Registration and Simulation for the Analysis of Intraoperative Brain Shift

---

Registrierung und Simulation zur Analyse von intraoperativen Gehirndeformationen

Der Technischen Fakultät der
Universität Erlangen-Nürnberg

zur Erlangung des Grades

DOKTOR-INGENIEUR

vorgelegt von

Grzegorz Soza

Erlangen – 2005

Als Dissertation genehmigt von
der Technischen Fakultät der
Universität Erlangen-Nürnberg

Tag der Einreichung:	20. Juni 2005
Tag der Promotion:	27. Juli 2005
Dekan:	Prof. Dr. A. Winnacker
Berichterstatter:	Prof. Dr. G. Greiner
	Prof. Dr. T. Ertl

Abstract

In recent years, the quality of surgical procedures has rapidly advanced in a large part due to the development of navigation systems. In particular, neurosurgical interventions have enormously benefited from the possibility of being able to track surgical tools simultaneously in the physical space associated with the operating room and in the virtual space related to volumetric images giving insight into patient's body. Navigation systems combined with intraoperative imaging allow precise removal of diseased tissue, thus reducing postoperative traumatological deficits. At the same time, there is an immense demand for new methods establishing a correspondence between time-shifted data such as pre- and intraoperative images. This would further improve surgical interventions by a more extensive integration of preoperative planning data into the operating environment.

In this doctoral thesis different algorithms for registration of medical tomographic images are presented and implemented. The introduced methods focus on application within an intraoperative environment. In neurosurgery restrictive time requirements are an extremely important issue. Thus, the presented methods focus on acceleration techniques and at the same time on a high accuracy and reliability. In this respect, a rigid registration strategy exploiting graphics hardware in order to accelerate the interpolation operations has been optimized and extended by a hierarchical technique. This implementation can be used for an initial alignment of pre- and intraoperative data.

However, the deformation of soft brain tissue occurring after craniotomy (referred to as *brain shift*) necessitates non-linear data registration. This has been achieved with a volumetric image deformation based on a flexible hexahedral model. This model is designed to allow acceleration of the registration with graphics hardware. In order to improve the quality of the algorithm, an extension has been proposed where the free-form deformation (FFD) concept is applied to conduct a non-linear registration. The developed system has proved its value in experiments with real clinical data acquired before and during open-skull surgery. It has also been successfully applied to correct the distortions in diffusion tensor image (DTI) data, thus showing the techniques flexibility and strength.

Furthermore, this thesis introduces a fundamental simulation framework for physically-based modeling of the intraoperative brain deformation. In this concept the brain is described as a linearly elastic medium saturated with a viscous fluid.

Implementing this system, prediction of the extent and the direction of the brain shift is possible, thus enabling a better surgery planning. Here, the important problem is establishing the correct elastic parameters for the brain tissue. This has been addressed with a novel approach combining simulation and registration. In an automatic optimization procedure the intraoperative data is compared to the image reconstructed after simulation, thus finding the best elastic constants. Furthermore, a parallel implementation of the simulation model has led to a significant reduction in computation times. Additionally, huge data structures which do not fit in the memory of a single-processor machine can also be processed with this approach.

Contents

Abstract iii

Table of Contents ix

List of Figures x

List of Tables xi

Listings xiii

Acknowledgements xv

I Introduction 1

1 Introduction 3
1.1 Contribution 5
1.2 Outline of the Work 8

II Registration 11

2 Survey of Image Registration 13
2.1 Mathematical Background 15
2.2 Applications 16
2.3 Rigid Registration 18
2.3.1 Related Work 19
2.4 Non-Rigid Registration 23
2.4.1 Related Work 24

3 Hardware-based Rigid Registration 29
3.1 Basics on Graphics Hardware 30
3.2 Similarity measures 33

3.3 Registration . . . 35
3.3.1 3D Textures . . . 36
3.3.2 Reslicing Algorithm . . . 36
3.3.3 Multiresolution Pyramid . . . 39
3.4 Results . . . 42
3.5 Discussion . . . 46
3.6 Conclusion . . . 49

4 Hardware-based Non-Rigid Registration 51
4.1 Previous Work . . . 52
4.2 3D Piecewise Linear Deformation Model . . . 53
4.2.1 Deformation in Texture Space . . . 53
4.2.2 Deformation in Geometry Space . . . 55
4.2.3 Hierarchical Octree Structure . . . 55
4.2.4 Implementation Issues . . . 57
4.2.5 Registration Algorithm . . . 58
4.2.6 Optimization . . . 60
4.2.7 Multiresolution approach . . . 61
4.2.8 Results . . . 61
4.2.9 Discussion . . . 67
4.3 3D Free-Form Deformation . . . 68
4.3.1 Deformation Model . . . 68
4.3.2 Hardware-Accelerated Free-Form Deformation . . . 69
4.3.3 Registration . . . 70
4.3.4 Results . . . 72
4.3.5 Discussion . . . 74
4.4 Conclusion . . . 76

III Simulation 79

5 Survey of Soft Tissue Simulation 81
5.1 Clinical Applications . . . 82
5.2 Survey of Brain Shift Simulation . . . 84
5.2.1 Linear Elasticity Models . . . 85
5.2.2 Viscoelastic Models . . . 86
5.2.3 Biphasic Theory and Poroelastic Models . . . 87
5.2.4 Conclusion . . . 87

6 Computational Model of Brain Shift 89
6.1 Finite Element Theory . . . 90
6.2 Governing Equations . . . 93
6.2.1 Stress Distribution . . . 93

6.2.2 Darcy's Law . . . 96
6.2.3 Boundary Conditions . . . 97
6.3 Finite Element Discretization . . . 98
6.4 Geometry Generation . . . 99
6.4.1 Extraction of Brain Volume from MR Data . . . 99
6.4.2 Triangular Surface Extraction . . . 99
6.4.3 Tetrahedrization . . . 102
6.5 Boundary Conditions . . . 104
6.6 Volume Reconstruction . . . 104
6.7 Results . . . 106
6.7.1 Synthetic Dataset . . . 106
6.7.2 Clinical experiments . . . 107
6.8 Discussion . . . 110
6.9 Conclusion . . . 112

7 The Mechanical Properties of the Brain 113
7.1 Previous Work . . . 114
7.2 Method . . . 115
7.2.1 Similarity Function . . . 115
7.2.2 Biomechanical Model and Volume Reconstruction . . . 116
7.2.3 Minimization Framework . . . 116
7.3 Results . . . 116
7.4 Discussion . . . 118
7.5 Conclusion . . . 120

8 High Performance Simulation 121
8.1 Previous Work . . . 122
8.2 Parallel Framework . . . 123
8.2.1 Grid Partitioning . . . 123
8.2.2 Element Distribution and Management . . . 125
8.2.3 Matrix Storage . . . 125
8.2.4 Solving the Linear Equation System . . . 127
8.3 Results . . . 127
8.4 Discussion . . . 129
8.5 Conclusion . . . 132

IV Application of the Non-linear Registration 133

9 Non-linear Correction of DTI Data 135
9.1 Introduction . . . 135
9.2 Algorithm . . . 137
9.2.1 Image Acquisition . . . 137

9.2.2 Fiber Tracking . . . 138
9.2.3 Registration . . . 139
9.3 Results . . . 140
9.4 Discussion . . . 141
9.5 Conclusion . . . 142

V Conclusion **143**

10 Summary **145**
10.1 Future Challenges . . . 147

VI Appendix **149**

A Finite Element Discretization **151**
A.1 Weighted Residuals Form . . . 151
A.2 Discretization . . . 153

Bibliography **157**

VII German Part **175**

Kurzfassung **177**

Inhaltsverzeichnis **182**

Einleitung **183**
Wissenschaftliche Beiträge . . . 185
Gliederung der Arbeit . . . 189

Zusammenfassung **191**
Zukünftige Herausforderungen . . . 193

List of Figures

1.1 Brain shift phenomenon 5
1.2 Information flow in computer-assisted surgery 6

2.1 Registration and fusion 13
2.2 Registration steps 14
2.3 2D image interpolation 16
2.4 2D example of rigid and affine transformations 19

3.1 Development of PC graphics chips 30
3.2 The graphics pipeline 32
3.3 Rendering based on 3D textures 38
3.4 Reslicing of the source volume for the calculation of the joint histogram 40
3.5 Multiresolution pyramid for registration 41
3.6 Overlayed head images before rigid registration 42
3.7 Optimization of the normalized mutual information during a rigid registration procedure 43
3.8 Overlayed images after rigid registration 47

4.1 The geometry and the texture space of piecewise linear deformation model 54
4.2 Texture mapping artifacts for hierarchical octree structure 56
4.3 Constraints on edges and on faces 57
4.4 Texture mapping for the octree geometry 58
4.5 Explicit tessellation of polygons for the correction of texture mapping . 59
4.6 Comparison of the SPSA nd Powell optimization runs 62
4.7 Pre- and intraoperative brain images after piecewise-linear registration 65
4.8 Difference images after rigid and after piecewise linear registration . . 66
4.9 Bernstein polynomials of order 4 69
4.10 Subdivision of a slice into 2D patches 71
4.11 3d free-form deformation 72
4.12 Results of the FFD-based non-linear registration 75
4.13 Example of registration artifacts 76

6.1 Example of finite element discretization in 2D 92

6.2 Matrix and vector assembly . . . 93
6.3 Stresses acting upon the faces of an elementary volume . . . 95
6.4 Conservation of the fluid volume . . . 97
6.5 The brain segmentation process . . . 100
6.6 Projection onto an isosurface after refinement . . . 102
6.7 Extraction of a semi-regular mesh of the brain . . . 103
6.8 Anatomical details of the brain surface . . . 104
6.9 Histograms showing the quality of the generated tetrahedral grids . . . 105
6.10 The region of the boundary condition . . . 105
6.11 2D example of image reconstruction techniques . . . 106
6.12 Deformation of synthetic data . . . 108
6.13 Two-dimensional visualization of the simulation results . . . 110
6.14 Visualization of the simulation results . . . 111

7.1 Plots of the similarity measure with one coordinate held constant . . . 118
7.2 Color-coded visualization of the values of the similarity measure sampled on a two-dimensional domain . . . 119

8.1 An exemplary triangle mesh partitioned over three processors and the corresponding local-to-global mapping . . . 126
8.2 Parallel matrix and vector distribution . . . 126
8.3 Brain geometry partitioned over eight processors . . . 128
8.4 Timing results of the parallel implementation . . . 130

9.1 Artifacts typical of DTI data . . . 137
9.2 The left pyramidal tract extracted from DTI data . . . 139
9.3 Non-linear correction of DTI images . . . 140
9.4 Distortion of the left pyramidal tract . . . 141

G.1 Brain Shift Phänomen . . . 185
G.2 Der Informationsfluss in computergestützter Chirurgie . . . 186

List of Tables

3.1 Computation times of the rigid registration algorithm 44
3.2 Distances before rigid registration 45
3.3 Point errors after rigid registration 46

4.1 Computation data of the 3D piecewise linear non-rigid registration algorithm 63
4.2 Comparison of the performance of the SPSA and the Powell algorithm for non-linear registration 63
4.3 Point errors before piecewise linear non-rigid registration 64
4.4 Point errors after piecewise linear non-rigid registration 66
4.5 Computation data from experiments with the non-linear registration algorithm using 3D Bézier functions 73
4.6 Point errors after registration with the use of Bézier functions 74

6.1 Point errors after biomechanical simulation 109

7.1 The optimal values of the elastic parameters 117
7.2 A statistical analysis of the calculated elastic moduli 117

8.1 Architecture of the computational cluster 128
8.2 Acceleration in the computation times for the parallel framework . . . 129

9.1 Distortions of the left pyramidal tract 141

Listings

3.1 OpenGL setup for initialization of 3D textures 37
3.2 OpenGL setup for setting geometry-texture correspondence 38
8.1 Sample output of the mesh partitioning program 124

Acknowledgements

First of all, I want to express my gratitude to my supervisor Prof. Dr. Günther Greiner for his scientific support in all stages of this thesis. This collaboration invaluably contributed to the success of this work. Additionally, he deserves my great personal respect for his exceptional way of conducting the Dept. of Computer Graphics and contacting staff members.

Furthermore, I would like to thank Prof. Dr. Thomas Ertl from the Institute of Visualization and Interactive Systems in Stuttgart, who kindly agreed to review this thesis. A very special acknowledgment is due Prof. Dr. Rudolf Fahlbusch from the Dept. of Neurosurgery in Erlangen for his open-mindedness with respect to the computer scientists and for being a member of the examination board.

I am enormously grateful to Dr. Peter Hastreiter from Neurocenter at the University of Erlangen for the close scientific assistance during my entire PhD and for proof reading a large part of this manuscript. In particular, I am beholden to him for helping me through many difficulties. I am also much obliged to PD Christopher Nimsky for introducing me into medical environment and for providing clinical data.

Dr. Roberto Grosse deserves special gratitude for helping me to step into the finite element method. I am also very thankful to Stefan Zachow from Zuse Institut in Berlin for the cooperation in grid generation. Further, I want to thank the other colleagues, especially Maria Baroti, Frank Enders, Dr. Ulf Labsik, Dorit Merhof, Stefan Meinlschmidt, Frank Reck, Dr. Christof Rezk-Salama, Dr. Michael Scheuering, Prof. Dr. Marc Stamminger, Fernando Vega-Higuera, Christian Vogelgsang and my office-mates Horst Hadler, Prof. Dr. Kai Hormann and Dr. Salvatore Spinello.

In addition, I am very obliged to my students Ezgi Bingöl, Christian Eisenacher, Matthias Teßmann, Sylwia Urzęńska and Yali Yuan. I would like to especially thank Michael Bauer for his invaluable contribution into the registration framework.

Furthermore, I am inestimably indebted to my friend Joel Heersink for proofing many of my publications as well as this thesis.

My deep respect and gratitude are due to all my friends and my family, especially to my mother for giving my an exceptional childhood and for supporting my career.

Finally, and above all I want to express my deepest love to my fiancée Katarzyna Słodowska for unremittingly giving me her unconditional love and for supporting and motivating me thorough my work.

Grzegorz Soza

Part I

Introduction

Chapter 1

Introduction

The era of medical imaging started with the discovery of X-rays at the end of the 19th century. The first image-guided surgery was accomplished by Prof. Cox from the Mc Gill University in Montreal only six weeks after Wilhelm Röntgen had published a preliminary report on his X-ray discovery [140]. Prof. Cox made a radiograph of the leg of a young patient that had been shot, exposing a bullet that had not been found in previous surgical explorations. The X-ray images were then manually aligned to the position of the patient, which guided the surgeon to the successful removal of the bullet. This surgery was the first application of medical image registration.

Since this landmark operation the technology used for image-guided surgeries has been continuously improving. The advent of X-ray computer tomography (CT), ultrasound (US), positron emission tomography (PET) and magnetic resonance imaging (MRI) have revolutionized traditional surgical techniques, providing high-quality three-dimensional (3D) images giving insight into the body of the patient. These images can be manipulated and adapted by a computer software to transfer additional diagnostic information into the operating room. Further, the invention of navigation hardware brought the possibility of tracking surgical tools during the intervention, thus enabling simultaneous navigation both in the physical space associated with the operating room and in a virtual image space. Especially in recent years, a very rapid development of navigation techniques has been observed [145]. These advances are due to the enormous progress in computer processor and graphics hardware technology which allows manipulation and real-time visualization of 3D images.

Nowadays, image-guidance is used for diagnosis, therapy planning and surgery in many diseases such as breast biopsy and tumor resection, prostate cryosurgery[1] and brachytherapy[2], craniofacial corrections, spinal procedures including radiother-

[1] Cryosurgery is a method of super-freezing tissue (e.g. cancerous tumors) to destroy it.
[2] Radiation therapy given at a short distance.

apy, posttherapeutic monitoring of carotid atherosclerosis[3], renal[4] tumors ablation and in liver surgery planning.

The area of medicine that has probably most benefited from image-guidance techniques is neurosurgery. On the one hand, neurosurgery requires highest precision in planning the access to the target area and the optimal position of medical instruments. On the other hand, accuracy is essential in preventing damage to critical structures. It is also crucial for the resection of diseased tissue, since it is often visually indistinguishable from the surrounding healthy structures.

Current commercially available neuronavigation systems heavily rely on the assumption that the preoperatively acquired images remain valid during surgical interventions. However, soft tissue can move which results in changes between the pre- and intraoperative images. During an open cranial surgery, the brain tissue can significantly deform (see Figure 1.1). This phenomenon is known as *brain shift* and is a result of neurosurgical events and a variety of physical actions such as the influence of gravity, the change of the pressure acting upon the brain or leakage of cerebrospinal fluid (CSF) [102, 122].

There are non-commercial approaches which deal with this issue using complex mathematical and numerical techniques. However, the requirements of the intraoperative neurosurgical environment are very restrictive both with respect to the precision of algorithms and the time available for computations. Therefore, there is an immense need for better and faster algorithms for an application in clinical practice.

In Figure 1.2, the overall information flow of a computer-assisted approach is presented supporting the process of surgery. Thereby, pre- and intraoperative image data is used for the generation of models and therapy planning. In this respect, a patient-specific model and a surgical plan are developed based on preoperative image data. During a surgery, this information is transferred into the operating environment with the use of navigation system.

However, the created patient-specific model and surgical plan require an intraoperative update. For this purpose, imaging is an option to capture the occurring brain shift. In addition to that, the intraoperative image data serves as a basis for a *registration* strategy which allows the integration of exclusively available preoperative information into a neuronavigation system. Alternatively, if no intraoperative imaging is available, a biomechanical *simulation* can be employed for the prediction of the occurring deformation. Optimally, intraoperative imaging, registration and simulation are combined to provide comprehensive information and to achieve a maximum of accuracy and safety.

[3] Calcification of the internal carotid artery which can lead to a stroke.
[4] The term “renal” refers to the kidney.

