b) represent the results from the image registration model, whereas (c) and (d) hold the results from the graph cut model.

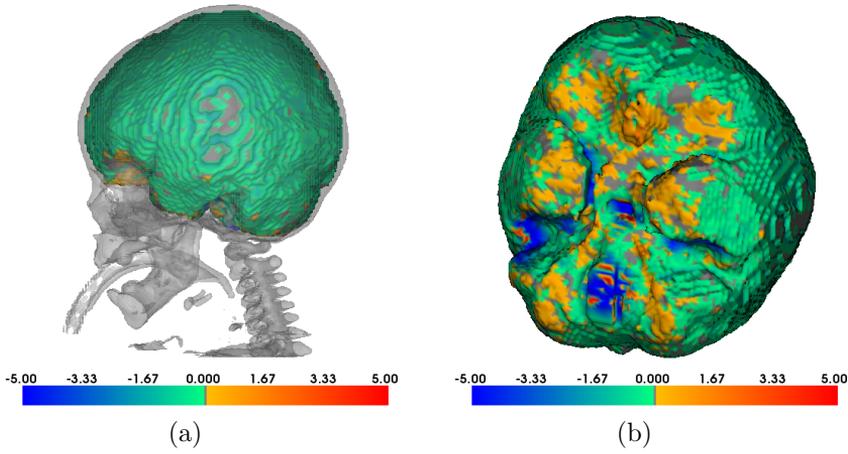


**Figure B.32:** Histograms showing the frequency of point-to-point distances [mm] between the manual and a model based segmentation. (a) holds the image registration result, and (b) the graph cut result, for dataset #13

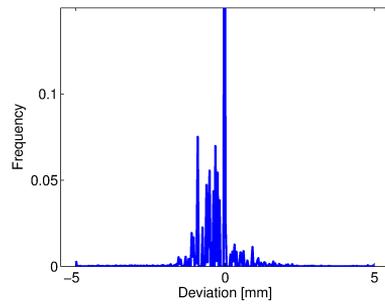


**Figure B.33:** Segmentation boundaries for the manual (red line) and the model based segmentation (green line). (a) and (b) presents the boundary obtained from the image registration, while (c) and (d) holds the boundary for the graph cut, for data #13

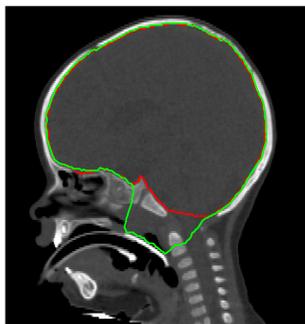
## Dataset #14



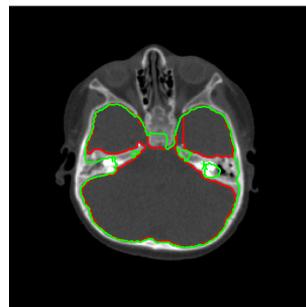
**Figure B.34:** Error maps illustrating the extent and location of the voxel deviation for dataset #14. (a) and (b) represent the results from the graph cut model.



**Figure B.35:** Histograms showing the frequency of point-to-point distances [mm] between the manual and a model based segmentation. (a) holds the graph result, for dataset #14



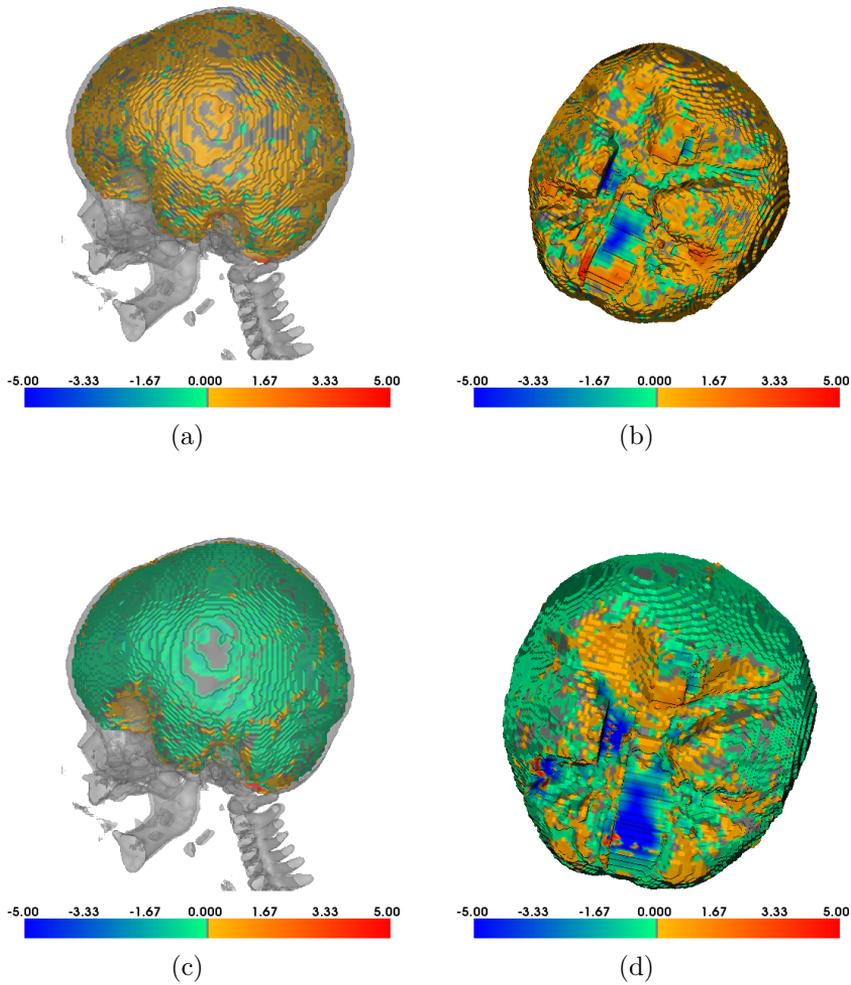
(c)