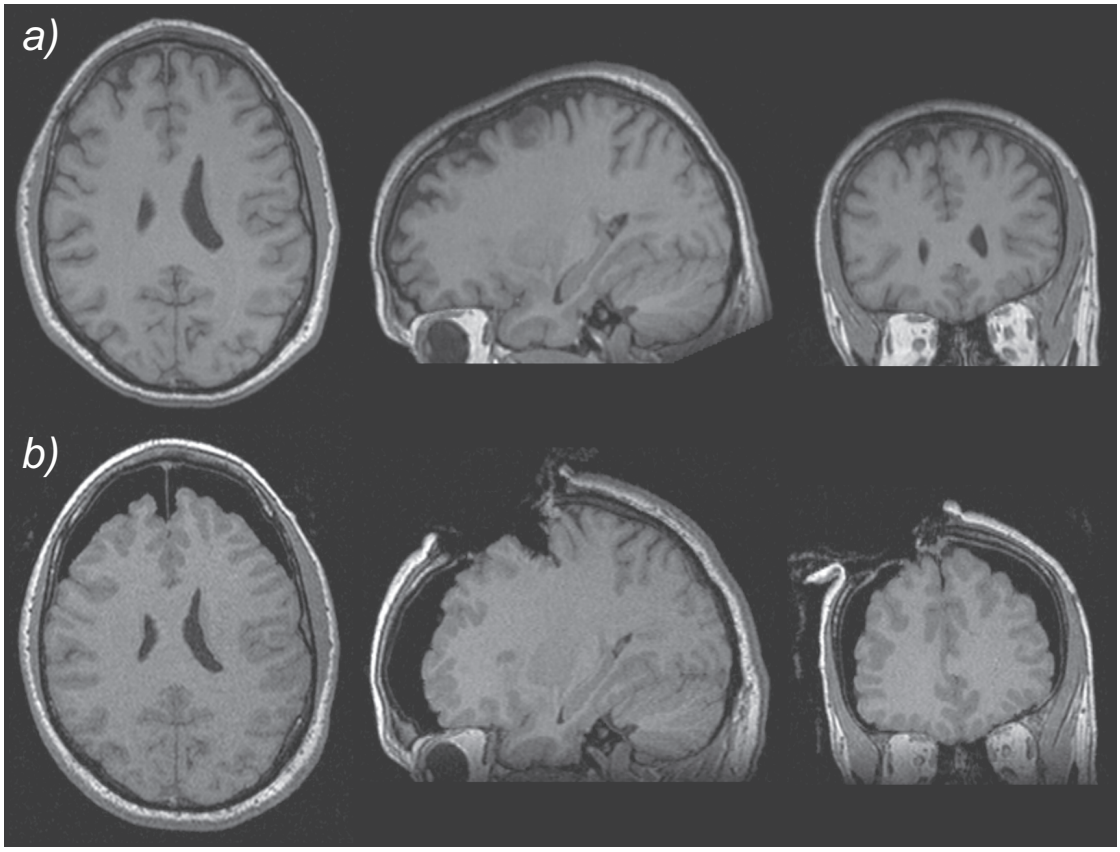

Figure 1.1: Preoperative *a)* and intraoperative *b)* MR images of the head showing the brain shift phenomenon. A substantial relocation of the soft tissue is visible

1.1 Contribution

The main focus of this doctoral thesis is the development of new methods supporting the update of a patient-specific model and of a surgical plan during computer-assisted neurosurgery. In this respect, algorithms compensating for the brain shift phenomenon have been developed. On the one hand, linear and non-linear image-based registration approaches have been implemented which enable calculating the transformation of preoperative MR images onto intraoperative data acquired after brain shift has occurred. On the other hand, a physically-based simulation framework has been implemented which allows the prediction of the occurring brain deformation. With this system, the extent and the direction of the intraoperative brain movement can be demonstrated, thus supporting intervention planning.

Finally, a data collection has been constructed which consists of intraoperative MR data registered to the corresponding preoperative MR images deformed according to simulation calculations. Statistical information derived from this data collection has been incorporated into the developed simulation framework to optimize

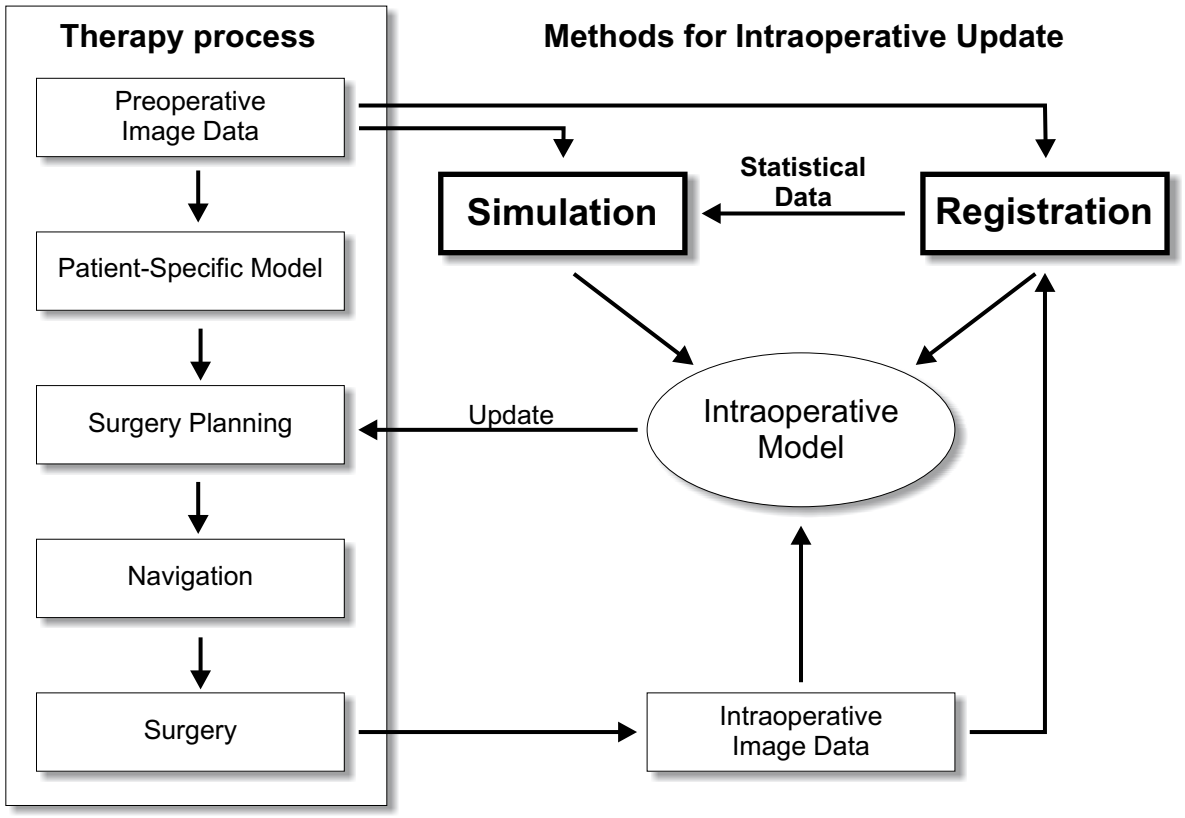

Figure 1.2: Information flow in computer-assisted surgery showing the interaction between a conventional therapy process and advanced methods for intraoperative update of the planning concept

the simulation, thus improving the quality of an intraoperative update. The developed registration and simulation strategies are characterized by considerably reduced calculation times which is an essential prerequisite for intraoperative use in an image-guidance system. This has been achieved with the development of novel techniques exploiting modern PC graphics cards and parallelization strategies.

The presented thesis comprises the following main contributions:

HARDWARE-BASED RIGID REGISTRATION

Rigid registration is the basic step needed for an update of preoperative imagery. The obtained global, linear transformation is an essential prerequisite for a subsequent local, non-linear registration. Additionally, for the statistical method combining registration and simulation, the quality of the rigid alignment is essential for setting appropriate boundary conditions. For these purposes, an approach for the 3D/3D rigid alignment of medical images [72] introduced for specialized hardware systems

has been optimized for standard PC architectures and greatly extended, which resulted in a hierarchical framework allowing automatic registration. To accelerate the computations, trilinear interpolation capabilities of graphics hardware has been extensively used. The implemented system has been presented in [73, 74].

HARDWARE-BASED NON-LINEAR REGISTRATION

Non-linear registration algorithms are characterized by high computational costs. Aiming at clinical applications, acceleration techniques are of great importance. In order to fulfill this requirement, fast and at the same time accurate methods of non-linear registration have been developed.

In this context, non-linear volume deformation is approximated with 3D piecewise linear functions where an object is subdivided into a number of volumetric patches which form can be easily manipulated. This idea has been transferred to graphics hardware where the deformation is modeled by shifting vertices in a 3D texture space. Hardware-based trilinear interpolation significantly accelerates the iterative registration process. The value of this approach has been proved in [164, 77].

As a next step, 3D Bézier functions have been integrated into the developed registration framework further exploiting capabilities of modern graphics cards. This makes the algorithm an outright novel method in the field of registration. It has been introduced in [157, 165].

BIOMECHANICAL MODEL FOR BRAIN SHIFT SIMULATION

The developed framework contains all the steps required for conducting a finite element simulation. Starting with the segmentation of the brain from the MR data, a triangular surface is calculated in a manner which ensures the regularity of the mesh. Subsequently, a tetrahedral mesh is generated. This technique of volume discretization is crucial for the numerical stability of the simulation process and is an important contribution. Further, physical equations describing brain deformation are defined based on the model proposed by Miga *et al.*[107] which has been greatly extended. The necessary boundary conditions are defined according to the information inferred from the rigid registration of pre- and intraoperative MR head scans. The work has been described in [158, 159].

COUPLED SIMULATION AND REGISTRATION FRAMEWORK

To reliably predict the deformation of soft brain tissue, an elastic characterization of the tissue is required. For the simulation it is crucial to reconstruct both Young's modulus E and Poisson's ratio ν since their combination is important for the model. For this purpose, a new method

has been introduced where both elastic moduli are calculated iteratively based on the registration of intraoperative MR data to the corresponding result of a biomechanical simulation. The ideas of this work have been published in [161, 160, 163].

High Performance Implementation of the Simulation

An important requirement for clinical applicability is time efficiency. Therefore, a parallel framework has been constructed as an essential extension to the simulation model. Due to distributed processing of the data, calculations with a very high resolution geometry can be conducted efficiently . Thus, detailed brain structures can be modeled, whilst the response of the system is provided at rates acceptable for use during neurosurgery as described in [162].

Non-linear Correction of Diffusion Tensor Images

The diffusion tensor imaging (DTI) has been introduced recently and provides important information about neurological connections within the brain. However, this data is characterized by large distortions which are a result of the underlying fast MR imaging protocol. In order to correct the artifacts in DTI data, the developed registration approach based on 3D Bézier functions has been applied [105]. Thus, a powerful support aiming at minimizing postoperative neurological deficits is available.

1.2 Outline of the Work

This thesis is composed of five main parts. In the first part, the main concepts of the presented research and the motivation therefor are given. The original contribution of this doctoral thesis is emphasized and the outline of this work is presented.

In the second part, image registration algorithms are addressed. Chapter 2 contains the basic definitions and notions of registration. Various existing linear and non-linear medical image registration approaches are presented and discussed. Chapter 3 begins with an introduction to graphics hardware. Then, the implemented hardware-based rigid 3D/3D registration approach and results of its application to medical images are provided. In Chapter 4 the concept of the invented piecewise linear registration and its implementation using graphics hardware are described. An extension of this approach based on 3D Bézier functions is presented and evaluated.

The third part of this thesis deals with a biomechanical simulation of soft-tissue deformation. Chapter 5 reviews related works on modeling soft tissue deformation. Chapter 6 begins with an introduction to selected finite element simulation techniques, where the fundamental notions are defined and relevant mathematical properties are explained. Then, a detailed description of the constructed framework

for numerical simulation of brain shift is provided. Thereby, various constituent elements of the implemented system are described. Starting with the geometry generation, a numerical solution of the underlying partial differential equations, proper boundary conditions and finally the visualization and interpretation of the results are successively addressed. In Chapter 7, a system for the estimation of the mechanical properties of brain tissue is introduced. In this context, the idea of coupling simulation and registration for the calculation of the elastic moduli of the brain is introduced and the results of the experiments are discussed. Finally, Chapter 8 introduces a parallel framework for conducting high performance simulations Implementation issues are addressed and computation times are presented.

In the fourth part, Chapter 9 exemplifies another possible application of hardware-based non-linear registration for the correction of DTI data. The location and the form of fiber tracts before and after correction are addressed with respect to the physical parameters of the measurement.

The fifth and final part of this doctoral thesis includes Chapter 10 which summarizes the presented work and addresses future prospects showing possible extensions of the introduced techniques. Additionally, Appendix A is included which presents some detailed derivations related to the presented simulation framework.

Part II

Registration

Chapter 2

Survey of Image Registration

In general, *registration* is the process of determining a geometrical transformation between corresponding points or regions defined in two or more images or coordinate spaces. This mapping can be utilized in two ways. One application is to employ it to calculate a transformed view of one of the registered images. A second possible purpose is an integrated display of information obtained from all (two or more) images, after registration has been performed. This visual integration of different sources is referred to as *fusion* (see Figure 2.1), which is often interchangeably used

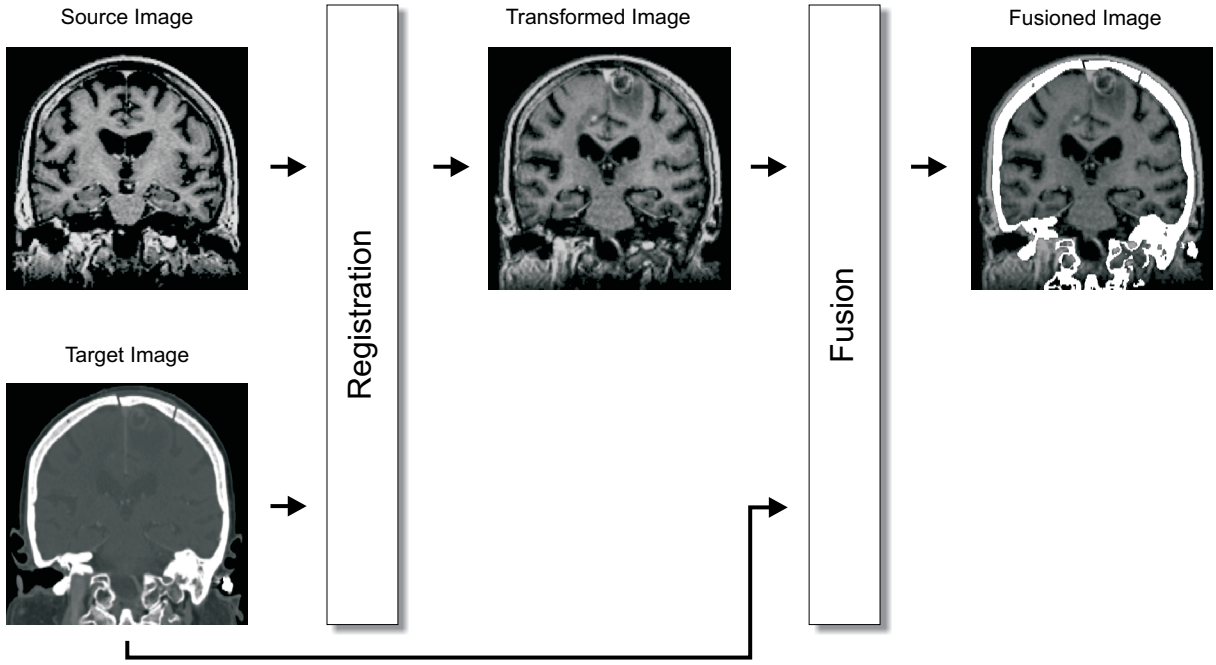

Figure 2.1: Registration and fusion. *Left:* slices from the source (MR) and the target (CT) dataset before registration. *Middle:* slice of the registered source dataset. *Right:* bone and soft tissue are combined in a fused view of the transformed source and the target image

with the term *registration*. In this thesis, these two terms will be consequently differentiated. The term *image registration* is a special case of the general notion of registration where only image spaces are considered. In the following chapters registration will be mainly discussed in this context.

In order to clearly speak about registration, it is useful to define two further notions. The *registration problem* is how to find the registering transformation. The algorithms that provide solutions to this are called *registration algorithms*. According to [26], registration algorithms can be categorized into three major research areas. In industrial computer vision, registration techniques are utilized for e.g. target localization, segmentation, motion tracking, light field rendering, stereomapping or in virtual reality, to name only a few. Remotely sensed data processing is the second important application area. In this respect, registration algorithms are used in cartography, for monitoring natural and urban resources, for multispectral classification and for many other tasks. The third application domain is medical image analysis, where registration is applied for the integration of functional information into anatomical images, tumor detection or in the context of image-guided surgery (for a detailed list of possible clinical applications see Section 2.2).

Generally, as depicted in Figure 2.2, registration algorithms can be divided into five steps. In the *preprocessing* step, the data is prepared (e.g. filtered or hierarchically resampled). A set of *features* used for registration is then manually or, preferably, automatically detected. In the next step, *matching* of the features detected in the source and in the target object is performed. In this context, feature descriptors and similarity measures can be used. Then, the *transformation* function is estimated, where the parameters defining the transformation are degrees of freedom in some optimization process. Computationally, this is usually the most expensive

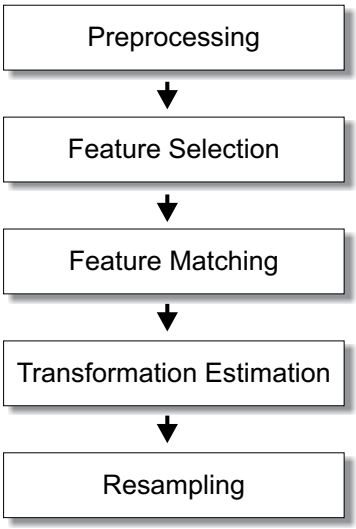

Figure 2.2: Five steps of an image registration algorithm

part of the registration. Finally, the source dataset is transformed and eventually resampled by means of the transformation calculated in the previous step.

The next two chapters of this thesis are concerned with image registration in the medical context. Several original algorithms are presented and different aspects of clinical application of the developed methods are covered. Firstly, in this chapter, basic definitions are given and previous work related to medical image registration is addressed. Various registration algorithms introduced in the literature are reviewed and classified according to the methods and the types of transformations that have been considered.

This chapter is divided into four sections. Mathematical definitions and formulations are provided in Section 2.1. Furthermore, registration is introduced in general and a classification of registration algorithms is presented revealing various application areas in Section 2.2. Section 2.3 focuses on registration with the use of rigid transformation. In this context, essential definitions are introduced and related work is reviewed. Existing, general, non-linear registration algorithms are the subject of Section 2.4.

2.1 Mathematical Background

Mathematically, an image $\mathbf{I}$ can be generally thought of as a mapping of points x from image space $\mathcal{D}$ to color values i_x from color space $\mathcal{C}$ according to:

$$\begin{aligned} \mathbf{I} &: \mathcal{D} \mapsto \mathcal{C} \ , \\ \mathbf{I}(\mathbf{x}) &= i_x \ . \end{aligned} \tag{2.1}$$

In real applications, $\mathcal{D}$ is a discrete grid domain over which an image is defined. This subdivides an image into a finite number of *pixels*[1] (2D) or *voxels*[2] (3D). Furthermore, the data coming from medical sensors are *intensity* images[3] only. Thus, in further considerations the term *image* will be used in this restricted sense, with regard to both constraints. The grey value subspace of the color space $\mathcal{C}$ will be denoted by $\mathcal{I}$.

The transformation resulting from registration can be considered in two ways: as a spatial transformation or as a mapping between image intensities. In the former case, registration is defined as a mapping f of position $\mathrm{x_s}$ in *source* space $\mathcal{D}_s$ (e.g. image space) to position $\mathrm{y_t}$ in *target* space $\mathcal{D}_t$ (e.g. physical space):

$$\begin{aligned} \mathbf{f} &: \mathcal{D}_s \mapsto \mathcal{D}_t \ , \\ \mathbf{f}(\mathbf{x_s}) &= \mathbf{y_t} \ . \end{aligned} \tag{2.2}$$

[1] Picture element.

[2] Volume pixel.

[3] Medical imaging devices can produce output images in which the value of each voxel is a single sample from the set $\{0, \ldots, 4095\}$. These images are typically displayed in shades of gray, varying from black at the smallest sample value to white at the highest. Therefore, they are called gray value or *intensity* images.

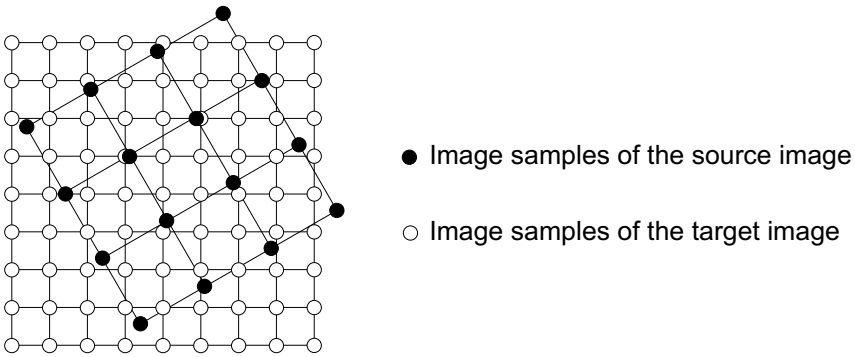

Figure 2.3: 2D image interpolation. The values of the target image are not necessarily defined in the discrete grid of the source image

Let us denote $\mathcal{I}(\mathcal{D}_t)$ and $\mathcal{I}(\mathcal{D}_s)$ by $\mathcal{I}_t$ and $\mathcal{I}_s$, respectively. With respect to intensities, registration transformation g can then be defined as:

$$\begin{aligned} \mathbf{g} &: \mathcal{I}_t \mapsto \mathcal{I}_s \ , \\ \mathbf{g}(i_t) &= \mathbf{I}_s(\mathbf{f}(\mathbf{x_s})) \ . \end{aligned} \tag{2.3}$$

We notice that the right-hand side of Equation 2.3 is well defined only if $\mathbf{f}(\mathcal{D}_s) \cap \mathcal{D}_s \neq \varnothing$. For the continuous definition of an image, this condition would usually be satisfied (e.g. for two images of the same object). However, assuming the discrete nature of images, it is unlikely that the value of $\mathbf{I}_s(\mathbf{f}(\mathbf{x}))$ is defined on the discrete grid of the source image for an arbitrary transformation f. An example for such situation in 2D is presented in Figure 2.3.

Thus, in order to perform a registration mapping, it is necessary to resample source image $\mathbf{I}_s$ on the discrete grid space $\mathcal{D}_t$. This produces another problem: after resampling we have to ensure that the condition $\mathbf{I}_s(f(\mathbf{x})) \in \mathcal{I}_s$ is fulfilled, which means that the calculated value has to be clamped to the discrete set $\mathcal{I}_s$. Inappropriate interpolation between image samples can lead to significant artifacts and to an incorrectness of the registration. This makes the choice of the resampling strategy one of the variability factors among the registration algorithms.

2.2 Applications

Numerous clinical applications take advantage of registration algorithms. In image-guided surgery, registration is one of the major issues. It is required to perform a spatial mapping from image space (and thus the information contained in the images) into the physical coordinate system of the patient. Another example is *multimodal registration*, where complementary information coming from different imaging modalities benefits the surgeon during diagnosis and therapy planning. In this section, we categorize medical image registration tasks according to the kind

of subjects that are involved. For each case, some exemplary medical applications are enumerated.

Altogether, in a clinical context, registration includes:

A. Establishing the correspondence between anatomical structures and their spatial relationship in images derived from various imaging modalities. It enables a more comprehensive image interpretation and better medical *diagnosis*.

Examples:

- Fusion of functional and anatomical imaging (e.g. fMRI and MR), which enables improved detection of functional activity in the brain [138, 125]. In *surgery planning*, this helps to conserve important structures of the human brain (e.g. motor cortex[4] or Broca's and Wernicke's areas[5]) during surgery.
- Registration of a positron emission tomography (PET) image to an MR image for localization of epileptogenic foci in the brain. The transformation can then be used in an image-guided system to guide *surgery* for the implantation of intracranial[6] electrodes [175].
- CT-PET registration for improvement of diagnosis and radiologic *therapy* of lung cancer patients [4].

B. Alignment of the imagery with the physical space of the patient and to track medical instruments within the surgical field.

Examples:

- Placement of dental implants with image-guided navigation system [30].
- Augmented reality systems superimposing computer images with real word images within an overlay display [18, 148].

C. Finding the spatial and temporal correlation between patient-specific computer models and the corresponding medical images.

Examples:

- Biomechanical modeling of brain deformation for intraoperative update of surgical planning [107, 190, 57, 97].
- Predicting breast soft tissue deformation with biomechanical models [176].

[4]This part of the human brain controls muscle movements throughout the body.

[5]These associated areas of the human brain are involved in language processing, both speech and understanding. Damage of one of them leads to a disability called *aphasia*.

[6]Stands for "within the skull".

- Reconstruction of elastic properties of breast [176] and brain tissue [106].

D. Calculation of the transformation between time-shifted images of a given patient, possibly taken under different conditions.

Examples:

- Comparison of registered scans from a given patient acquired at consecutive time points, which enables *monitoring* of disease progression (e.g. dementia [127] or multiple sclerosis [134]).
- Motion correction of contrast-enhanced breast MRI [142].
- Tracking of brain deformation during surgery and update of preoperative images using intraoperatively acquired MR images [57, 190, 71].

E. Construction of average models of the function or anatomy of an individual patient and relating them to an atlas.

Examples:

- Statistical models of brain [141, 62].

2.3 Rigid Registration

The choice of registration transformation is a central issue in any image registration procedure. This affects geometric constraints and distortions that can be produced during the registration process. In the context of medical image registration, the most fundamental spatial mapping describing the relationship between points in two images is the *rigid* transformation. This model assumes rigid-body behavior of an object, which means that an object can only change its position in space, but not its geometric shape. The rigid transformation has the property of preserving distances and internal angles. Such a model is suitable for inelastic objects, e.g. bone structures.

In three dimensions, six parameters are required to describe a rigid-body transformation. These are three translations along the x-, y- and z-axis and three rotations around these axes. In further considerations, we will denote the translations by t_x, t_y and t_z, and the angles of rotation by θ, ω and ϕ. The rigid transformation can be expressed as:

$$\mathbf{f}_{rigid}(\mathbf{x}) = \mathbf{x}' = \mathbf{R}\mathbf{x} + \mathbf{t}\ , \tag{2.4}$$

with $\mathbf{x} = (x, y, z)^T$ and $\mathbf{t} = (t_x, t_y, t_z)^T$.

The rotation matrix $\mathbf{R}(\theta, \omega, \phi)$ has the form (order of rotations: first around the x-, then y- and finally z-axis:

$$\begin{bmatrix} \cos\omega\cos\phi & \sin\theta\sin\omega\cos\phi + \cos\theta\sin\phi & -\cos\theta\sin\omega\cos\phi + \sin\theta\sin\phi \\ -\cos\omega\sin\phi & -\sin\theta\sin\omega\sin\phi + \cos\theta\cos\phi & \cos\theta\sin\omega\sin\phi + \sin\theta\cos\phi \\ \sin\omega & -\sin\theta\cos\omega & \cos\theta\cos\omega \end{bmatrix} . \tag{2.5}$$

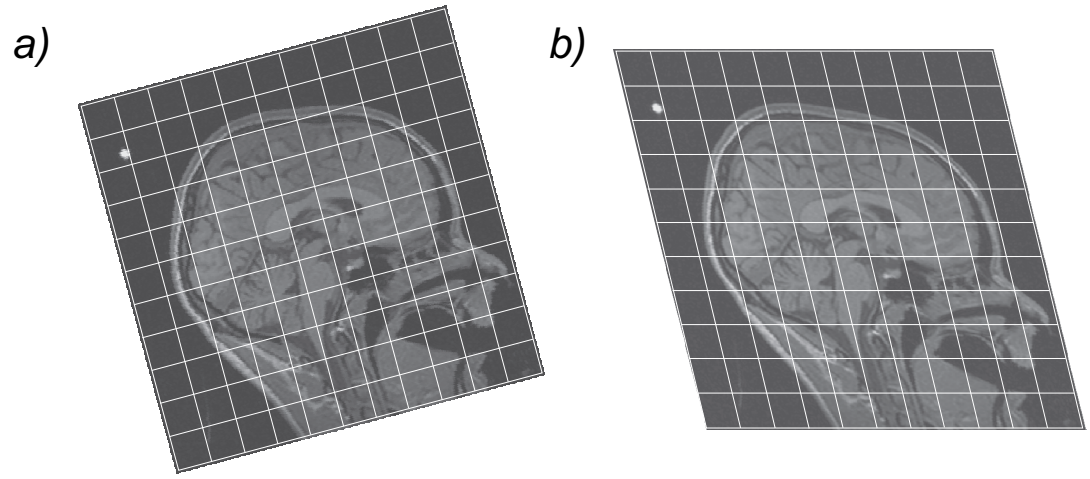

Figure 2.4: 2D example of rigid and affine transformations. *a)* Rotation. *b)* Shearing

Using homogenous coordinates, a single 4×4 matrix can be used to represent this transformation:

$$\begin{bmatrix} \mathbf{x}' \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} , \tag{2.6}$$

where $\mathbf{0} = (0, 0, 0)^T$, $\mathbf{t} \in \mathbb{R}^{3\times1}$ represents the translation vector and $\mathbf{R} \in \mathbb{R}^{3\times3}$ denotes the rotation matrix of Equation 2.5.

This transformation model can be extended by considering all but the last row in the homogenous matrix as free parameters resulting in 12 degrees of freedom. This type of transformation is called *affine* and can be used for an initial alignment of images from different subjects in order to create an image atlas. Thus, scaling is incorporated into the registration transformation in addition to translation and rotation. Another application of affine registration is the correction of scanner distortions. An example of a rigid and an affine 2D transformation are presented in Figure 2.4.

2.3.1 Related Work

There has been a lot of research in the field of rigid registration over more than 20 years. Here, we will review existing algorithms according to feature extraction and feature matching, as this is the most differentiating issue. In this section, we concentrate on rigid-body transformation methods. However, many of the techniques described here can also be adapted for non-linear registration. They will be addressed in Section 2.4.1.

2.3.1.1 Point Registration

Point registration involves the identification of corresponding points in the images or physical spaces. Based upon this correspondence, the three rotational and three translational parameters are calculated to determine the rigid registration. Generally, point-based registration can be divided into *intrinsic* and *extrinsic* methods.

In extrinsic methods, in addition to the image data, foreign objects are introduced specifically in order to perform registration. In the context of neurosurgery, stereotactic frames have been commonly used for this purpose [129]. In this technique, a reference frame is rigidly fixed with screws and pins to the patient's skull. Thus, a stable coordinate system is provided in the physical space to which the image space can be aligned. In [137], the accuracy of such a system[7] was inspected and reported to lie within 0.3 - 2.8 mm. Another group of extrinsic methods utilized non-invasive skin-attached [55, 199] *fiducial markers* which are individually designed for different imaging modalities and can be automatically identified within the respective scans. However, Maurer *et al.* [103] showed that only invasive bone-mounted markers guarantee the same accuracy as stereotactic frames.

Intrinsic point-based registration algorithms are based on the image information only. Salient points or features were utilized in various research works as a set of corresponding features for registration. Selecting corresponding anatomical landmarks manually by the user was one of the simplest strategies [54]. Thirion *et al.* [179] applied differential geometry to automatically identify corresponding points in two images. For this purpose, first, *crest lines* were extracted from the brain surface. In the second step, a very small number of *extremal points* on the respective crest lines were considered. Alignment of two small point sets was then iteratively calculated using an iterative closest point (ICP) algorithm [13]. Finally, this alignment was transferred onto whole datasets.

Based on the ICP technique, Maurer *et al.* proposed a weighted geometrical feature (WGF) algorithm to register CT images of the head to the physical space using one bone-implanted marker and points collected on the skin and bone surfaces [104]. With this method, they achieved a mean *target registration error*[8] (TRE) of 1.0 mm with standard deviation of 0.3 mm.

2.3.1.2 Structure-based Registration

Structures of higher order (e.g. surfaces, edges) have also been considered for rigid registration. Maintz *et al.* [99] compared several invariant *ridge* and *edge* extraction operators and applied them to the multiresolution rigid registration of CT and MR

[7] In this work, accuracy of the Cosman-Robert-Wells (CRW) stereotactic head frame system was evaluated.

[8] Measure of registration accuracy taking into account distance between corresponding points after registration other than those used for the estimation of the transformation (for formulas and details see [59]).

images. In the field of radiology, chamfer matching algorithm [20] was used for the registration of CT and PET lung images [29] or MR and CT pelvic images [184]. In this method, relevant structures were segmented in both the source and the target image. A distance transform giving the shortest distance to the segmented feature was then applied to the target image. Subsequently, optimization was performed to minimize the overall distances.

Probably the most popular methods in this context were based on explicitly segmented surfaces. Pelizzari *et al.* introduced a surface fitting technique called the *head and hat* algorithm [128]. In this approach, Powell's optimization method [133] was used for minimizing the distances between corresponding points in two surface-like structures (skin surface represented by a set of two-dimensional contours forming the "head" in the image with higher resolution and a "hat" consisting of unconnected points on the skin surface in the second image). In [64], a method for the registration of 3D images was presented based on a modification of a 3-4-5 chamfer distance algorithm for surface alignment. A general framework for the parametric rigid alignment of two 3D surfaces was proposed in [182]. Analytic formulas were presented there for the calculation of the required derivatives, thus reducing the computational expense of the non-linear Newton minimization strategy.

Chow *et al.* [33] utilized genetic algorithms for the matching of surfaces. Applying subsampling of one surface and a *kd-tree* structure for identifying points in the second surface, a significant acceleration was achieved for the evaluation of the fitness function for one chromosome. Thus, the overall computation time was drastically reduced.

In the very recent years, triangulated point clouds approximating the patient's face were acquired with specialized laser systems to register the physical space of the patient to the respective preoperative CT or MRI data [101]. This technique is very promising. However, it is constrained by the assumption that the original face geometry (concerning both soft tissue and bones) has not been destroyed in the course of surgery or otherwise.

2.3.1.3 Intensity-based Registration

Principal-axes and moments-based registration algorithms can be counted into the group of intensity-based algorithms (also referred to as voxel-based algorithms). Within these methods, the image center of gravity and its principal axes are computed. Registration then aligns the centers and the principal axes of two images [2]. Nowadays, this approach is mainly used for finding an initial alignment before another, more precise intensity-based algorithm is started.

Intensity-based registration algorithms (also referred to as voxel-based algorithms) using the full image content have an intuitive basis: if two images are well aligned, the intensity patterns in both images are correlated with each other. Relying on this assumption, a *similarity measure* as a function of two or more images can be defined which quantifies the degree of the similarity of involved images in

mathematical or statistical terms. Approaches based on the optimization of such functions measuring the quality of registration have dominated the field of multi- and monomodal medical image registration for more than the last 10 years.

In the context of monomodal registration, one of the simplest similarity measures is the *sum of squared intensity differences* (SSD). It was used for example for serial MR registration [70]. Junck *et al.* applied *cross correlations* originating from statistics theory for the alignment of functional brain images [83]. Both these measures, however, do not take into account possible differences in image intensities that can occur even if an affine transformation between two images of the same modality is considered (e.g. scaling can introduce artifacts that are not included in these models). A more general similarity measure called *correlation coefficient* (CC) allows a linear relationship between the intensity values in the images and was applied for the registration of CT data with camera-calibrated radiographs [91]. Finally, the *correlation ratio* (CR) was introduced by Roche *et al.* [136] as a multimodal extension of these concepts and was evaluated for rigid registration of MR, CT and PET scans.

An algorithm for the three-dimensional alignment of PET images was proposed by Woods *et al.* which introduced *ratio image uniformity* (RIU) as a further voxel similarity measure [197]. This concept was also used by the same author for serial MR registration [198]. An extension of this algorithm adapted for intermodal PET-MR registration was introduced in [196]. The modified measure was referred to as *partitioned intensity uniformity* (PIU) and inspired further research in the field of voxel-based registration techniques.

Beginning with the works of Collignon *et al.* [40] and Viola *et al.* [186], information theory approaches were utilized in medical image registration. They proposed *mutual information* (MI), which is based on Shannon entropy [151] as an appropriate similarity measure. Using this concept, registration is treated as a reduction of information in the combined image. This similarity measure is based on the statistical dependency between intensities in two datasets and not directly on the intensity values. Thus, it is suitable for the registration of images from different modalities. Currently, the most commonly used similarity measure is *normalized mutual information* [172] which is a normalized version of MI. This formulation overcomes the problem of MI sensitivity to changes in the image overlap region.

A voxel-based technique integrating registration and visualization was presented by Hastreiter *et al.* [75]. In this approach, the computation of the mutual information was performed with the use of a special purpose SGI graphics hardware. An example of an alternative method was presented in [200], where a set of filters was applied to 2D images to constitute a feature space. Based on this, feature clusters were formed and a genetic algorithm using an ellipse search technique for the reduction of computation time was started to find the optimal affine registration.