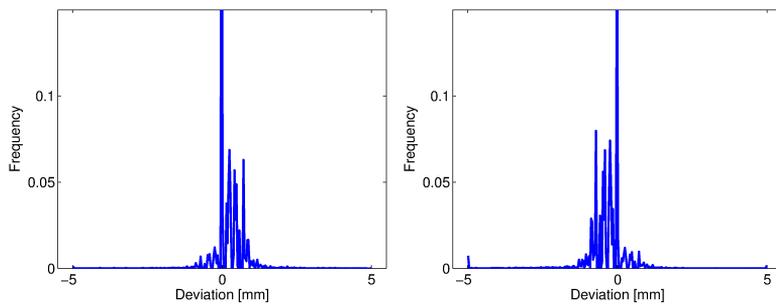
(d)

**Figure B.36:** Segmentation boundaries for the manual (red line) and the model based segmentation (green line). (a) and (b) presents the boundary obtained from boundary for the graph cut, for data #14

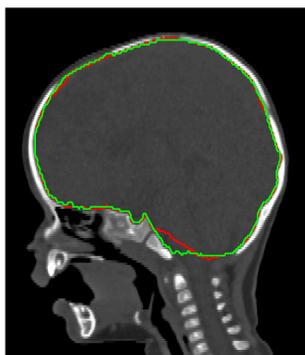
## Dataset #15



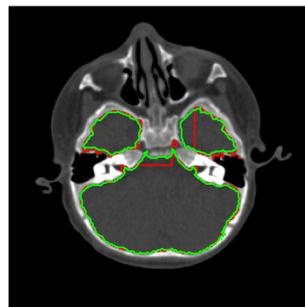
**Figure B.37:** Error maps illustrating the extent and location of the voxel deviation for dataset #15. (a) and (b) represent the results from the image registration model, whereas (c) and (d) hold the results from the graph cut model.



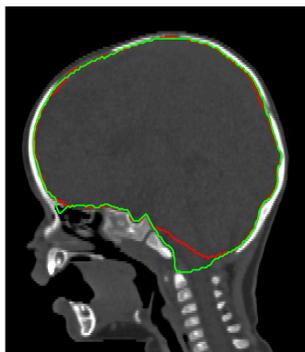
**Figure B.38:** Histograms showing the frequency of point-to-point distances [mm] between the manual and a model based segmentation. (a) holds the image registration result, and (b) the graph cut result, for dataset #15



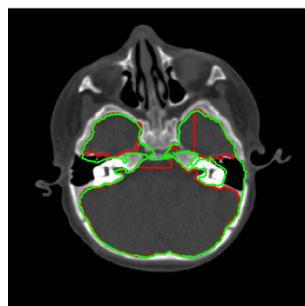
(a)



(b)



(c)



(d)

**Figure B.39:** Segmentation boundaries for the manual (red line) and the model based segmentation (green line). (a) and (b) presents the boundary obtained from the image registration, while (c) and (d) holds the boundary for the graph cut, for data #15



# List of Figures

---

2.1	The normal skull of a newborn with identification of the sutures and fontanelles. From [2]. . . . .	8
2.2	Brain growth (percentage of adult brain weight) as a function of time. It should be noted that the figure depicts brain weight and not brain volume. From [21]. . . . .	9
2.3	Three characteristic manifestations, caused by different types of craniosynostosis. From [25]. . . . .	11
2.4	Left unicoronal craniosynostosis before and after surgery. The two images to the left are preoperative, the middle and right hand images are postoperative, taken in the age of 6 and 15 month, respectively. From [35]. . . . .	12
3.1	Prior probability maps for gray (A) and white matter (C), and the resulting segmentation (B and D). Modified from [6]. . . . .	18
3.2	Illustration of the initial spherical surface (a) and the resulting brain/non-brain surface (b) for the BET algorithm. Modified from [43]. . . . .	19

3.3	3D surface rendering from dataset #6, showing a thresholded CT image of a child with unilateral synostosis. (a) depicts the full skull, and a severe gap between the cranial bones is seen. In (b) the skull base is viewed from the top. The foramen magnum is seen at the bottom of the image. . . . .	21
4.1	Simple illustration, defining the three scan directions. . . . .	25
4.2	Definition of image planes. Modified from [14]. . . . .	26
4.3	Examples of 2D images in the three image planes; coronal, transversal and sagittal. . . . .	26
4.4	Illustration of the four level image pyramid. In (a) the idea of the image pyramid is outlined, with the original image, $L_0$ , with full resolution in front, followed by the lower downsampled levels. In (b)-(e) a slice image from each level is shown. The voxel size is increased for each level, giving a resolution that at level $L_3$ results in a clearly blurred image presentation. . . . .	28
5.1	Same axial slice of a dataset presented with three different thresholds. In (a) a too low threshold is seen, which induces a lot of noise components. The image in (b) has a too high threshold, meaning that some of the wanted information of the cranial bones is lost. In (c) a threshold in between the two previous was selected, which entails that the noise components are removed, while the cranial bones are still remained, more or less, unaffected. . . . .	30
5.2	Surface reconstruction of the image, when using the threshold found in 5.1(c). The surface can in Figure <i>Landmarker</i> be rotated and seen from all angles and thereby give the user a good possibility to evaluate the result. Here an angle is chosen, where the orbits clearly can be seen, showing no holes in that area. . . . .	31
5.3	A sagittal slice showing the seed, given by the green cross, and found outlining illustrated with the pink line (a). In (b) the region inside the line is filled with pink color. . . . .	32
5.4	Two situations where the thresholded segmentation must be manually corrected. Manual corrections are visualized in green. . . . .	33