A very comprehensive retrospective comparison of rigid image registration techniques was presented by West *et al.* in [193] as a part of the so called Retrospective Image Registration Evaluation Project. All registrations were performed by par-

ticipants with no knowledge of the optimal results. A secondary goal of this study was the evaluation of the influence of geometrical distortion in MR images on the quality of the registration. In conclusion, the authors pointed out that may techniques are able to produce satisfactory results, however, each method still requires an additional visual verification of the achieved results.

That evaluation work confirmed that the described existing measures and methods do not solve all registration problems for an arbitrary clinical application. There is still a need for the development of improved measures and registration techniques. This thesis contributes to this field. A reliable rigid registration method is presented in Chapter 3 being a significantly optimized extension of the idea suggested in [75]. With this technique, a significant acceleration of computation times was achieved using the interpolation capabilities of graphics hardware.

2.4 Non-Rigid Registration

Rigid registration is sufficient in many applications except for situations where the underlying structures deform between successive scans. For example, soft tissue of the brain or liver can change its form significantly over time or due to interventions. This change in shape cannot be mathematically represented by translations and rotations only. In order to cover such deformation, *non-rigid* (non-linear) transformations have to be considered. In this context, an initial, *global* alignment is usually performed first using a rigid transformation. Then, a *local*, non-linear registration process is started.

Some of the non-rigid transformations can be written as a linear combination of higher-order basis functions, which can be expressed through a homogenous matrix (as it was possible in the definition of the rigid transformation). Most of them, however, differ very significantly, so that there is no single general formula to express an arbitrary non-linear transformation and a different mathematical formulation for each transformation is required.

There are many application fields for non-rigid registration. Non-linear algorithms were employed in order to eliminate distortions in echo planar magnetic resonance images (EPI) [170]. Another example is correction for motion in contrast enhanced MR images of the breast [142] or detection and quantification of intraoperative brain movement [102]. Non-linear algorithms were also applied for the generation of statistical atlases [62].

Another application is the compensation of the effects related to brain shift. This deformation is considered to be strongly non-linear and cannot be covered with the rigid transformation only. There has been very intensive research in this field in recent years [57, 190, 71]. However, due to the complexity of the problem, no satisfactory solution has been found so far. This fact was a motivation for developing original algorithms for non-linear registration which are presented in Chapter 4.

2.4.1 Related Work

There is a growing interest in the development of new and the evaluation of available non-linear registration techniques, documented in the number of works to this topic appearing recently [130]. In this section, a review of existing non-rigid registration algorithms is given, classifying the methods and emphasizing differences between them. The algorithms are categorized in similar groups as the ones for rigid registration in Section 2.3.1.

2.4.1.1 Point-based Registration

Analogous to the use of point landmarks in rigid registration, there exists a class of non-linear registration algorithms based on the specification of corresponding sets of points in two images. In order to extend the correspondence at chosen points to a transformation defined on the whole image domain, a *regularizing functional* has to be defined, which usually penalizes large values of the derivatives, thus smoothing the resulting registration transformation. In this context, e.g. *thin-plate* [19] or *elastic-body splines* [34] were applied. Thereby, in [34] definition of landmarks was not necessarily required. If they were present, they were incorporated in the boundary condition of the method. This correspondence can be defined as an approximation [139] or an interpolation condition [19].

A generalized concept of point-based registration was presented in [37]. In that method, the binary correspondences between points were replaced with partial matches, which made the energy function behave better during optimization. For the purpose of optimization, deterministic annealing was employed incorporating a special term related to the weights of point-to-point relations.

Davis *et al.* [49] presented a framework based on a neural network to compute the deformation field between two images. Synaptic weights were established according to landmark positions or based on the intensity similarity measure only.

2.4.1.2 Structure-based Registration

The concept of crest lines (also see Section 2.3.1.2), was adapted in [173] to obtain a morphometric anatomical atlas of the human skull. Extracted lines served as characteristic features, which were non-linearly matched in an iterative framework.

A hybrid technique presented in [93] involves feature-based volumetric registration as initialization for a subsequent attribute-based surface registration. The attribute was defined for each vertex as a histogram of changes in the direction of the normal vector in the neighborhood of a considered vertex. Curvature and normal direction differentiating cortical *gyri*[9] and *sulci*[10] were also used by Wang *et al.* [187] to establish a non-linear transformation between cortical surfaces. In this method,

[9] *Gyri* is the plural form of *gyrus*, which is a Latin term for convolutions on the cortical surfaces.
[10] *Sulci* is the plural form of *sulcus*, which stands for narrow furrows separating adjacent gyri.

geodisical interpolation was hierarchically applied for the reconstruction of the transformed surface.

A non-linear technique for the analysis of brain shift was proposed by Hastreiter *et al.* [76]. In this approach, brain geometries were extracted first with a semi-automatic method based on deformable surfaces. The obtained surfaces were then registered with the ICP algorithm and the results were visualized with a color-coding technique giving a good impression of the occurring deformation. Thompson *et al.* [181] used surfaces of the ventricles and the cortex to find a mapping between human brain images. The underlying data structure of the algorithm was a hierarchical graph consisting of parametric surface meshes. Several deep sulci were separately modeled within this system. All surfaces in the graph were then parameterized over a uniform rectangular domain and were mapped onto each other. The surface-based transformation was then extended onto the whole brain volume using a weighted linear combination of radial function.

Polynomial transformations of higher order were considered in [39] for a non-iterative geometric registration of animal brain surfaces. Prior to the computations, surfaces to be registered were firstly converted into a representation based on B-splines. Inflection and torsion points and a function of the zero, first and second derivative were then selected to be affine invariants. Finally, correspondence between the landmarks was established and the registration transformation was calculated.

2.4.1.3 Intensity-based Registration

The intensity-based non-linear registration algorithms, in comparison to the analogous class of rigid registration methods, are characterized by a much greater variability. For both rigid and affine registration, the main differentiating element is the voxel similarity measure. In the context of non-rigid registration, the similarity measure is not an inherent part of a non-linear registration method and can be easily exchanged. The most essential factor differentiating non-linear approaches from each other is the transformation that is considered. The choice of transformation is crucial, since it influences the kind of possible geometric distortions within a given non-linear method.

Optical flow

The concept of optical flow originates from computer vision. It was introduced as a method for the detection of the relative motion of an object in a time sequence of images. This concept is based on a mathematical formulation, where the image intensity at a given point before and after a movement in a given time period remains constant [79].

Optical flow was related by Thirion *et al.* to a diffusion process [180]. In this approach, the objects' boundaries in one image were considered as a semi-permeable membrane. They let the other image diffuse through this interface.

The driving force for this process was given by the optical flow. The presented algorithm was validated for two-dimensional synthetic deformations, for the analysis of cardiac motion and for inter-patients 3D image matching. Very recently, a model of lesion growth was coupled with the described technique for brain atlas deformation in the presence of large tumors [44].

A hierarchical optical flow model which allows for both global and local motion between two image datasets was presented in [185]. Within this model, the flow field was represented with a B-spline basis. For the optimization, a modified Newton method was utilized, where the Hessian of the squared intensity differences function was estimated at the optimum prior to starting iterative computation. This technique ensured a better stability and convergence.

A validation study of an optical flow method for the estimation of myocardial[11] displacement in MR tagged cardiac images was conducted by Dougherty *et al.* [52]. This method overcome the requirement of constant pixel intensity in images over time by suppressing in a preprocessing step the intensity bias coming from reduction in stripe contrast throughout the cardiac cycle.

B-Splines

Free-form deformation (FFD) is the concept that has been used extensively in computer graphics for animation purposes [150]. The main idea is to surround an object with a mesh of control points parameterized over a uniform regular domain. The volumetric form of this embedded object can then be modified by changing the positions of the control points. FFDs are based on locally controlled basis functions.

In the context of medical image registration, B-splines were considered as underlying basis functions for FFD [142]. In that work, B-splines were employed for non-linear registration of contrast enhanced breast MRI. The local influence scope of the B-Spline basis functions was the main advantage of this approach and allowed a reduction in computational complexity. Thin-plate regularization was applied to ensure smoothness of the registration transformation.

A generalization of [142] based on multi-level B-splines was presented in [149]. Non-uniform control point distribution for the acceleration of computation was simulated there by associating an *active* or *passive* status with each control point. The algorithm was validated in a variety of medical applications, e.g MRI brain tissue deformation compensation, correcting for breathing motion in liver images or registration of preoperative MRI to CT electrode implant data.

[11] Pertaining to the heart muscle.

Non-linear registration based on B-splines was employed for the tracking of the myocardium and heart ventricles [95]. In this method, an automatically built cardiac atlas was deformed to align temporally and spatially with images in a cardiac cycle. This method is suitable for labelling specific regions of the heart.

Elastic Registration and Mechanical Models

Mechanical models of soft tissue deformation have been widely investigated in the context of registration. There exist simulation approaches that are based on physics only, where a physically-based model is defined and image data is exclusively used for extracting the objects to be modeled. These algorithms are described in Sections 5.1 and 5.2. Another class of methods utilizes simulation techniques for registration. For this purpose, it uses information contained in the images more extensively.

Bajcsy *et al.* [6] were first to demonstrate non-rigid registration of medical images. They modeled image deformation as a physical process, based on Navier-Stokes equations for linear elasticity. Within this system, *external* forces worked against *internal* body forces[12] correlating similar (in terms of intensity) regions in two images. This resulted in the deformation of the elastic body in an equilibrium state.

An extension of this approach applied to 3D brain images was presented in [48]. In that framework, the inhomogeneity of the brain was incorporated, e.g. ventricles were given higher elastic parameters so that they could deform more freely than other brain regions. A similar, linear elastic model was analyzed by Ferrant *et al.* [57]. They presented a comprehensive framework for elastic registration with emphasis on the registration of pre- and intraoperative MR brain data for the update of preoperative imagery. Shape information was extracted from the objects and correspondences were defined with an active surface algorithm to specify the boundary conditions.

A set of coupled differential equations was used by Hagemann *et al.* [69] to develop a biomechanical model of the human head. Hooke's law and the theory of elasticity were combined in a finite element method. The model was validated on a set of pairs of 2D MR head images. Different physical descriptions for rigid, elastic and fluid structures were investigated in [68]. Stoke's equation was utilized for modeling incompressible fluids, whereas the Navier equation was taken for elastic and rigid materials with appropriate elastic constants. In both models, boundary conditions were defined according to established landmark correspondences in the images to be registered.

A further example from this group of methods was presented in a study evaluating the influence of different elasticity parameters for biomechanical breast

[12] These forces are often referred to as *stresses*.

models and the accuracy of the calculated displacement field [176]. Boundary conditions were derived based on image forces resulting from non-linear registration based on an approach presented in [149]. Subsequently, simulation was performed and the resulting transformation was compared to the result of the non-rigid registration used for setting the initial boundary conditions.

A variational framework for non-linear registration based on a curvature type smoother was introduced by Fischer *et al.* [58]. In this approach, internal forces were designed to minimize the curvature of the displacement field. This technique allowed for automatic affine alignment, thus eliminating the need for initial rigid registration. In order to solve the resulting large linear equation system, the authors proposed a fast and stable direct solver adapted to the underlying mathematical model.

Viscous fluid model

As an alternative to elastic registration, Christensen *et al.* proposed a *viscous fluid* model [35]. In the elastic registration concept, stresses increase proportionally to the extent of deformation, thus not allowing the modeling of extensive localized deformations. In the viscous fluid model, internal forces are relaxed over time, which permits e.g. modeling of sharp corners. As a disadvantage, however, the probability of a misregistration increases. An accelerated implementation based on convolutions with Green's function was described by Bro-Nielsen [24].

A model in which the viscosity of the fluid was not constrained to be spatially homogenous was proposed by Lester *et al.* [92]. Within this model, different regions could deform to different degrees. An extension of the viscous fluid technique allowing multimodal registration was introduced recently by D'Agostino *et al.* [46]. Parzen windowing was used to estimate the joint histogram required for calculation of the normalized mutual information.

Experiments with the registration of MR human brains with a hybrid algorithm combining both viscous and elastic methods were presented in [168]. That algorithm was composed of two levels of regularization. The first one acted on the correction field and could be interpreted as a viscous fluid constraint. On the other hand, the regularization of the displacement field imposed an elastic behavior.

Chapter 3

Hardware-based Rigid Registration

In the previous chapter, existing rigid and non-linear registration algorithms have been comprehensively reviewed. The approaches have been classified according to various algorithmic issues and to possible medical applications. Despite the large number and variability of registration methods, the registration problem still remains a challenge for applications. Even if an algorithm can provide reasonably accurate solutions, the main drawback are the extremely high computation times. This is due to the huge number of interpolations that have to be performed within a given registration procedure. This holds especially for non-linear registration, where a similarity measure has to be evaluated numerous times during optimization. To a smaller extent, but still significantly, rigid registration algorithms are also affected by this constraint. In the context of time efficiency, another concern is the choice of the similarity measure which is very important for the convergence rates during optimization course. Using a similarity measure suited to a given application is also an essential issue affecting the quality of the registration.

Our work in the registration field aims at the intraoperative use and possible integration within image-guided systems. This implies very rigorous requirements for the developed algorithms with respect to the computation time, which makes the employment of time expensive registration algorithms impossible. Therefore, the development of accelerated and at the same time still accurate registration methods has been a point of interest in our research.

A corrected version of "Moore's law", originally postulated in 1965 [118], says that the number of transistors contained on a single processor unit doubles every 18 months. In comparison to this, the computation power of graphics chips doubles every 9-12 months[1] (see Figure 3.1). This fact and the previous considerations about the high number of interpolations, which are the main source of time inefficiency (or the main point of improvement), have motivated the investigations of the application of graphics hardware for registration purposes. Within this thesis, a hardware-based approach has been implemented which utilizes the interpolation capabilities

[1]This rule is known as "Moore's law squared".

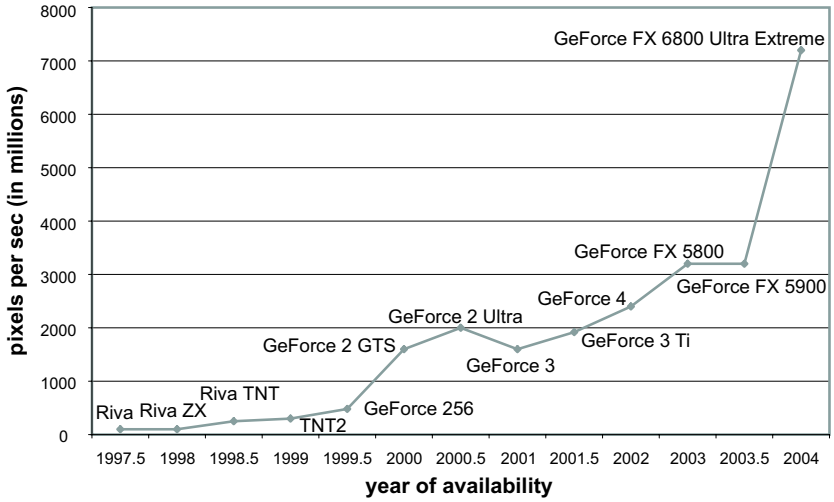

Figure 3.1: Development of PC graphics chips in the last years (shown for Nvidia graphics adapters). As measure of performance the peak fill rate expressed as the number of texels processed per second is considered

of *graphics hardware* for the acceleration of rigid registration. This chapter presents various constitutive elements of this approach, revealing both its algorithmic issues and its value for medical applications.

To understand the methods described in this chapter, it is necessary to know the functionality of graphics hardware. Therefore, in Section 3.1, the concept of the *graphics pipeline* is introduced and its working principles are explained. In Section 3.2, different similarity measures that are important components of the registration algorithm are addressed. Details about the constituent steps of the hardware-based rigid-registration method are addressed in Section 3.3. Furthermore, results of registration experiments with real clinical data are presented and discussed in Sections 3.4 and 3.5. Finally, this chapter is concluded in Section 3.6.

3.1 Basics on Graphics Hardware

Graphics cards have been progressing very rapidly in recent years. Both the general availability of this specialized hardware and the way information is processed on graphics cards have changed significantly. New features and greater flexibility have been introduced into the concept of graphics adapters. Nevertheless, the general architecture of graphics programs has remained unchanged: calculations are executed as a series of elementary steps constituting the graphics pipeline. These steps operate in parallel and in a fixed order, where each stage receives its input from the prior stage and sends the result to the consecutive stage. In addition to this, software interfaces have been provided which make proprietary programming at almost all stages of this pipeline possible.

The elementary steps of the graphics pipeline are depicted in Figure 3.2. The input of the graphics pipeline is a stream of vertices which usually result from the decomposition of a more complex scene into planar polygons. The vertices are firstly handled by the ***geometry processing unit*** which is responsible for the specification of the position of objects in the virtual world and for defining the viewing coordinate system. This unit executes on a *per-vertex* basis linear transformations (e.g. translations or rotations) each expressed in the form of a 4×4 homogenous matrix. All transformations performed at this stage can be combined to a single *modelview* matrix. In this geometry unit, local illumination is also calculated for each of the transformed vertices. In the most recent graphics hardware this stage of the graphics pipeline is programmable, which means that proprietary vertex transformations can be applied[2]. Moreover, predefined lighting models can be exchanged against the user-specific ones.

After this step, the projection matrix is applied and a normalized *clipping volume* is defined. Parts of rendering *primitives* (triangles, lines or points) generated from vertices are then removed if they lie outside this volume. Primitives can also be discarded in a *culling* process depending on whether they face forward or backward, with respect to the current viewing position. Finally, the *viewport* transformation is applied in the geometry unit in order to transform the coordinates of the vertices onto the viewing plane associated with the output device.

The second main element in the graphics pipeline is the ***rasterization unit***. Here, geometric data is scattered into *fragments*, which is usually referred to as the rasterization process. Each pixel on the screen usually corresponds to one fragment[3]. During a setup phase, interior fragments of geometric primitives are determined with the use of a *scan-line* algorithm. Up to this point, only vertices have had assigned color and depth values. Based on this information, now color and depth values are interpolated for each fragment. The same interpolation principle is utilized for the calculation of the texture coordinates required for later use.

Shading algorithms are then applied to the obtained fragments. For this purpose, the rasterizer calculates shading values for the endpoints of each scan-line and subsequently expands the shading to the whole scan-line. The next important stage involves *textures*[4]. Textures are mapped onto corresponding fragments according to the texture coordinates that have been distributed in an interpolatory way to each fragment at the previous pipeline stage. This mapping combines the calculated texture value with the color value previously stored in the respective fragment. Similar to the concept of the vertex programs, hard-wired operations can

[2]Programs using a specialized instruction set which customize the operations performed within the geometry processing unit are called *vertex shaders*.

[3]Relating many fragments to one pixel is also possible for multi-pass rendering.

[4]Textures are two- or three-dimensional images parameterized over a respective space. The coordinates in this space are called *texture coordinates*.

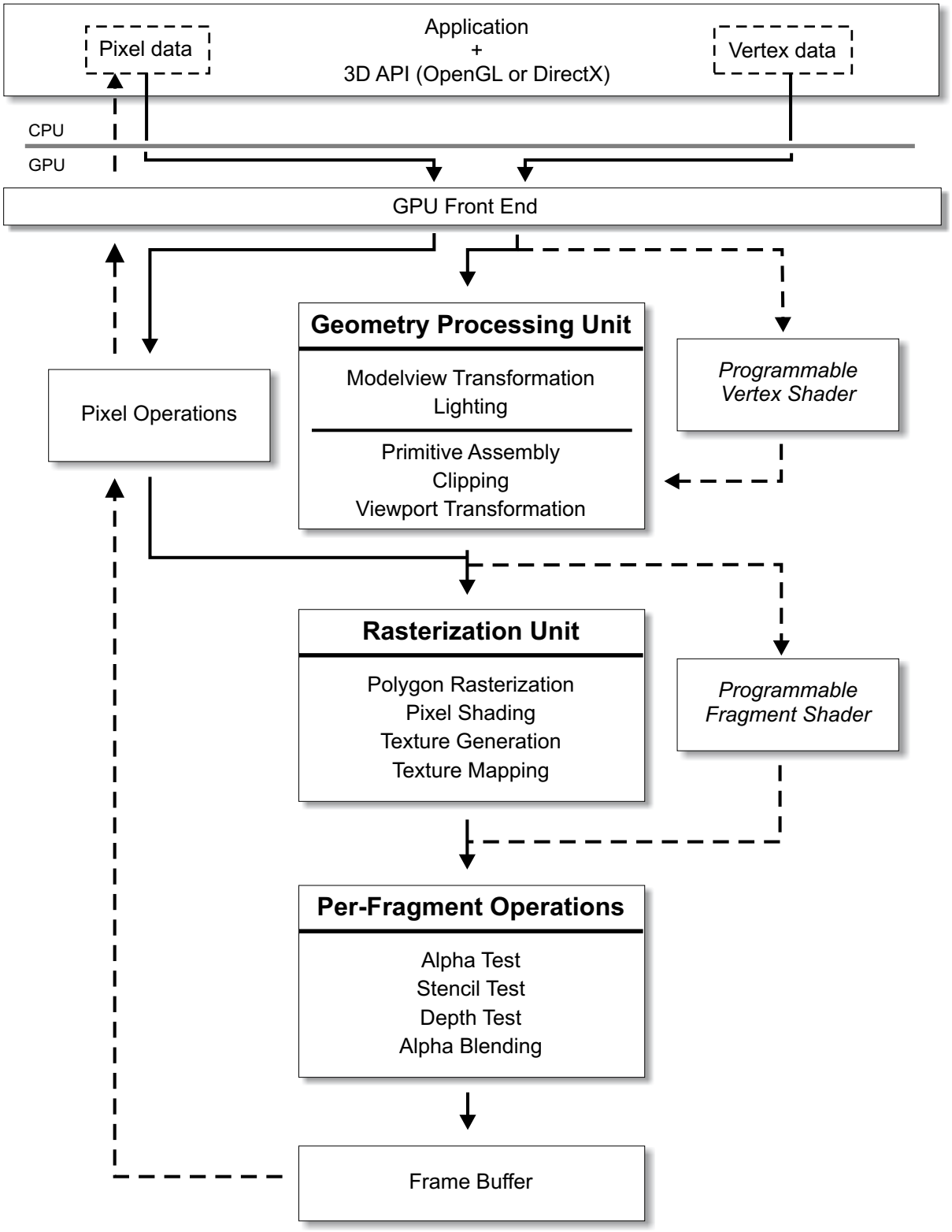

Figure 3.2: The graphics pipeline

be replaced at this stage with user-defined functions[5]. This creates new possibilities to exploit graphics hardware in a more flexible manner.

The third element of the graphics pipeline are the ***per-fragment operations***. In this unit, additional commands can be applied to fragments. Firstly, a number of tests are conducted on each fragment. An *alpha test* is performed in order to eliminate translucent fragments that do not contribute significantly to the final color value. A *stencil test* is performed where fragments are discard on the basis of a defined binary mask. During the *depth test*, occluded fragments are skipped by comparing the depth value of the processed fragment with the respective value stored in the *depth buffer*. *Alpha blending* combines the color of the fragment with the currently stored corresponding pixel color value, if this function is enabled. Finally, the *write* operation replaces the colors of the pixels in the frame buffer with the newly calculated values.

The hardware-based registration algorithms presented in this thesis build upon the described graphics pipeline architecture and focus on increasing the speed of the calculations while simultaneously achieving high quality results. In this context, a comprehensive description of the usability of hardware-based techniques for registration purposes will be provided in this chapter and in Chapter 4. Furthermore, the features, advantages and limitations of graphics adapters will also be addressed.

3.2 Similarity measures

Registration algorithms working directly on image intensities are almost always based on the same general principle, which requires the existence of a function measuring the similarity of the images considered for registration. The value of this function is iteratively optimized in order to determine the registration transformation. The source image is thereby transformed at each iteration according to the current estimate of the transformation. For this purpose, the image must be resampled for the recalculation of the similarity measure, which involves interpolation between source image samples on the discrete grid of the target image (see Section 2.1).

An appropriate choice for the underlying similarity measure is essential in many aspects. Usually this measure has to be calculated many times during a registration. Therefore, the computational costs of evaluating this function (including the time complexity of one function calculation and the total evaluation time related to the convergence rate achieved with a given similarity measure) should be as low as possible. On the other hand, the crucial issue for the registration quality is the choice of a similarity measure appropriate to the specific application (e.g. according to the involved image modalities). Under certain conditions this can imply higher

[5]Specialized programs making use of this possibility are commonly called *pixel* or *fragment shaders*.

computation times, which excludes a method from consideration in intraoperative use.

In Section 2.3.1.3, an overview of intensity-based similarity measures has been provided. Although there exist a number of measures potentially suitable to the context of registration, in practice recent image registration algorithms mainly involve the use of two similarity measures: the normalized mutual information (NMI) [172] or the correlation ratio [136]. The correlation-based methods, however, assume a linear dependence between the intensities, which restricts their application for a multimodal registration problem.

In contrast, the currently most generally formulated similarity measure is NMI. This information theoretic function is based on Shannon entropy [151] which, in reference to an image, is a measure of the diversity of the intensity distribution and is defined as:

$$\mathbf{H}(\mathbf{I}) = -\sum_i p_{\mathbf{I}}(i) \log p_{\mathbf{I}}(i) \ , \tag{3.1}$$

where $p_{\mathbf{I}}(i) = \mathbf{P}(\{\mathbf{x} : \mathbf{I}(\mathbf{x}) = i\})$ is defined as the probability of the occurrence of intensity i in an image $\mathbf{I}$. The number of elements in the sum can either reflect all possible intensities in the image (MR or CT images can have up to 4096 intensity values) or be a reduced number of intensity bins. In our method we use 256 bins.

The so defined entropy is zero when occurrence of an intensity (or a group of intensities, if bins are considered) in the image is certain ($p_{\mathbf{I}}(i) = 1$) and is maximal when all intensities have the same probability ($p_{\mathbf{I}}(i) = \frac{1}{n}, \ i = 1, \ldots, n$). In the classical image registration problem, two images are considered. Therefore, two intensities at each voxel position are provided. In this context, the joint entropy measures the amount of information that is comprised in the source image $\mathbf{I}_s$ and the target image $\mathbf{I}_t$:

$$\mathbf{H}(\mathbf{I}_s, \mathbf{I}_t) = -\sum_i \sum_j p_{\mathbf{I}_s,\mathbf{I}_t}(i,j) \log p_{\mathbf{I}_s,\mathbf{I}_t}(i,j) \ . \tag{3.2}$$

This joint distribution is defined for the voxels in the overlapping region of the target and the source images. $p_{\mathbf{I}_s,\mathbf{I}_t}(i,j)$ is thus the joint probability of observing the pair of intensities (i,j) at corresponding voxel locations in the target and in the transformed source image. Intuitively, registering images produces a small number of entries with high probabilities and increases the number of elements with probability equal zero, which minimizes the joint entropy.

Considering only the joint entropy in the similarity measure would make the registration very dependent on the overlapping volume of the images. Thus, the mutual information [40, 192] also incorporates the marginal entropies related to the intensity probability distribution in the overlapping region for each image separately. Altogether, the mutual information is given by:

$$\mathbf{M}I(\mathbf{I}_s, \mathbf{I}_t) = \mathbf{H}(\mathbf{I}_s) + \mathbf{H}(\mathbf{I}_t) - \mathbf{H}(\mathbf{I}_s, \mathbf{I}_t) \ . \tag{3.3}$$

Substituting into this formula the respective expressions from Equation 3.1 and Equation 3.2 for the entropies, a simplified form can be obtained:

$$\mathrm{M}I(\mathbf{I}_s, \mathbf{I}_t) = \sum_i \sum_j p_{\mathbf{I}_s,\mathbf{I}_t}(i,j) \log \frac{p_{\mathbf{I}_s,\mathbf{I}_t}(i,j)}{p_{\mathbf{I}_s}(i) \cdot p_{\mathbf{I}_t}(j)} \ . \tag{3.4}$$

Registration using mutual information aims to reduce the uncertainty for a value in one image, if the value at the corresponding location in the second image is known. This means that the amount of information that one image contains about the other is maximized at alignment.

Mutual information is not entirely independent of the overlapping region of two images. Changes in some regions can disproportionately influence the value of the similarity measure. Therefore, a normalized mutual information (NMI) [172] has been proposed. In this formulation, mutual information is normalized with respect to the joint entropy:

$$\mathrm{N}MI(\mathbf{I}_s, \mathbf{I}_t) = \frac{\mathrm{M}I(\mathbf{I}_s, \mathbf{I}_t)}{\mathbf{H}(\mathbf{I}_s, \mathbf{I}_t)} + 1 = \frac{\mathbf{H}(\mathbf{I}_s) + \mathbf{H}(\mathbf{I}_t)}{\mathbf{H}(\mathbf{I}_s, \mathbf{I}_t)} \ . \tag{3.5}$$

This normalized version has been shown to be considerably better than the standard mutual information and is therefore used in our registration framework.

3.3 Registration

Generally, registration algorithms are characterized by high computing times. This is a result of the fact that, during an iterative optimization procedure the similarity measure has to be evaluated many of times. For this purpose, one dataset has to be completely recalculated at each iteration, which is a very time consuming operation due to the huge number of required interpolations.

In this section, an efficient rigid registration approach is detailed. It greatly extends the idea introduced by Hastreiter *et al.* [75]. The algorithm implemented here is designed for general purpose PC graphics hardware and allows fully automatic hierarchical registration. Contrary to a variety of existing software-based approaches, the basis for the method presented here is the exploitation of graphics hardware for the acceleration of the computations. Especially, trilinear interpolation is performed on a graphics card since this task can be calculated very efficiently using the 3D texture mapping subsystem of the rasterization unit. This and further algorithmic and implementation issues of the hardware-based rigid registration approach are addressed according to the following steps:

1. OpenGL setup necessary for using 3D textures

2. Iterative evaluation of the similarity function which can be split into:

- Transformation of the *source* dataset and its reinterpolation in the grid of the *target* volume
- Calculation of a similarity measure between the source and the target volume
- Multiresolution optimization

3.3.1 3D Textures

In order to execute any OpenGL commands on the datasets considered for registration, the images have to be available as 3D texture volumes in the graphics hardware memory. In Listing 3.1, a framework accomplishing this task is presented with the OpenGL API, although an equivalent implementation with DirectX is also possible.

Within this implementation framework, the OpenGL state machine has to be switched first into a mode operating with 3D textures. This mode can be activated with a call to `glEnable(GL_TEXTURE_3D)` (line 1). The texture name `texName` generated in lines 7-8 is bound in line 11. Lines 16-18 specify how the texture is to be wrapped. Furthermore, a set of texture environment commands is used in order to set the trilinear texture interpolation behavior (lines 21-22). A texture function affecting the produced fragment color is specified in line 25. After this, the data has to be transferred from the main memory into the graphics adapter. For this purpose, the OpenGL function `glTexImage3D(...)` is utilized (lines 28-38). Thus, the parameters `sizeX`, `sizeY` and `sizeZ` are the dimensions of the 3D texture[6]. The parameter `dataset` is the address of a one-dimensional array containing intensities of all voxels. The voxels are stored in an inverse linear order, which means that for voxel coordinates (x, y, z) the fastest changing coordinate is x and z is the slowest changing coordinate.

3.3.2 Reslicing Algorithm

The OpenGL functionality of 3D textures is "slice-oriented", which means that considering a given 3D texture volume, simple geometric forms can be textured based on setting a correspondence between vertices in the geometry space and texture coordinates forming planar polygons in the 3D texture space. The OpenGL setup for assigning texture coordinates to a vertex is presented in Listing 3.2.

Due to the underlying algorithmic techniques, the registration procedure is closely related to a visualization task [75]. Hardware-based 3D texture mapping has been used by Cabral *et al.* for direct volume rendering [28]. In this method, the dataset considered for visualization is first loaded into the 3D texture memory of the graphics card. Equidistant planes perpendicular to the viewing direction

[6]These dimensions always have to be a power of 2. If the dimensions of the datasets considered for registration do not fulfill this condition, additional "empty" slices have to be added to the 3D texture.

```
0   // Enabling 3D texture mode
1   glEnable(GL_TEXTURE_3D);
2
3   // Set pixel storage mode
4   glPixelStorei(GL_UNPACK_ALIGNMENT, 1);
5
6   // Generate a texture name
7   GLuint texName;
8   glGenTextures(1, &texName);
9
10  // Bind the texture name
11  glBindTexture(GL_TEXTURE_3D, texName);
12
13  // Set texture parameters
14
15  // Set wrap parameters for the texture
16  glTexParameteri(GL_TEXTURE_3D, GL_TEXTURE_WRAP_S, GL_CLAMP_TO_EDGE);
17  glTexParameteri(GL_TEXTURE_3D, GL_TEXTURE_WRAP_T, GL_CLAMP_TO_EDGE);
18  glTexParameteri(GL_TEXTURE_3D, GL_TEXTURE_WRAP_R, GL_CLAMP_TO_EDGE);
19
20  // Set interpolation mode to trilinear
21  glTexParameteri(GL_TEXTURE_3D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
22  glTexParameteri(GL_TEXTURE_3D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
23
24  // Specify a texture function acting on a textured fragment
25  glTexEnvi(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
26
27  // Define 3D texture image
28  glTexImage3D(GL_TEXTURE_3D,       // target
29               0,                   // level of details
30               GL_LUMINANCE,        // internal format
31               sizeX,               // width
32               sizeY,               // height
33               sizeZ,               // depth
34               0,                   // border
35               GL_LUMINANCE,        // pixel format
36               GL_UNSIGNED_BYTE,    // pixel data type
37               dataset              // pointer to the image data
38              );
```

Listing 3.1: OpenGL setup for initialization of 3D textures

```
0   // Set the current texture coordinates
1   glTexCoord3d(r, s, t);
2
3   // Set the vertex coordinates associated with
4   // the current texture coordinates
5   glVertex3d(x, y, z);
```

Listing 3.2: OpenGL setup for setting geometry-texture correspondence

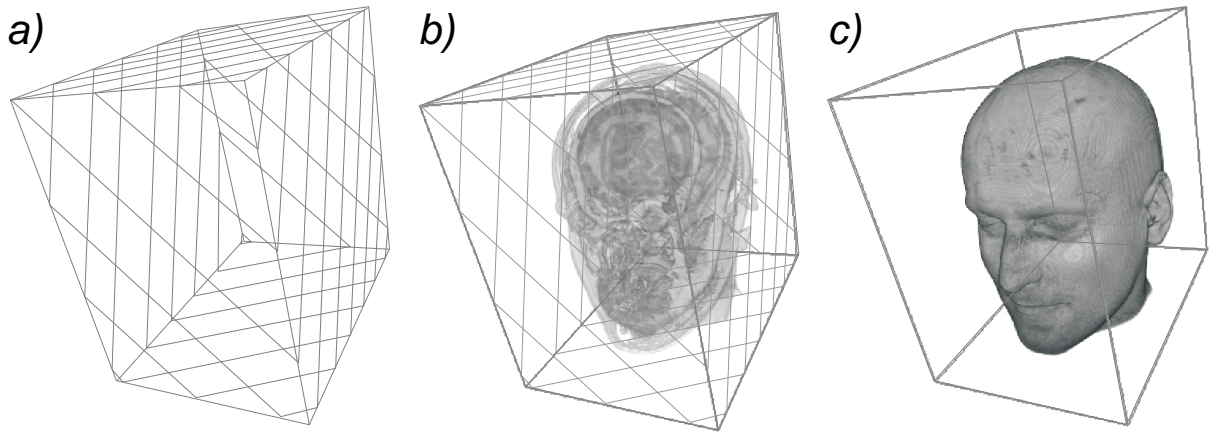

Figure 3.3: Rendering based on 3D textures. *a)* Geometry for rendering with slices parallel to viewport. *b)* 3D texture mapped polygons. *c)* Rendered image

are then clipped against the bounding box of the visualized volume. Subsequently, hardware-accelerated texturing is applied to this geometry and the resulting textured polygons are blended in back-to-front order and rendered into the framebuffer, as depicted in Figure 3.3.

Similar to this visualization concept, texture and imaging subsets are employed in our registration approach in a slice-oriented manner in order to reslice the volume data after a rigid transformation. In the context of texture mapping, for a given level of detail, *nearest neighbor* and *trilinear interpolation* strategies are supported in the OpenGL specification and its implementations. In the former method, the value of the texture element that is nearest to the center of the voxel being textured is assigned to this voxel. The latter technique assigns a given voxel being textured the weighted average of the eight *texels* closest to the center of this voxel. The presented registration approach utilizes trilinear interpolation in order to reduce artifacts during reslicing.

In the initial step of the registration algorithm, the source dataset is loaded into the 3D texture memory of a graphics card, using the framework presented in Section 3.3.1. Then, an iterative optimization procedure is started. In each iteration, the following calculations have to be conducted.