5.5	Examples of situations where manual editing for larger regions is necessary. . . . .	33
5.6	Example of a segmented mask seen as the red surface superimposed on its corresponding bone surface. . . . .	34
6.1	3D grid illustration. The dots specify the center of each cell. Modified from [5]. . . . .	37
6.2	Simple 2D grid illustration. Modified from [33]. . . . .	37
6.3	The two top images show the template and reference image, respectively. The template represented by the square and the reference as the circle. The aim is now to transform the square to the circle. In the two images below, the initial grid point with the deformation vectors is seen to the left and to the right the resulting warped grid is shown. The bottom image shows the transformed template image, which is obtained when applying the warp grid to the original template image. A high similarity is now seen between the initial reference image and the transformed template. Modified from [27]. . . . .	39
6.4	The cubic spline basis functions given for $-2 \leq x < 2$ . . . . .	42
6.5	Distribution of shifted B-splines in one dimension. . . . .	43
6.6	Line Search with a too large step size in(a) and a too small step in (b). Idea from [20]. . . . .	46
6.7	Illustration of the Armijo condition. The blue curve indicates the cost function, while the dotted line defines the gradient. When the cost function lies below the gradient, the corresponding $\alpha$ values are accepted. Modified from [1]. . . . .	47
6.8	Three curves illustrating a multilevel problem with three levels. Green = coarsest level, Blue = coarse level and Red = fine level. It is seen that the coarser the level is, the less fluctuating the curve and a significant decrease in the number of local minima is seen. Modified from [32]. . . . .	49

- 7.1 The concept behind the augmenting path. (a) The initial graph setup. The terminal nodes from the source set the maximum possible flow to 29. (b) The shortest path from the source to the sink with the highest possible flow is found and marked in red. The residual flow is then decreased by the cost of the edge that is saturated, here 12. (c) Repetition of the procedure results in at least one new edge saturation and a decrease of the residual flow by 7. (d) A new path is now necessary in order to saturate an edge. The flow thereby decreases by 4. (e) No further flow can be pushed through from the source to the sink due to the saturated edges. From the Ford and Fulkerson theorem the minimum cut is therefore found at these saturated edges. Idea from [9]. . . . 53
- 7.2 Search tree setup for a short overview of the min-cut/max-flow algorithm and its terminology. Red nodes,  $S$ , are connected to the search tree with root in the source, whereas blue nodes,  $T$ , belong to the sink. The labeling with  $A$  and  $P$  specifies the active and passive nodes, respectively. The free nodes are marked with black. The found path from the source,  $s$ , to the sink,  $t$ , at the end of a growth stage is marked with yellow. Edited from [7]. . . 54
- 7.3 Nodes and edges in an  $s$ - $t$  cut. The terminology of links and edges are explained with different line types. The t-links are edges to terminal nodes (thick lines), whereas n-links connect the nodes (thin lines). Intra column edges link the direct flow from the source to the sink (full and dotted lines), whereas the inter column edges connect to nodes in another column (dot-dash lines). Data term edges have costs based on the data set (full lines), and infinity cost edges facing backwards are applied to ensure a unidirectional flow from  $s$  to  $t$  (dotted lines). With these two opposite edges, the graph is said to be a directed graph. 56
- 7.4 The effect of the inter column weights. For the purpose of illustration, a value of 10 and 1 is assigned to the high (thick black lines) and low (black dashed lines) data term edges, respectively. In subfigure (a) the dark purple inter column weights are set to 1, whereas they in (b) are set to 2. The two optimal minimum cuts are superimposed on the two figures. A smoother cut is seen to be enforced in (b) due to the higher inter column cost. The nodes marked with a red contour are thereby assigned to the source. . . 57

- 
- 8.1 An example of a transversal slice of the original data. The head is seen in the middle in light gray. The darker gray circle indicates the actual scan tube, wherein two slightly visible gray areas at each side of the head is stabilizing objects. The lines in the bottom indicate the scan bed. . . . . 60
- 8.2 An illustration of the landmark locations (red dots). The landmarks are superimposed on a polygonal surface rendering based on the HU value of skin. . . . . 60
- 8.3 Preprocessing steps in removal of background noise. The steps are further elaborated in the text. . . . . 62
- 8.4 Two transversal CT slices from two different datasets. The reference (green) and template image (red) seen before and after application of the parameters from the principal axis transformation. The RGB image space is used in order to visualize both images concurrently. . . . . 64
- 8.5 Reference and template image, with principal axis and center of mass (blue dot). The first principal axis is visualized in red and the second in green. . . . . 64
- 8.6 Simple illustration of a reference (a) and template (b) image, where the first principal axis for the reference image is found in the negative direction, hence opposite its corresponding vector from the template image. This will give a 180 degree rotation as seen in (c). . . . . 65
- 8.7  $T_1$  and  $R_1$  defines the principal axis of a template and reference, respectively. The angle  $\beta$  between the two vectors is seen to be above 100 degrees, and the  $R_1$  vector is therefore multiplied by minus one, which gives the vector  $R_{1New}$  illustrated with the dotted arrow. The rotation angle between  $T_1$  and  $R_{1New}$  will now provide a correct alignment. . . . . 66
- 9.1 3D surface rendering of the template image, dataset #14, with its intracranial volume found from manual segmentation. . . . . 71