The target dataset is sliced by a set of equidistant planes $\{p_i\}_{i\in I}$ in the direction perpendicular to one of the main coordinate axes. The distance between the consecutive planes is set according to the resolution of the dataset along the axis perpendicular to the planes. The intersections of the planes with the *bounding box* of the target image volume define image slices, where the number of these slices is equal to the resolution of the image in the respective dimension (see Figure 3.4a). Let us denote by $\mathrm{f}_{\mathbf{current}}$ a current estimation of the registering rigid transformation mapping the source image onto the target image. With the inverse ${\mathrm{f}_{current}}^{-1}$ of this linear function, each plane p_i containing one slice of the target image is transformed into the coordinate system of the source dataset, defining a transformed plane $p_i{}'$.

For each transformed plane $p_i{}'$, its intersection with the bounding box of the source volume defines a planar polygon[7] (as in Figure 3.4b) or is an empty set. If the intersection is non-empty, the vertices $\{v_{ij}{}'\}_{j\in J'}$ of the resulting polygon are transformed back using $\mathrm{f}_{current}$ into the target volume, defining the corresponding vertices $\{v_{ij}{}''\}_{j\in J''}$. The transformed polygon is then clipped against the bounding rectangle of the corresponding slice in the target volume to form a new polygon with vertices $\{\bar{v_{ij}}\}_{j\in \bar{J}}$ (see Figure 3.4c)).

Parallel projection is applied to the resulting polygon, projecting it into the viewport of a size defined by the respective slice in the target volume. Clipping is then performed. Finally, the polygon is rasterized and textured according to the specified texture coordinates and rendered into the framebuffer (see Figure 3.4d)).

Applying this transformation technique to all planes $\{p_i\}_{i\in I}$, the corresponding slices from the overlapping volume of both 3D datasets are interpolated onto the discrete grid of the target dataset. This is the basis for the calculation of the joint histogram, which is required for the evaluation of the NMI. For this purpose, the interpolated data is read back from the framebuffer with the use of the `glReadPixels()` command. According to the intensity values read at the corresponding voxel positions, the respective entries in the joint histogram are generated for each pair of slices from the target and the interpolated source image. Since individual histograms of the images can be considered to be marginal distributions of the 2D histogram, they are obtained by summation of the respective columns or rows of the joint histogram. Finally, the value of the normalized mutual information is calculated for the current transformation. This value is then optimized by an automatic procedure searching for the best registration transformation.

3.3.3 Multiresolution Pyramid

One of the most common problems in the context of the registration problem are local minima in the similarity measure. These minima "trap" optimization methods and hence the global minimum cannot be achieved. To overcome this difficulty,

[7] A polygon resulting from such intersection can be either a triangle, a quadrangle, a pentagon or a hexagon.

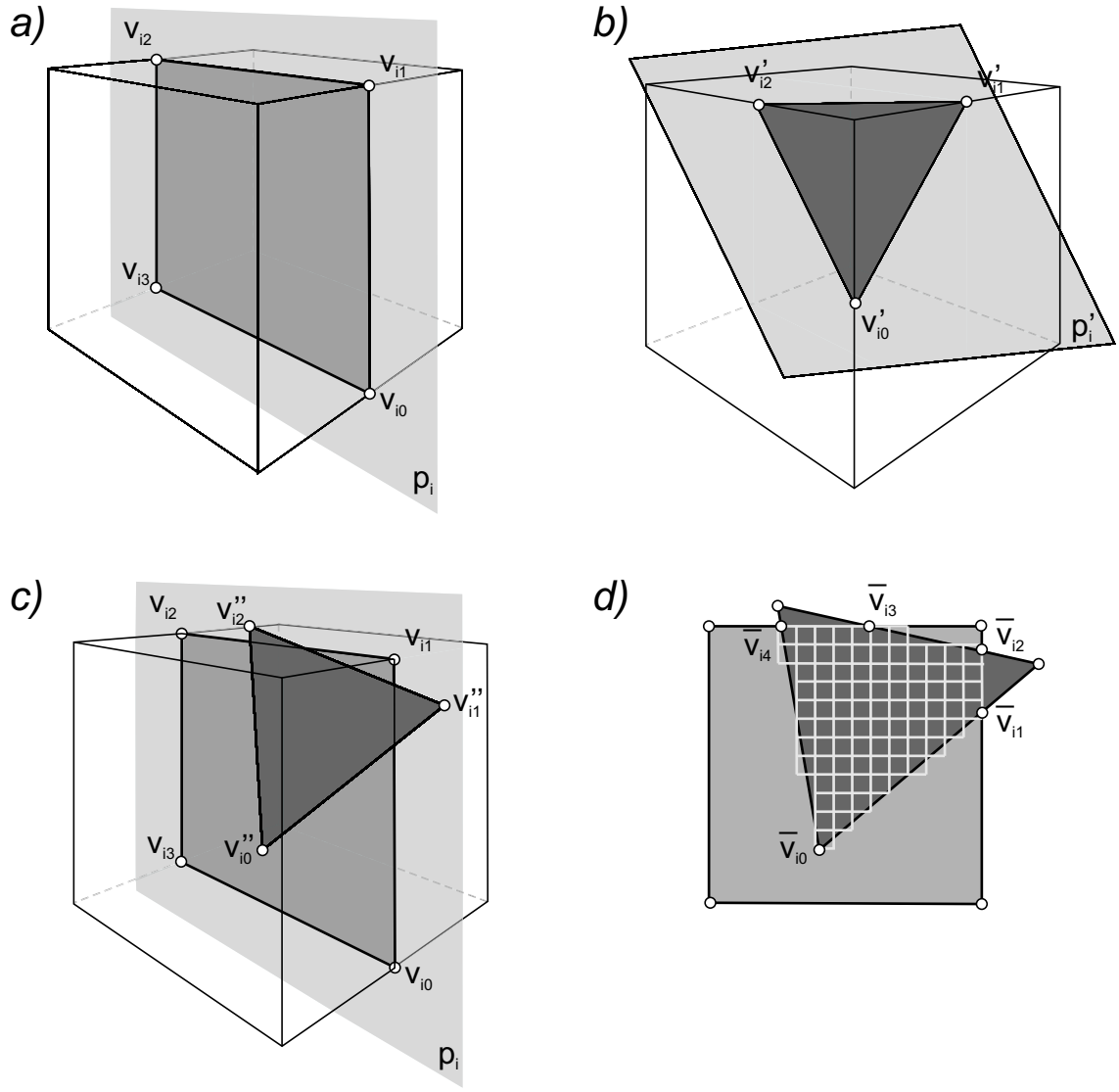

Figure 3.4: Reslicing of the source volume for the calculation of the joint histogram. *a)* Intersection of a plane p_i perpendicular to one of the main coordinate axes with the bounding box of the target dataset. *b)* Intersection of the transformed plane $p_i^{'}$ with the bounding box of the source dataset. *c)* The back-transformed polygon resulting from the intersection. *d)* Parallel projection, clipping and rasterization of the corresponding polygons

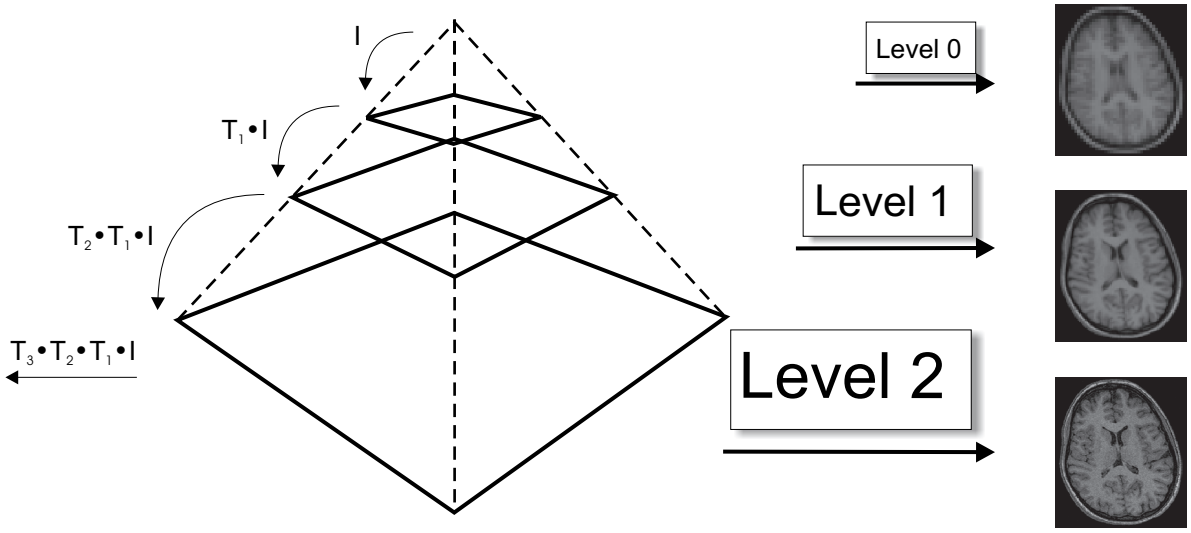

Figure 3.5: Multiresolution pyramid for registration. Although the images are downscaled in image space, the physical dimension of the data remains the same since the voxel size is adapted

different multiresolution frameworks have been proposed and evaluated by various researchers [171, 98, 131]. A similar concept has been incorporated in our registration approach. Within this framework, three resolution levels are considered. For this purpose, a sequence of resampled image pairs $\{(\mathbf{I_s}^l, \mathbf{I_t}^l)\}_{l=0,\ldots,2}$ is created with a resolution increasing progressively with the level number l. For the registration along the pyramid, the images are not transformed from level to level. Rather, the transformations from respective levels are multiplied and applied each time to the original image in order to avoid concatenation of the interpolation errors (see Figure 3.5).

In our implementation, the images are not explicitly downscaled. Instead, a given hierarchy level is created implicitly by rendering slices of the respective images in a viewport of an adapted size. At the hierarchy level $l = 2$, the size of the viewport matches the respective dimensions of the considered image, at the level $l = 1$ and $l = 0$, the size of the viewport is reduced in each dimension by a factor of two and four, respectively.

One of the advantages of the multiresolution pyramid is a smaller computation effort in comparison to methods involving only one, highest resolution. At the lower resolution level, larger steps in the parameter space are considered during optimization, whereas at higher levels a finer search is performed. This results in a significantly lower computation time, as the total number of iterations required to achieve the optimum global value is generally reduced. Moreover, considering the fact that evaluation of the similarity measure is significantly faster at lower resolution levels, this leads to a significantly reduced overall computation expense.

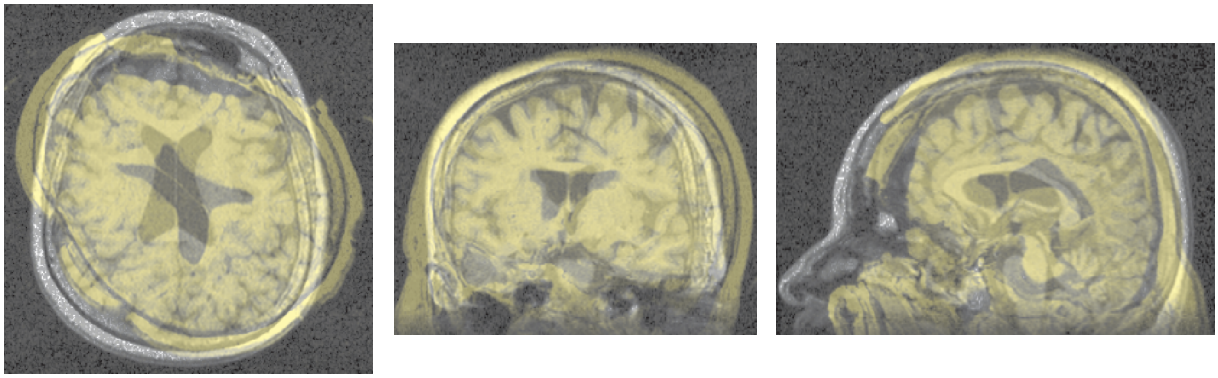

Figure 3.6: Overlayed slice views of pre- and intraoperative head images (Patient 3) before rigid registration. The preoperative image is depicted in gray, the intraoperative data is colored yellow

In addition, optimizing first a subsampled image allows some local minima to be avoided, which would be difficult when working at the highest resolution level only. All these properties prove the multiresolution strategy to be well suited for the registration problem.

3.4 Results

In order to evaluate the efficiency and quality of the described method, a series of registration experiments were performed. For the evaluation, five pairs of MR T1-weighted images were considered where each pair consisted of scans of the head acquired before and during surgery on an open skull at the Department of Neurosurgery at the University of Erlangen-Nuremberg. All the scans were performed with a Siemens Magnetom Sonata Maestro Class 1.5 Tesla scanner and had a resolution of $512 \times 512 \times 160$ voxels by the voxel size of 0.49 mm $\times$ 0.49 mm $\times$ 1.0 mm. In all image pairs, a significant misregistration between the pre- and intraoperative 3D images was observed. Figure 3.6 presents an example of one of the considered pairs of MR scans before the registration. Three slice views of the respective datasets are displayed in one overlayed image with the use of alpha blending and intensity scaling.

In our study, two registrations were conducted for each pair of datasets. First, the unchanged original MR images were considered for registration. Second, a three-stage multiresolution registration was conducted. A viewport of size 128×128 was chosen at the coarsest level, whereas at the finest level the original size of the images (512×512) was chosen. In all experiments, the normalized mutual information was implemented as the similarity measure and Powell's direction search method was employed to optimize the similarity function with respect to the six free parameters of a rigid transformation. In all cases, the intraoperative

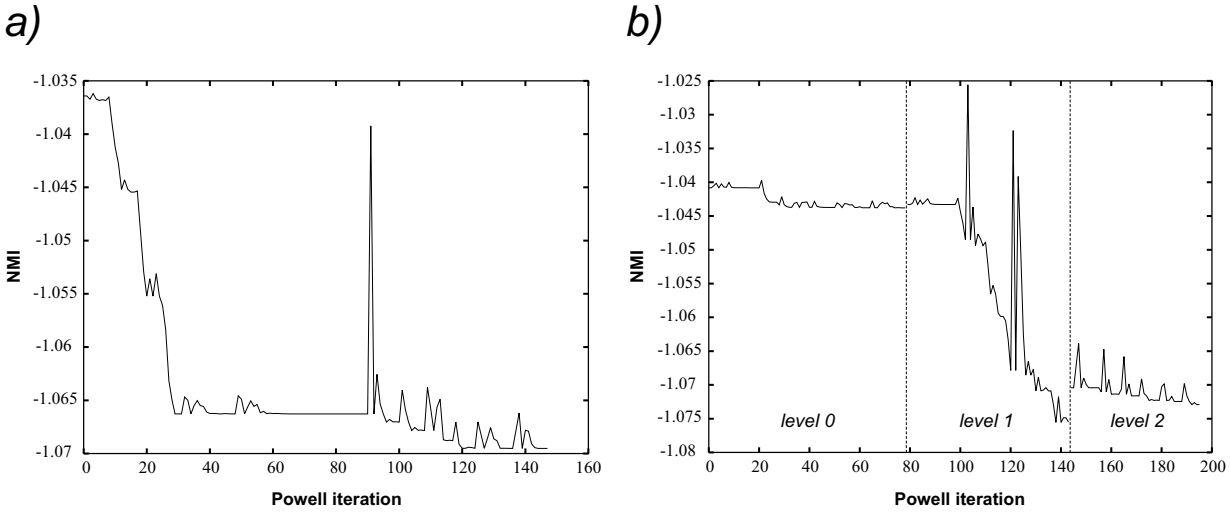

Figure 3.7: Optimization of the normalized mutual information during a rigid registration procedure. *a)* Values of the similarity function for non-hierarchical registration. *b)* Optimization using a hierarchical structure. With this strategy a higher similarity value was achieved. The plots for the consecutive hierarchy levels are separated with vertical dashed lines

dataset was set as the target image. This choice is justified since this is the usual direction of registration in the neurosurgical context, where preoperative data has to be transferred into the intraoperative environment. For both registration procedures, computation times and the convergence behavior were recorded. Figure 3.7 shows the behavior of the similarity measure function (NMI) during the optimization process. On the left, the plot for a non-hierarchical registration experiment is shown. The right-hand plot depicts the progression of the similarity for the corresponding multiresolution procedure.

In Table 3.1, the computation data for all experiments is summarized. It is shown whether hierarchical approach was applied or not and the size of the viewport considered for rendering. Further, the number of steps required for the termination of the optimization, the tolerance parameter TOL used by Powell's optimization algorithm for controlling the stopping criterion and the optimized similarity measure value are presented. Finally, the calculation time for one iteration and the overall time expense are given. All measurements were conducted on a P IV 3.0 GHz system equipped with Nvidia GeForce FX 5950 Ultra 256 MB graphics card running under Suse 8.2 Linux.

Our evaluation of the accuracy of the registration method was carried out based on several specific anatomical regions. In this context, the following six anatomical features were marked with a single point consistently throughout all im-

Table 3.1: Computation times of the rigid registration algorithm. TOL is the parameter used by Powell's algorithm for controlling the precision of calculations

Patient No	Hierarchy	Viewport Size	Steps	TOL	NMI value	1 Iteration Time [s]	Total Time [s]
1	no	512×512	150	10^{-2}	1.069511	1.18	176.58
	level 0	128×128	79	1.0	1.043789	0.11	8.54
	level 1	256×256	65	10^{-1}	1.075551	0.31	20.16
	level 2	512×512	52	10^{-2}	1.072935	1.18	61.38
2	no	512×512	377	10^{-3}	1.078420	1.17	441.21
	level 0	128×128	73	1.0	1.070075	0.10	7.42
	level 1	256×256	51	10^{-1}	1.071050	0.31	15.86
	level 2	512×512	158	10^{-3}	1.087276	1.17	185.07
3	no	512×512	78	10^{-2}	1.051638	1.19	92.9
	level 0	128×128	45	1.0	1.082149	0.10	4.54
	level 1	256×256	42	10^{-1}	1.089333	0.31	13.06
	level 2	512×512	54	10^{-2}	1.089325	1.14	61.41
4	no	512×512	65	10^{-2}	1.108939	1.16	75.52
	level 0	128×128	81	1.0	1.112582	0.10	8.20
	level 1	256×256	68	10^{-1}	1.111716	0.31	21.06
	level 2	512×512	79	10^{-2}	1.108983	1.16	91.71
5	no	512×512	83	10^{-2}	1.090837	1.15	95.58
	level 0	128×128	105	1.0	1.094630	0.10	10.34
	level 1	256×256	85	10^{-1}	1.093854	0.31	26.21
	level 2	512×512	73	10^{-2}	1.090866	1.15	84.14

ages: (P_1) the roof of the fourth ventricle, (P_2) junction between the pons[8] and the midbrain, (P_3,P_4) the occipital[9] horns of the left and right lateral[10] ventricles, respectively, (P_5) the anterior genu part of corpus callosum[11] and (P_6) the culmen of

[8]The pons is a knob on the brain stem relaying sensory information between the cerebellum and cerebrum. Some theories posit that it has a role in dreaming.