9.2	Illustration of an image registration procedure. The top image shows the reference image, for which intracranial volume information is wanted. In order to obtain the wanted information, a known template image is registered to the reference. This is conducted by iteratively minimizing the sum-of-squared differences between the template and the reference image. For each iterative step a parameter set, $w$ , is obtained, providing the information for the given transformation. When the process reaches its minimum, hence its optimal solution, defined on the basis of a number of stopping criteria, the parameters providing the final transformation are referred to as optimal, $w_{opt}$ . These parameters are extracted, and by applying $w_{opt}$ to the mask of the template, which is known, the intracranial volume for the reference image is found. . . . .	73
9.3	Illustration of voxel deviation after affine transformation as a function of five level situations. . . . .	74
9.4	Illustration of voxel deviation after B-spline transformation as a function of five level situations. An affine transformation on $L_2$ has been applied to all cases prior to the B-spline transformation.	75
9.5	Voxel deviation after affine transformation on $L_2$ followed by the B-spline transformation performed on $L_3-L_1$ as a function of nine different regularization parameters for five different datasets. . .	77
9.6	Deformation grids for the B-spline transformation on the three levels. . . . .	78
10.1	Walk through of the graph cut implementation. (a) 3D surface visualization based on the HU bone value. (b)-(c) Collapsed image with the source and sink superimposed in red. (d) Transversal slice of the head. (e) Transversal slice from the 3D radial gradient. Only positive gradient values are of interest. (f) Resampled version of the radial gradient in spherical coordinates. (g) The cut found in the spherical coordinates. (h) Resulting graph-based segmentation. (i) Manual segmentation with superimposed closest point deviation [mm] found between the manual and graph-based segmentation. . . . .	80
10.2	Surface representation of the 3D test set consisting of an inner (red) and a outer (blue) ellipsoid. . . . .	82

10.3	A transversal (a) and sagittal (b) slice of the positive 3D radial gradient. . . . .	83
10.4	Spherical resampling of a sphere (a) and resulting terrain-like surface (b). The dark red and dark blue surfaces represent the sink and the source, respectively. Each point on the source has a corresponding point at the sink, and the line between represents a column which cuts the surface only once. . . . .	84
10.5	Visualization of the definitions for converting spherical coordinates to Cartesian coordinates and <i>vice versa</i> . . . . .	84
10.6	The spherical scheme for resampling. Note that the sampling density is exaggerated in order to visualize the sampling spheres created by varying $\phi$ and $\theta$ , and their radius controlled by $r$ . With this spherical sampling grid, the number of sampling points is constant in each sampling sphere. . . . .	86
10.7	Investigation of the effect of the sampling density. $ss$ controls the sampling points in each sampling sphere, whereas $rd$ controls the density of the spheres. The performance is evaluated based upon the voxel deviation given in percentage. . . . .	86
10.8	Number of connections associated with the sampling point in the center of the structures. The structures are extracted from the spherical sampling matrix, so the orientation is the same as in Figure 10.4. $\phi$ is increasing from right to left, $\theta$ from top to bottom and $r$ from front to back. . . . .	88
10.9	Test of the influence of the smoothness factor (the inter column weight) and the number of connection in the neighborhood. In (a) the performance is defined by means of the mean voxel deviation in percentages compared to the manual segmentation. (b) is a zoomed version of (a). (c) summarizes the mean processing time in hours for the weight construction and graph cutting. Seven datasets contribute to the results, and their individual results can be inspected in Appendix A. . . . .	89

10.102D visualization of the spherical sampling points on which the different majority counts are performed. Note that the 2D visualization entails no sketch of the sampling points when varying $\phi$ . The blue circles mark the transformed image voxel of interest. (a) is a "majority count" only based on the nearest sampling point (red dot). (b) is a majority count based on seven sampling points, where the closest sampling point again is chosen together with its six neighbors. (c) is the majority count in use, where the eight points that encapsulate the converted Cartesian grid points constitute to the majority. . . . .	91
11.1 Error maps, illustrating the voxel deviations between the two users for dataset #1, calculated as closest point difference with U1 as source. The deviations are given in millimeter, where a green/blue color indicates that U2 lies on the outside of the source, whereas a yellow/red color indicates that U2 lies on the inside of the source segmentation. Deviations above 5 mm and under -5 mm are truncated to 5 mm and -5 mm, respectively. Further, it should be noted that (a) depicts the bone surface, whereas (b) facilitate a view of the bottom of the segmentation mask (foramen magnum is found in the bottom of the image.) . .	95
11.2 Histogram showing the frequency of point-to-point calculations as a function of deviation between to two manual segmentations performed on data #1. The histogram is presented in a cropped version to facilitate a better visualization. . . . .	95
11.3 Segmentation boundary for the manual (red line) and the model based segmentation (green line) for data #1, superimposed on the corresponding CT image. (a) and (b) present the results from the image registration, and (c) and (d) the graph cut. . . . .	98
11.4 Segmentation boundary for the manual (red line) and the model based segmentation (green line) for data #7, superimposed on the corresponding CT image. (a) and (b) present the results from the image registration, and (c) and (d) the graph cut. . . . .	99
11.5 Surface rendering illustrating the difference between the manual segmented mask (underlying red), and the image registration segmented mask (green) in (a) and the graph cut segmented mask (blue) in (b) for dataset #1. . . . .	100