[9]Pertaining to the back of the head.

[10]Lying away from the median and sagittal plane of the body.

[11]The corpus callosum is the largest white matter structure in the mammalian brain connecting the left and right cerebral hemispheres. The genu is a kneelike band in the anterior part of the corpus callosum.

Table 3.2: Distances before rigid registration. RMS, VAR and STD denote the root-mean-square error, variance and standard deviation, respectively

Patient No	Point 1	Point 2	Point 3	Point 4	Point 5	Point 6	RMS error	VAR σ^2	STD σ
1	14.311	15.483	17.437	11.868	17.058	14.202	15.178	4.261	2.064
2	7.357	12.675	8.214	3.563	13.965	7.485	9.539	14.647	3.827
3	30.579	28.379	25.869	49.598	38.998	36.559	35.887	75.659	8.698
4	17.881	9.704	9.078	7.644	8.748	9.581	10.978	13.835	3.720
5	28.879	12.965	39.057	33.417	33.428	36.816	31.930	87.980	9.380

the cerebellum[12]. On the one hand, relatively simple and accurate localization in all three dimensions is possible for the chosen points. On the other hand, the choice of the points was motivated by their location, i.e. in the occipital and the median part of the head. This is very important for the assessment method since the anterior brain lobes underwent significant intraoperative deformation which cannot be compensated for with a rigid registration only (see further chapters).

We analyzed the quality of the registration experiments by means of statistical measures. In this context, the distances between marked points in the corresponding datasets were calculated before registration. Based on this information, the *root-mean-square* (RMS) error, the bias-corrected *variance*[13] σ^2 and the *standard deviation* σ were estimated, which are defined as:

$$RMS = \sqrt{\frac{\sum_{i=1}^{N} |P_{i,s}P_{i,t}|^2}{N}} \,, \tag{3.6}$$

$$\sigma^2 = \frac{\sum_{i=1}^{N} (|P_{i,s}P_{i,t}| - \bar{P})^2}{N-1} \,,\ \bar{P} = \frac{1}{N}\sum_{i=1}^{N} |P_{i,s}P_{i,t}| \,, \tag{3.7}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N} (|P_{i,s}P_{i,t}| - \bar{P})^2}{N-1}} \,. \tag{3.8}$$

Here, $N = 6$, $P_{i,s}$ and $P_{i,t}$, $i = 1, \ldots, 6$ denote the points defined in the source and target datasets, respectively. $|P_iP_j|$ stands for the distance between two given points. $\bar{P}$ is the mean value of the distances between corresponding points. The numerical results of this analysis are collected in the Table 3.2.

[12]The cerebellum is located in the occipital region and is responsible for a number of motor and cognitive functions. The culmen is located in the superior part of the cerebellum.

[13]Contrarily, the not corrected sample variance is defined as $\sigma^2 = \frac{\sum_{i=1}^{N} (|P_{i,s}P_{i,t}| - \bar{P})^2}{N}$.

Table 3.3: Point errors after rigid registration. RMS, VAR and STD denote the root-mean-square error, the variance and the standard deviation, respectively. The presented values refer to the hierarchical approach

Patient No	Point 1	Point 2	Point 3	Point 4	Point 5	Point 6	RMS error	VAR σ^2	STD σ
1	1.090	0.899	1.421	1.100	1.900	1.245	1.316	0.124	0.352
2	1.092	2.059	3.304	2.663	0.0	0.941	2.013	1.492	1.221
3	0.691	1.109	0.490	1.465	1.619	0.691	1.095	0.213	0.461
4	1.092	1.777	0.691	0.977	0.488	0.690	1.041	0.211	0.459
5	0.843	0.511	0.0	1.551	0.488	0.0	0.776	0.339	0.583

Subsequently, these statistical terms were calculated after conducting rigid registration with the use of the presented algorithm. In order to obtain the corrected formulae, only points $P_{i,s}$ have to be replaced in Equations 3.6 - 3.8 with their images $P'_{i,s}$ after the calculated transformation. The resulting values of these statistical measures are presented in Table 3.3.

Additionally, a visual assessment of the quality of the registration was carried out in all cases. For this purpose, an interactive technique with the use of a magic lens was employed. In this method, one image is displayed in a window that can be interactively positioned over the second window containing the registered image. An alpha blending factor can be set arbitrarily, enabling a comprehensive visual impression. Additionally, the intraoperative (target) images were overlayed with the contours of the transformed preoperative (source) image. An example of such a visualization is presented in Figure 3.8.

3.5 Discussion

As can be seen from the numerical results in Tables 3.2 and 3.3, the presented multiresolution approach can significantly benefit the registration, both in terms of quality and computation expense. For datasets 1-3, the final value of the similarity measure was significantly higher for the hierarchical algorithm. Only for the datasets 4-5 no real improvement of the alignment quality could be observed. Figure 3.7 gives an example of the differences in the similarity values between the unregistered and registered states. The cracks in the exemplary plot of the similarity function that are observed between the different resolution levels can be explained by the changes in the intensity values from one level to another. These changes influence the shape of the joint histogram of two given images and thus the values of the similarity function even if the underlying transformation is the same.

The analysis of the numerical data was in all cases followed by a visual assess-

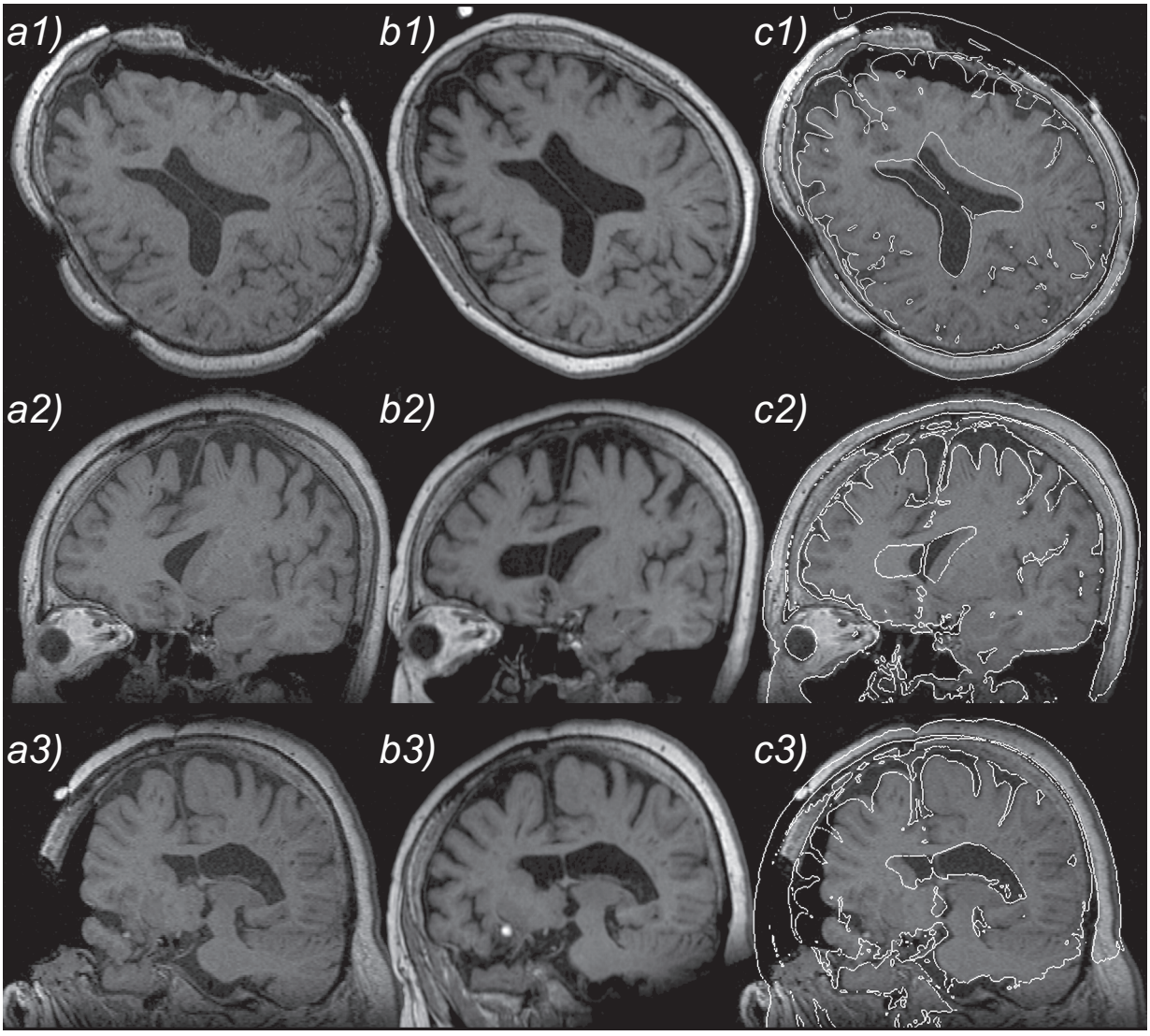

Figure 3.8: Intraoperative images overlayed with the contours of the registered preoperative image (for Patient 3). *Top:* axial, *middle:* sagittal and *bottom:* coronal views. *a)* An intraoperative slice image. *b)* The corresponding slice of rigidly registered preoperative image. *c)* The contours of the image from b) overlayed with the image a)

ment using methods described in the previous section. After this analysis, setting parameter TOL=10^{-2} proved not to be sufficient for a satisfying registration in both experiments with dataset 2. Setting TOL=10^{-3} resulted in a visually good registration and at the same time in higher computation times. However, the times were still acceptable, for the hierarchical case (the overall time, the sum of all resolution levels, was 208.35 s, in comparison to 441.21 s for the case without the hierarchical strategy).

A significant time benefit between the hierarchical and non-hierarchical algorithm was exhibited for datasets 1-3. The improvement in computation expense correlated interestingly with the improvement of the quality. This can be explained by getting trapped in a local minimum in the one-stage optimization process, a problem that is overcome in the three-level approach. Occurrence of these minima influenced, on the one hand, the quality of the one-level experiment result, and on the other hand trapped the optimization process locally (see the middle part of the left plot in Figure 3.7), thus resulting in a higher number of iterations.

Analyzing the statistical point errors before and after registration (see Tables 3.2 and 3.3), we can see that the biggest misalignment before registration took place for Patient 3. This was then compensated for by the registration experiment. In contrast, for Patient 2, where the misalignment before the registration was lowest, the optimized result was the worse among all pairs, despite the adaption of the tolerance parameter for the optimization. This reinforces the assumption for occurrence of strong local minima preventing the approaching of the true optimum.

Altogether, a good quality of the procedure was observed. The RMS error measured in the experiments was between 0.776 mm and 2.013 mm. This means that in the best case the misregistration was less than 2 voxel in the x- and y- direction, and less than 1 voxel in the z-direction. Considering pre- and intraoperative head images, although pairs of images from the same patient were analyzed during the experiments, the observed brain shift introduced additional changes between the images making the registration problem significantly more difficult. Moreover, it is well known that the brain pulsates within the skull by about 0.5 cm, even in a closed skull. Taking into account these facts and the resolution of the images the results of the experiments are very satisfactory.

Visual validation with a magic lens and contour images (see Figure 3.8) confirms these observations. Though the images differ strongly in the region of the anterior brain lobe and the anterior part of the ventricles due to brain shift, a very good alignment of the images can be seen in the back part of the head (Figure 3.8c1) and c3) and in lateral ventricles where no soft tissue deformation occurred. Also, the skull and the surrounding skin which had not changed its shape (apart from the region of its opening) were registered with a good precision, despite their inconsistent intensity representation in an MR scan.

3.6 Conclusion

In this chapter, a hardware-accelerated algorithm for fast rigid registration has been presented. The registration process has been treated as an optimization problem. During the optimization, an image similarity metric, the normalized mutual information, is maximized with time based on an underlying hierarchical pyramid of images. Since the normalized mutual information is based on voxel intensities only, no user interaction is required during this process. On the other hand, the algorithm implementation is flexible enough to allow user intervention if desired. Thus, the number of hierarchy levels and precision of the calculations can be set or *a priori* knowledge can be incorporated in the form of an initial alignment matrix.

This approach utilizes texture mapping of viewport-aligned slices for reinterpolation of a transformed volume. This makes the strategy similar to hardware-accelerated volume rendering techniques. The benefit of the method is a reduction of computational complexity[14], which is achieved by hardware acceleration of the trilinear interpolation that is the most expensive computational part of each registration procedure.

A comprehensive evaluation of the proposed algorithm in a series of experiments with MR images showed its practical value. In the evaluation, misaligned pre- and intraoperative head images were considered. Defining landmarks in parts of the images not affected by brain shift allowed a reliable quality assessment. The choice of the datasets was motivated by the fact that the rigid registration algorithm will be a prerequisite for a non-linear registration algorithm applied to pre- and intraoperative brain images presented in the next chapters. In this context, evaluation of the initial rigid registration step is important before the quality of the non-linear alignment can be assessed.

In addition to the results from the previous section, successful experiments with the registration of multimodal images were performed underlining the flexibility of this method. The achieved results are, however, outside the main scope of this thesis. For an example of a multimodal fusion based on the results of the presented algorithm see Figure 2.1.

Despite a good time behavior of the method, further reduction of the computation time is possible. The main part of the calculation time is caused by reading the frame buffer content. This operation takes an average of 5 ms per frame and for the whole volume about 800 ms (using the viewport size 512×512), which amounts to 68 % of the total computation time. With the new generation of graphics cards, elimination of this time expense will be possible. This can be achieved with the use of *superbuffers* introduced in the newest ATI graphics adapters.

[14]The computational expense is meant here in terms of the optimization process where a number of reinterpolation steps are required.

Chapter 4

Hardware-based Non-Rigid Registration

The previous chapter focused on rigid registration with a strong emphasis on the medical implementation. As discussed in Chapter 2, there is a large demand for non-linear image registration algorithms in the medical field. This demand has continued growing in the last years, which can be attributed to the fact that only recently the computation power of computers has enabled performance of complex calculations in reasonable times. In particular, registration algorithms concerned with soft tissue deformation have attracted considerable attention in recent years [142, 57, 190]. Thereby, one of the main concerns is the large amount of time required for performing the computations. Although the running times of the existing non-linear registration algorithms are acceptable in the sense of their feasibility for selected clinical problems and for research work, they are still not acceptable for clinical applications that require response within shortest possible times (e.g. systems applied in intraoperative environment).

In the context of neurosurgery, navigation systems are an example where application of time efficient non-linear registration algorithms would benefit the surgery procedure. Preoperative tomographic data of the patient's head is used by the surgeon in order to familiarize himself with the patient-specific anatomy and to perform preprocessing (e.g. segmentation of tumors) and planning steps for the surgery. However, the validity of the preoperative preparations decreases rapidly with ongoing surgery due to the changes in the soft tissue (*brain shift*) resulting from craniotomy and other surgical and physical events.

Therefore, there is a need for the introduction of computationally efficient and at the same time accurate methods for non-linear volumetric deformation and for subsequent non-linear registration. These methods can then be used to perform intrasurgical updates of the preoperative planning imagery. The high computation costs of existing approaches are one of the main reasons why commercial neuronavigation systems still do not contain such features.

In this chapter, a powerful hardware-based technique is presented with which it

is feasible to register volumetric 3D volume images in a non-linear fashion. Acceleration of calculations is achieved with mechanisms provided by graphics adapters. Related work concerning volume deformation in the context of computer graphics is reviewed in Section 4.1. In Section 4.2, a 3D piecewise linear model used for volumetric deformation is presented and its relevant geometrical properties are addressed. Thereby, hardware implementation issues, a registration algorithm based on the introduced model and clinical applications of the developed system are discussed. Section 4.3 introduces an extension to the constructed registration framework implementing 3D Bézier functions with the use of graphics hardware. The deformation model, its implementation and clinical applications are presented. Finally, this chapter is concluded in Section 4.4.

4.1 Previous Work

A comprehensive overview of the existing methods for non-linear medical image registration has been presented in Chapter 2. Independent of the mathematical method used for registration, most of the existing algorithms perform a deformation of a registered volume numerous times during some optimization procedure. Techniques for volumetric deformation have been investigated in research works in a wider context, not only for registration purposes. In computer graphics, there a number of different approaches for efficient 3D deformation have been developed. Since the method for non-linear registration presented in this chapter is based on graphics hardware-based volume deformation, it is important to review these approaches here.

Free-form deformation (FFD) [150] has been one of the most powerful techniques for object modeling. This approach combines free-form surfaces and solid modeling into a common framework for deforming solid geometry and surfaces. This deformation concept inspired further research in this area, which has resulted in various extensions of the basic FFD algorithm. An extension of this work allowing the design of arbitrarily shaped bumps and surface bending along arbitrarily shaped curves was presented in [41]. Volume preserving techniques based on FFD and use of level-of-detail representations was described by Hirota *et al.* [78]. In this approach, new positions of the deformation lattice are calculated by minimizing the elastic energy of deformation. In [61], a volumetric skeleton is computed from a dataset with a thinning procedure. This extraction is reversible, which means that reconstruction of the volume from the skeleton is possible. With this technique, animation of volumetric objects is possible within standard graphics animation software.

For the purpose of volume deformation, hardware-based implementations have also been presented. An approach employing OpenGL for accelerated computation of FFD was described in [36]. However, deformed positions for every object point were explicitly calculated there, which introduces an additional time effort. A modified ray casting algorithm was introduced by Kurzion *et al.* [88]. In that work, a ray

deflector algorithm was described where instead of changing the geometry of an object, the viewing rays were bend to achieve a deformed view of a volume. This idea was then further adapted for 3D texture-based volume rendering overcoming previous limitations related to modeling intersecting objects [89]. Within this modified approach, additional tessellation of the polygons was conducted before rendering in order to account for interior deformation in a volume.