11.6	Surface rendering illustrating the difference between the manual segmented mask (underlying red), and the image registration segmented mask (green) in (a) and the graph cut segmented mask (blue) in (b) for dataset #7 . . . . .	100
11.7	Illustration of the extent and location of the voxel deviation for dataset #1, given as the closest point differences found between the manual and model based segmentation, superimposed on a 3D rendering of the manual segmentation. (a) and (b) present the results from the image registration, and (c) and (d) the graph cut. The extent of the deviations are shown in millimeter, where a green/blue color indicates a model based segmentation lying on the outside, whereas a yellow/red color indicates a model segmentation located on the inside of the manual segmentation. Furthermore, it should be noted that deviation above -5 mm and below -5 mm are truncated to 5 mm and -5 mm, respectively. . . . .	102
11.8	Illustration of the extent and location of the voxel deviation for dataset #7, given as the closest point differences found between the manual and model based segmentation, superimposed on a 3D rendering of the manual segmentation. (a) and (b) present the results from the image registration, and (c) and (d) the graph cut. The extent of the deviations are shown in millimeter, where a green/blue color indicates a model based segmentation lying on the outside, whereas a yellow/red color indicates a model segmentation located on the inside of the manual segmentation. Furthermore, it should be noted that deviation above -5 mm and below -5 mm are truncated to 5 mm and -5 mm, respectively. . . . .	103
11.9	Histograms showing the frequency amount of point-to-point calculations as a function of deviation between the manual and the model based segmentations. (a) and (b) show the frequency distribution for #1 for image registration and graph cut, respectively, while (c) and (d) similarly show the distribution for #7. All histograms are shown in a cropped version to facilitate a better visual interpretation. . . . .	104
11.10	Segmentation boundary for the manual (red line) and the graph cut based segmentation (green line) for data #7, found with a neighborliness of 26 and a smoothness factor of 9. . . . .	105
11.11	Surface rendering illustrating the difference between the manually segmented mask (red) and the graph cut segmented mask (blue) for #7 with the new setup. . . . .	106

- 11.12 Illustration of the extent and location of the voxel deviation for dataset #7 with the new graph cut setup, given as the closest point differences found between the manual and model based segmentation, superimposed on a 3D rendering of the manual segmentation. The extent of the deviations are shown in millimeter, where a green/blue color indicates a model based segmentation lying on the outside, whereas a yellow/red color indicates a model segmentation located on the inside of the manual segmentation. Furthermore, it should be noted that deviation above -5 mm and below -5 mm are truncated to 5 mm and -5 mm, respectively. . . . 107
- 11.13 Voxel deviations in percent for all datasets processed with image registration and graph cut, represented with green and blue bars, respectively. The red line defines the maximum manual segmentation error found at 1.74 % based on the small study in Section 11.1. . . . . 108
- 11.14 Scatter plot and curve showing the intracranial volume. The black curve is based on 18 data points from [41] and is produced by use of the Lowess smoothing technique, with a 50 % smoothing window and three iterations. The volume estimations from the manual, image registration and graph cut segmentations are represented with a red circle, green triangle and a blue square, respectively. The vertical black lines connect the three results associated with the same dataset, and are only present to ease the interpretation. For the dataset with a value above 1200 cm<sup>3</sup> it should be noted that no value exist for the image registration model, as this dataset figures as a template. The image registration value is set equal to the manually segmented volume. . . . . 110
- 11.15 Visualization of the correspondence in estimated volumes between the two model based segmentations. The dashed line illustrates the mid-line *i.e.* equal estimation of the volume. . . . . 110
- A.1 Test of the influence of the smoothness factor (the inter column weight) and the number of connection in the neighborhood for seven datasets. The performance is defined by means of the voxel deviation in percentages compared to the manual segmentation. Note that the graphs are zoomed version. . . . . 126

---

A.2	Test of the influence of the smoothness factor (the inter column weight) and the number of connection in the neighborhood for dataset #14. The performance is defined by means of the voxel deviation in percentages compared to the manual segmentation. Note that the graphs are zoomed version. . . . .	127
B.1	Error maps illustrating the extent and location of the voxel deviation for dataset #2. (a) and (b) represent the results from the image registration model, whereas (c) and (d) hold the results from the graph cut model. . . . .	130
B.2	Histograms showing the frequency of point-to-point distances [mm] between the manual and a model based segmentation. (a) holds the image registration result, and (b) the graph cut result, for dataset #2 . . . . .	131
B.3	Segmentation boundaries for the manual (red line) and the model based segmentation (green line). (a) and (b) presents the boundary obtained from the image registration, while (c) and (d) holds the boundary for the graph cut, for data #2 . . . . .	131
B.4	Error maps illustrating the extent and location of the voxel deviation for dataset #3. (a) and (b) represent the results from the image registration model, whereas (c) and (d) hold the results from the graph cut model. . . . .	132
B.5	Histograms showing the frequency of point-to-point distances [mm] between the manual and a model based segmentation. (a) holds the image registration result, and (b) the graph cut result, for dataset #3 . . . . .	133
B.6	Segmentation boundaries for the manual (red line) and the model based segmentation (green line). (a) and (b) presents the boundary obtained from the image registration, while (c) and (d) holds the boundary for the graph cut, for data #3 . . . . .	133
B.7	Error maps illustrating the extent and location of the voxel deviation for dataset #4. (a) and (b) represent the results from the image registration model, whereas (c) and (d) hold the results from the graph cut model. . . . .	134

B.8	Histograms showing the frequency of point-to-point distances [mm] between the manual and a model based segmentation. (a) holds the image registration result, and (b) the graph cut result, for dataset #4 . . . . .	135
B.9	Segmentation boundaries for the manual (red line) and the model based segmentation (green line). (a) and (b) presents the boundary obtained from the image registration, while (c) and (d) holds the boundary for the graph cut, for data #4 . . . . .	135
B.10	Error maps illustrating the extent and location of the voxel deviation for dataset #5. (a) and (b) represent the results from the image registration model, whereas (c) and (d) hold the results from the graph cut model. . . . .	136
B.11	Histograms showing the frequency of point-to-point distances [mm] between the manual and a model based segmentation. (a) holds the image registration result, and (b) the graph cut result, for dataset #5 . . . . .	137
B.12	Segmentation boundaries for the manual (red line) and the model based segmentation (green line). (a) and (b) presents the boundary obtained from the image registration, while (c) and (d) holds the boundary for the graph cut, for data #5 . . . . .	137
B.13	Error maps illustrating the extent and location of the voxel deviation for dataset #6. (a) and (b) represent the results from the image registration model, whereas (c) and (d) hold the results from the graph cut model. . . . .	138
B.14	Histograms showing the frequency of point-to-point distances [mm] between the manual and a model based segmentation. (a) holds the image registration result, and (b) the graph cut result, for dataset #6 . . . . .	139
B.15	Segmentation boundaries for the manual (red line) and the model based segmentation (green line). (a) and (b) presents the boundary obtained from the image registration, while (c) and (d) holds the boundary for the graph cut, for data #6 . . . . .	139
B.16	Error maps illustrating the extent and location of the voxel deviation for dataset #8. (a) and (b) represent the results from the image registration model, whereas (c) and (d) hold the results from the graph cut model. . . . .	140

B.17 Histograms showing the frequency of point-to-point distances [mm] between the manual and a model based segmentation. (a) holds the image registration result, and (b) the graph cut result, for dataset #8 . . . . .	141
B.18 Segmentation boundaries for the manual (red line) and the model based segmentation (green line). (a) and (b) presents the boundary obtained from the image registration, while (c) and (d) holds the boundary for the graph cut, for data #8 . . . . .	141
B.19 Error maps illustrating the extent and location of the voxel deviation for dataset #9. (a) and (b) represent the results from the image registration model, whereas (c) and (d) hold the results from the graph cut model. . . . .	142
B.20 Histograms showing the frequency of point-to-point distances [mm] between the manual and a model based segmentation. (a) holds the image registration result, and (b) the graph cut result, for dataset #9 . . . . .	143
B.21 Segmentation boundaries for the manual (red line) and the model based segmentation (green line). (a) and (b) presents the boundary obtained from the image registration, while (c) and (d) holds the boundary for the graph cut, for data #9 . . . . .	143
B.22 Error maps illustrating the extent and location of the voxel deviation for dataset #10. (a) and (b) represent the results from the image registration model, whereas (c) and (d) hold the results from the graph cut model. . . . .	144
B.23 Histograms showing the frequency of point-to-point distances [mm] between the manual and a model based segmentation. (a) holds the image registration result, and (b) the graph cut result, for dataset #10 . . . . .	145
B.24 Segmentation boundaries for the manual (red line) and the model based segmentation (green line). (a) and (b) presents the boundary obtained from the image registration, while (c) and (d) holds the boundary for the graph cut, for data #10 . . . . .	145
B.25 Error maps illustrating the extent and location of the voxel deviation for dataset #11. (a) and (b) represent the results from the image registration model, whereas (c) and (d) hold the results from the graph cut model. . . . .	146

B.26 Histograms showing the frequency of point-to-point distances [mm] between the manual and a model based segmentation. (a) holds the image registration result, and (b) the graph cut result, for dataset #11 . . . . .	147
B.27 Segmentation boundaries for the manual (red line) and the model based segmentation (green line). (a) and (b) presents the boundary obtained from the image registration, while (c) and (d) holds the boundary for the graph cut, for data #11 . . . . .	147
B.28 Error maps illustrating the extent and location of the voxel deviation for dataset #12. (a) and (b) represent the results from the image registration model, whereas (c) and (d) hold the results from the graph cut model. . . . .	148
B.29 Histograms showing the frequency of point-to-point distances [mm] between the manual and a model based segmentation. (a) holds the image registration result, and (b) the graph cut result, for dataset #12 . . . . .	149
B.30 Segmentation boundaries for the manual (red line) and the model based segmentation (green line). (a) and (b) presents the boundary obtained from the image registration, while (c) and (d) holds the boundary for the graph cut, for data #12 . . . . .	149
B.31 Error maps illustrating the extent and location of the voxel deviation for dataset #13. (a) and (b) represent the results from the image registration model, whereas (c) and (d) hold the results from the graph cut model. . . . .	150
B.32 Histograms showing the frequency of point-to-point distances [mm] between the manual and a model based segmentation. (a) holds the image registration result, and (b) the graph cut result, for dataset #13 . . . . .	151
B.33 Segmentation boundaries for the manual (red line) and the model based segmentation (green line). (a) and (b) presents the boundary obtained from the image registration, while (c) and (d) holds the boundary for the graph cut, for data #13 . . . . .	151
B.34 Error maps illustrating the extent and location of the voxel deviation for dataset #14. (a) and (b) represent the results from the graph cut model. . . . .	152

- 
- B.35 Histograms showing the frequency of point-to-point distances [mm] between the manual and a model based segmentation. (a) holds the graph result, for dataset #14 . . . . . 153
- B.36 Segmentation boundaries for the manual (red line) and the model based segmentation (green line). (a) and (b) presents the boundary obtained from boundary for the graph cut, for data #14 . . . 153
- B.37 Error maps illustrating the extent and location of the voxel deviation for dataset #15. (a) and (b) represent the results from the image registration model, whereas (c) and (d) hold the results from the graph cut model. . . . . 154
- B.38 Histograms showing the frequency of point-to-point distances [mm] between the manual and a model based segmentation. (a) holds the image registration result, and (b) the graph cut result, for dataset #15 . . . . . 155
- B.39 Segmentation boundaries for the manual (red line) and the model based segmentation (green line). (a) and (b) presents the boundary obtained from the image registration, while (c) and (d) holds the boundary for the graph cut, for data #15 . . . . . 155



# List of Tables

---

2.1	Classification of the four most common forms of non-syndromic craniosynostosis, based on [22, 38]. . . . .	10
3.1	Pros and cons for ICV estimation using MR. . . . .	19
3.2	Pros and cons for ICV estimation using CT . . . . .	20
4.1	Data information, including place of origin and patient information. . . . .	24
4.2	Data and image information. . . . .	24
4.3	Definition of the four levels defining the image pyramid. $L_0$ gives the image with its original size and resolution, followed by the coarser levels. For each downsampled level, the size is reduced, which entails in larger voxel dimensions. $m^3$ defines the slice number, which varies between the datasets. . . . .	27
5.1	User chosen intensity thresholds for all datasets. . . . .	31
8.1	Example of the directions principal axes found in two 3D images. . . . .	66

---

9.1	Mean time consumption for the five level situations when applying the affine transformation. . . . .	74
9.2	Mean time consumption for the five level situations run with the B-spline transformation. . . . .	76
11.1	Volume estimations from two individual segmentations and their appurtenant voxel deviation in percentage. . . . .	94
11.2	Result overview, listing the volume estimations from the manual and the two model based segmentations. The voxel deviation between the manual and the two model based segmentations are furthermore presented. . . . .	97
11.3	Result overview, listing the volume estimations from the graph cut segmentation with a parameter setup consisting of a 26 neighborliness and a smoothness factor of 9. . . . .	106
11.4	Summary of the performance of the two models, image registration (IR) and graph cut (GC), expressed in mean processing time and voxel deviation, respectively. Results for both graph cut setups are presented, where the parameters are given by smoothness factor/neighborliness. . . . .	107

# Bibliography

---

- [1] Line search methods (chapter 3). [urlhttp://pages.cs.wisc.edu/feris/cs730/chap3.pdf](http://pages.cs.wisc.edu/feris/cs730/chap3.pdf).
- [2] What is craniosynostosis? <http://www.kidsgrowth.com/resources/articledetail.cfm?id=2398>.
- [3] Odontology 3D Laboratory. Standardization of head orientation according to the frankfort horizontal and midsagittal plane. <http://www.lab3d.odont.ku.dk/landmarker/>.
- [4] P. J. Anderson, D. J. Netherway, K. McGlaughlin, and D. J. David. Intracranial volume measurement of sagittal craniosynostosis. *Journal of Clinical Neuroscience*, 14(5):455–458, 2007.
- [5] AquaVeo. Gms:3d grid types. [http://www.xmswiki.com/xms/GMS:3D\\_Grid\\_Types](http://www.xmswiki.com/xms/GMS:3D_Grid_Types).
- [6] J. Ashburner and K. J. Friston. Unified segmentation. *NeuroImage*, 26(3):839–851, 2005.
- [7] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2134:359–374, 2001.
- [8] Amanda Moffitt A.A Kane Jeffrey L Marsh Jayesh Panchal Joan T Richtsmeier C. A Hill, S Vaddi and Kristina Aldridge. Intracranial volume and whole brain volume in infants with unicoronal craniosynostosis. *The Cleft Palate-Craniofacial Journal*, 48(4):394–398, 2011.

- [9] T.H. Cormen. *Introduction to algorithms*. The MIT press, 2001.
- [10] T.A. Darvann. *Methods for Measurement and Analysis of Craniofacial Morphology and Growth in Children with Cleft Lip and Palate*. PhD thesis, Technical University of Denmark, 2003.
- [11] T.A. Darvann. A vtk-based tool for landmarking of polygonal surfaces. in: Takada k, kreiborg s. (eds) *in silico dentistry - the evolution of computational oral health science*. *Medigit, Osaka, Japan*, 2008.
- [12] A.S. Dekaban. Table of cranial and orbital measurements, cranial volume, and derived indexes in male and females from 7 days to 20 years of age. *Annals of Neurology*, 2(6):485–491, 1977.
- [13] Neurosurgery Department. Craniosynostosis. <http://www.neurosurgery.ufl.edu/patients/pediatric-craniosynostosis.shtml>.
- [14] E-Anatomy. Planes and motions used in anatomy. [urlhttp://www.imaios.com/en/e-Anatomy/Limbs/Planes-and-motions-diagrams](http://www.imaios.com/en/e-Anatomy/Limbs/Planes-and-motions-diagrams).
- [15] J.M Fitzpatrick, D.L.G Hill, and C.R Maurer Jr. Image registration (chapter 8). In *Handbook of Medical Imaging*, pages 447–513.
- [16] D. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, 2010 (1962).
- [17] K. Lin F.R. Johns, J.A. Jane Sr. Syndromic craniosynostosis. In *Neurosurgery: Principles and Practice*, pages 461–473. Springer London, 2005.
- [18] D.M. Greig, B.T. Porteous, and A.H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 271–279, 1989.
- [19] B.K. Hall. Sutures and craniosynostosis. In *Bones and Cartilage - Development and Evolutionary Skeletal Biology*, pages 429–439. Elsevier, 2005.
- [20] R Hauser. Line search methods for unconstrained optimization (lecture slides), 2007.
- [21] The Urban Child Institute. Memphis' education funding misses best chance for impact. <http://www.theurbanchildinstitute.org/articles/perceptions/memphis-education-funding-misses-best-chance-for-impact>, April 2011.
- [22] K.Y.K. Lin J.A. Jane Jr, A.S. Dumont and J.A. Jane Sr. Craniosynostosis. In *Neurosurgery: Principles and Practice*, pages 445–461. Springer London, 2005.

- [23] Wyler A.R. Jane, J.A. Surgery for craniosynostosis. <http://www.emedicine.medscape.com/article/1281182-verview#aw2aab6b5>, April 2010.
- [24] J. Jensen and C.B. Anker. Segmentation of abdominal adipose tissue in mri in a clinical study of growth and diet. Bsc. Thesis - IMM, June 2011.
- [25] Gold JA. Hoffman T.L. Panchal J. Boyadjiev S.A. Kimonis, V. Genetics of craniosynostosis. *Seminars in Pediatric Neurology*, 14(3):150–161, 2007.
- [26] H. Ólafsdóttir, T. A. Darvann, N. V. Hermann, E. Oubel, B. K. Ersbøll, A. F. Frangi, P. Larsen, C. A. Perlyn, G. M. Morriss-Kay, and S. Kreiborg. Computational mouse atlases and their application to automatic assessment of craniofacial dysmorphology caused by the crouzon mutation fgfr2c342y. *Journal of Anatomy*, 211(1):37–52, 2007.
- [27] R Larsen. Medical image analysis, course note, 2008.
- [28] K. Madsen and H.B Nielsen. Introduction to optimization and data fitting. 2008.
- [29] K.Y. Manjunath. Estimation of cranial volume-an overview of methodologies. *Jour. Anat. Soc. India*, 51(1):85–91, 2005.
- [30] Warren S.M. Bernstein J. Kreiborg S. et. al McCarthy, J.G. Parameters of care for craniosynostosis. *Cleft Palate?Craniofacial Journal*, 49(1), 2012.
- [31] J Modersitzki. *Numerical Methods for Image Registration*. Oxford Science Publications, 2004.
- [32] J Modersitzki. Mathematics meets medicine: An optimal alignment. url-<http://www.siam.org/meetings/op08/Modersitzki.pdf>, 2008.
- [33] J Modersitzki. *FAIR Flexible Algorithms for Image Registration*. siam Society for Industrial and Applied Mathematics, 2009.
- [34] Wilkie A.O.M. Morriss-Kay, G.M. Growth of the normal skull vault and its alteration in craniosynostosis: insights from human genetics and experimental studies. *Jour. Anat.*, 207:637–653, 2005.
- [35] Clayman M Seagle M.B. White S. Murad, G.J.A. Endoscopic-assisted repair of craniosynostosis: Results, 2006.
- [36] G. Pengas, J. M. S. Pereira, G. B. Williams, and P. J. Nestor. Comparative reliability of total intracranial volume estimation methods and the influence of atrophy in a longitudinal semantic dementia cohort. *Journal of Neuroimaging*, 19(1):37–46, 2009.

- [37] J. Petersen, M. Nielsen, P. Lo, Z. Saghir, A. Dirksen, and M. de Bruijne. Optimal graph based segmentation using flow lines with application to airway wall segmentation. In *Information Processing in Medical Imaging*, volume 6801 of *Lecture Notes in Computer Science*, pages 49–60. Springer Berlin / Heidelberg, 2011.
- [38] S. Podda. Craniosynostosis management. <http://www.emedicine.medscape.com/article/1281182-verview#aw2aab6b5>, Sep 2008.
- [39] Stephens T.D. Tate P. Seeley, R.R. Skeletal system: Bones and bone tissue. In *Anatomy and Physiology*, pages 167–196. McGraw-Hill Higher Education, 2006.
- [40] S. Sgouros. Skull vault growth in craniosynostosis. *Childs nerv. Syst.*, 21:861–870, 2005.
- [41] S. Sgouros, A.D. Hockley, J.H. Goldin, M.J.C. Wake, and K Natarajan. Intracranial volume change in childhood. *Jour. Neurosurg*, 91:610–616, 1999.
- [42] S. Sgouros, A.D. Hockley, J.H. Goldin, M.J.C. Wake, and K Natarajan. Intracranial volume change in craniosynostosis. *Jour. Neurosurg*, 91:617–625, 1999.
- [43] S. M. Smith. Fast robust automated brain extraction. *Human Brain Mapping*, 17(3):143–155, 2002.
- [44] M Vester-Christensen. *Image Registration and Optimization in the Virtual Slaughterhouse*. PhD thesis, IMM, DTU, 2008.