A hierarchical structure for hardware-accelerated volume deformation was proposed by Fang *et al.* [56]. This implementation was based on 3D textures. Within that framework, an octree structure was constructed dividing the volume into smaller subcubes. Based on this representation, slicing and subsequent texture mapping could be performed in a more efficient way compared to non-hierarchical approaches. Westermann *et al.* [194] developed a mechanism whereby the shape of existing objects represented by a triangle mesh could be manipulated. The changes in the shape achieved by shifting the vertices of the triangle mesh were followed by the changes in the object appearance which was defined in the form of a 3D texture. In order to correctly deform the inner structures, appropriate texture coordinates had to be adapted. For the purpose of 3D rendering, an efficient method for calculation of intersections between slicing planes and the triangle mesh was introduced.

OpenGL Volumizer [53] is a library of C++ classes which was developed to facilitate display and manipulation of volumetric datasets. Tetrahedra were taken as volumetric primitives. For the purpose of deformation, an affine transformation was calculated for each tetrahedron according to the displacements specified in the tetrahedron vertices. This mapping was then used for volume deformation. This interesting approach, however, was characterized by performance problems due to the expensive depth sorting procedures and intersection calculations between the tetrahedral geometry and the slices defined for rendering.

4.2 3D Piecewise Linear Deformation Model

In this section, an efficient volumetric deformation model is detailed. A mathematical description of the model and the implementation issues are addressed. This model is based on a hierarchical hexahedra structure which has been especially designed to exploit hardware-accelerated texturing features of graphics adapters.

4.2.1 Deformation in Texture Space

Conceptually, the approach of volumetric deformation introduced here can be considered as related to the free-form deformation (FFD) approaches [14, 150], though the underlying mathematical description is completely different. In the model presented in this thesis the geometry of the volume is subdivided into smaller *patches* having a cubic form (Figure 4.1a).

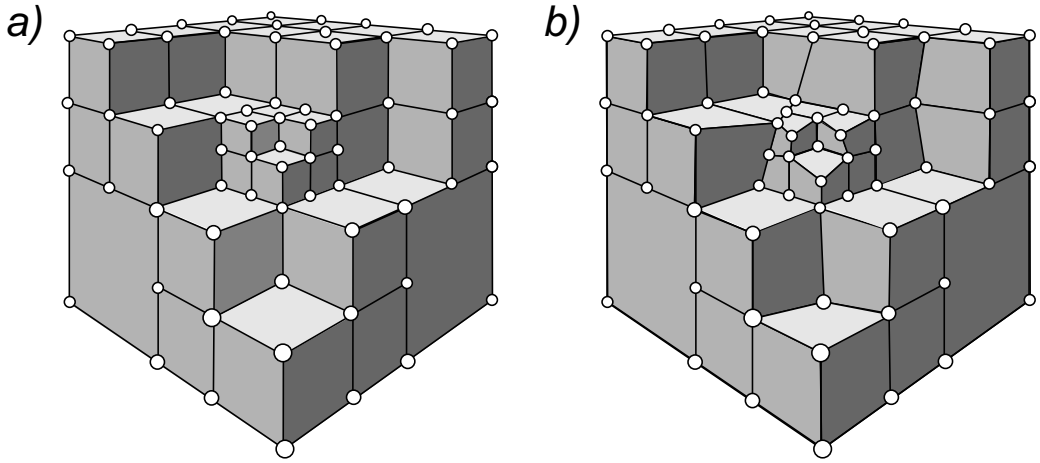

Figure 4.1: The geometry and the texture space of the piecewise linear model. *a)* The geometry is static and consists of cubes. *b)* The deformation is modeled in the texture space by moving texture coordinates

The idea behind the deformation is to assign a 3D texture to the underlying cubic structure and perform texture warping only, while the geometry remains static (Figure 4.1b). This allows exploitation of the hardware-accelerated texturing while at the same time avoiding complicated calculations which would have been performed for changing geometry.

In order to precisely describe the deformation approach, the definitions of some basic notions are necessary. Let us associate the object space $\mathcal{OS}$ with the object undergoing deformation. The function $\mathbf{T} : \mathcal{OS} \mapsto \mathcal{TS}$ is considered to lead from the object space to the texture space $\mathcal{TS}$ being $[0,1]^3$. This function defines the correspondence between the geometry and the texturing in the undeformed state, whereas the function $\mathbf{H} : \mathcal{OS} \mapsto \mathcal{TS}$ describes the texture mapping after deformation and is expressed in the form:

$$\mathbf{H}(x,y,z) = \mathbf{T}(x,y,z) + \sum_{i=0}^{7} w_i(x,y,z) \cdot t_i \ . \tag{4.1}$$

The coefficients $t_i \ \ (i = 0,\ldots,7)$ refer to the offset vectors of texture coordinates specified before and after deformation at the vertices of the patch including the points (x,y,z). Additionally, w_i with $i = 0,\ldots,7$ are the corresponding trilinear interpolation weights associated with the points (x,y,z) with respect to the geometry (thus, in the unchanged state).

For the purpose of deformation, the translations t_i of the texture coordinates have to be specified for the corner vertices of the cube structure. To calculate the deformation at an arbitrary point (x,y,z) lying inside the volume, the corresponding

patch containing this point has to be first considered[1]. Based on the values of t_i at the patch vertices and using the definition of function H from Equation 4.1, the new texture information can be assigned to any given point inside the patch.

With respect to the performance, an important feature of this method is that the object geometry remains unchanged during the deformation process and only the texture coordinates are shifted. Thus, the interpolation coefficients w_i can be computed prior to any deformation. Furthermore, in the context of slice-based volume rendering, this means that the intersections between the cutting planes and the geometry must only be calculated once, which is an enormous benefit if deformation is performed a number of times (e.g. in an iterative process as in registration). Additionally, due to the simple geometry, the calculation of these intersections can be implemented very efficiently.

4.2.2 Deformation in Geometry Space

Volumetric deformations can be calculated in automated applications with the technique described in the previous section very conveniently. However, this does not ensure a comfortable way to manually accomplish changes in volume. To make manipulation more intuitive, object manipulation in the geometry space, rather than in the texture space, must be provided. For this purpose, the inverse transformation $\mathbf{H}^{-1}$ of the deformation function has to be computed. The inverse function, however, is more complex than trilinear mapping. Such functions cannot be efficiently implemented with the use of graphics hardware. This excludes the direct evaluation of $\mathbf{H}^{-1}$ within the presented deformation framework. Instead, the translations t_i of the texture coordinates are negated and an approximation of $\mathbf{H}^{-1}$ is calculated with:

$$\mathbf{H}^{-1}_{app}(x, y, z) = \mathbf{T}(x, y, z) + \sum_{i=0}^{7} w_i(x, y, z) \cdot (-t_i) \ . \tag{4.2}$$

This yields a sufficiently well approximation of the inverse function $\mathbf{H}^{-1}$. More precisely, for a deformation d, the approximation error can be estimated by

$$\mathbf{H}^{-1}_{app}(\mathbf{H}(x, y, z)) = \mathbf{T}(x, y, z) + o(d^2) \ . \tag{4.3}$$

4.2.3 Hierarchical Octree Structure

To model a complex local deformation while avoiding unnecessary uniform subdivision of the volume, a hierarchical octree structure is introduced in this section. With this concept, an adaptive subdivision of the volume can be conducted with respect to some arbitrarily defined criterion.

[1]Due to our particular implementation it is not necessary to establish this correspondence explicitly, thus no additional computation time is required for this step. More technical details related to the implementation of the algorithm are depicted in Section 4.2.4.

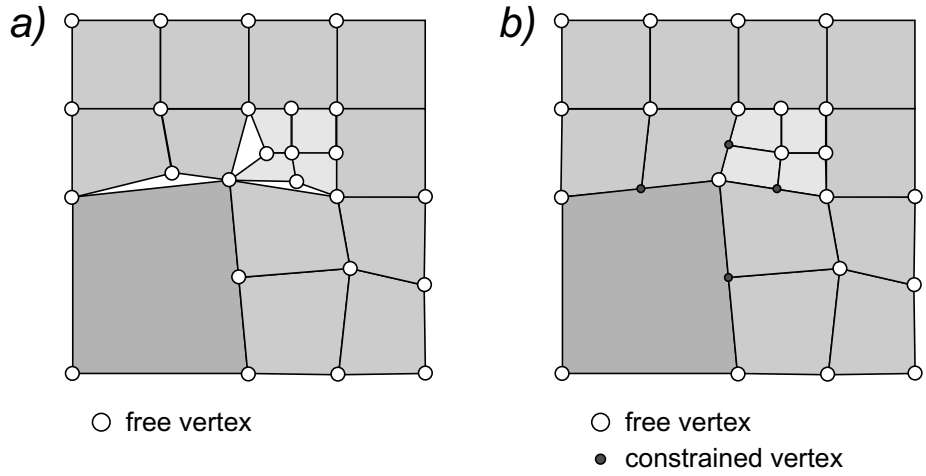

Figure 4.2: Texture mapping corresponding to an exemplary slice through the hierarchical octree structure. *a)* Texture artifacts occur if no additional conditions on vertices are required. *b)* Introducing constraints on the vertices can prevent gaps in the texture space

In Section 4.2.5, a detailed description of the application of this framework for hierarchical image registration is provided. In comparison to the uniform subdivision, this allows significant reduction of the number of free vertices involved in the optimization process.

For adaptive subdivision, however, additional conditions have to be introduced into the 3D piecewise linear model. Without defining these constraints, occurrence of holes in the texture space is possible during texture mapping which is a visual disturbance and injures the physics of the deformed object (Figure 4.2).

Two types of constraints can prevent these undesirable effects:

Collinearity of Edges: Let us consider two neighboring patches belonging to different octree subdivision levels. Such patches share a common edge. The vertex lying on the subdivided edge e has to be provided with a constraint which ensures that the vertex stays on the line between the end vertices of the edge (Figure 4.3a):

$$v^e_{inner} = c \cdot v^e_0 + (1 - c) \cdot v^e_1 \; . \tag{4.4}$$

This means that collinearity of the considered edge is preserved.

Coplanarity of Faces: Similar to the edge constraints, it has to be ensured that the faces of neighboring patches remain coplanar. This condition can be violated only if the neighboring faces belong to different subdivision levels. To prevent a breach of coplanarity, the inner vertex of a given face has to fulfill

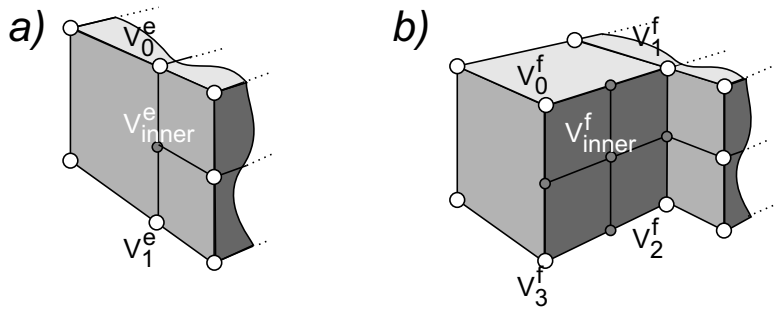

Figure 4.3: Constraints in the texture space. *a)* Constraint on edges. *b)* Constraints on faces

the condition:

$$v^f_{inner} = \sum_{i=0}^{3} c_i \cdot v^f_i \ , \tag{4.5}$$

with $v^f_0, \ldots, v^f_3$ being the corner vertices of the considered face and $c_i, i = 0, \ldots, 3$ summing up to unity. An example for this case is shown in Figure 4.3b).

In order to reduce computational cost for handling recursive dependencies, an additional constraint is introduced which ensures that a patch can only be subdivided if all surrounding patches have at least the same subdivision level.

4.2.4 Implementation Issues

As already mentioned, the presented deformation model makes use of graphics hardware acceleration techniques. Thus, OpenGL has been applied to implement the texture mapping part of the framework. In order to render the deformed volume after shifting the texture coordinates, the object is considered to consist of a set of slices parallel to one of the main axes. This set of slices has to be analyzed for intersection with the hierarchical octree geometry. The result of the intersection is a set of quadrilateral structures (see Figure 4.4). Since the geometry remains static, this calculation can be performed in a preprocessing step and the result can be reused in subsequent deformations[2].

The number of resulting slices is equal to the resolution of the image volume in the direction perpendicular to the slices. For each slice the corresponding deformed slice in the texture space is determined. Such a deformed slice usually consists of non-planar quadrilaterals. The vertices of these polygons result from the linear interpolation of the deformed texture coordinates assigned to the corresponding vertices in the octree geometry.

[2]Only if a new hierarchy level is introduced into the structure does the intersection have to be recomputed.

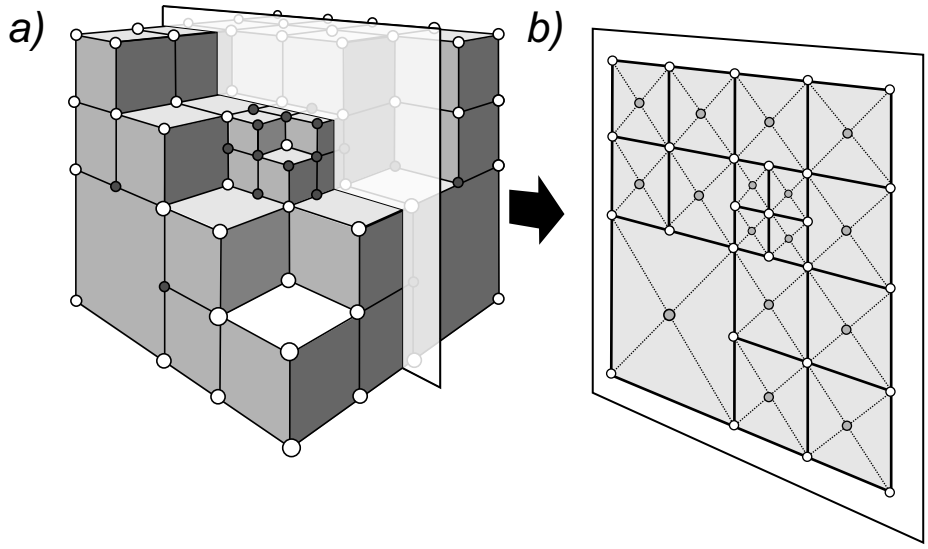

Figure 4.4: Texture mapping framework: *a)* The hierarchical octree geometry is sliced with planes parallel to one of the main axes. *b)* Additional vertices (in grey) are introduced into the resulting structure of quadrilaterals in order to ensure correct texture mapping after deformation

A simple approach to rendering such quadrilateral structures would be to assign the interpolated texture coordinates to the individual polygon vertices and to perform texture mapping. However, this would not lead to a correct deformation as described by Equation 4.1. This is due to the method used by OpenGL for automatical tessellation of polygons (see Figure 4.5). In this figure, the first row depicts the correct texturing of an exemplary polygon. In Figure 4.5b), incorrect texture mapping after OpenGL tessellation is shown where the appearance of the grey triangle did not change after deformation. The last row presents a solution, where an additional vertex is inserted into the interior of the rendered polygon dividing this polygon into four triangles. After texture mapping, the resulting approximation of trilinear interpolation is sufficiently close to the correct version presented in Figure 4.5a).

Using such explicit triangulation polygons are then textured and rendered into the frame buffer. The image information is recalculated for the entire volume after deformation on a slice by slice basis with the use of graphics subsystem.

4.2.5 Registration Algorithm

The presented approach for volumetric deformation can be applied for registration purposes. Considering the registration framework of two 3D images, one of them is set as a target and remains fixed during registration. The other one, referred to as the source image, is deformed by changing the texture coordinates of the previously

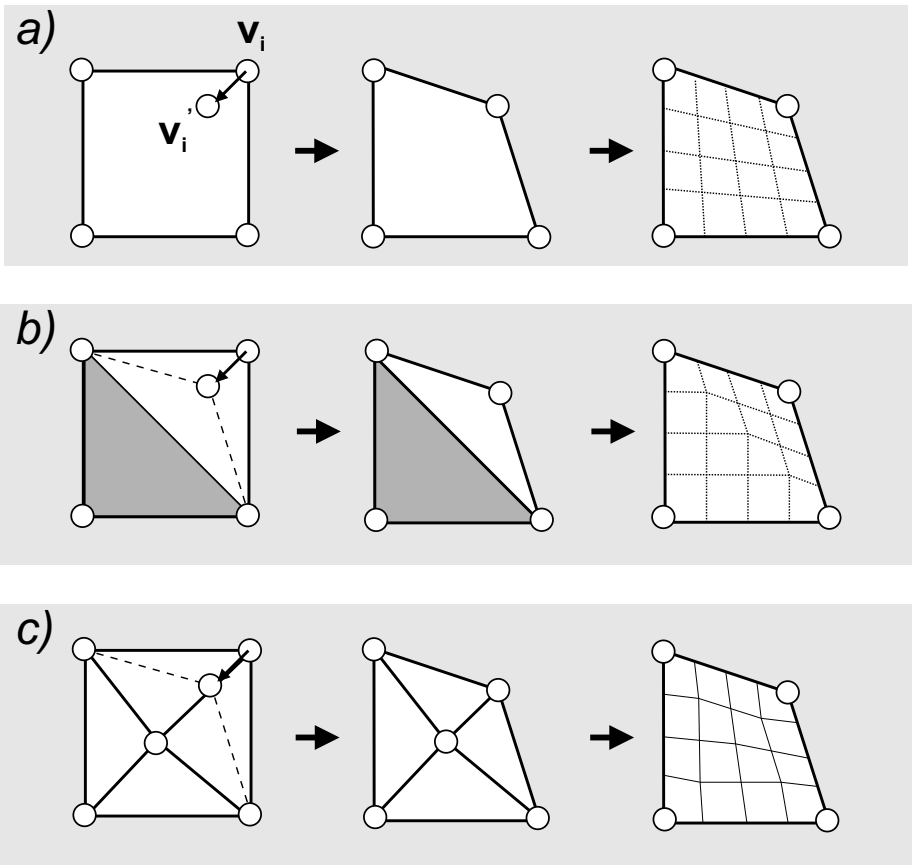

Figure 4.5: Explicit tessellation of quadrilaterals for the correction of texture mapping. *a)* Correct texture mapping. *b)* Incorrect texture mapping after OpenGL tessellation. *c)* Inserting an additional vertex leads to a sufficiently good approximation of the correct mapping

assigned 3D piecewise linear structure so that it optimally aligns with the target image. In our framework, the normalized mutual information is employed as a similarity measure which judges the quality of alignment.

Initially, a linear registration of the images is conducted with the algorithm presented in Chapter 3. This compensates for rigid-body motion. Then, the source image is subdivided in the texture space into a fixed number of 3D piecewise linear patches. The vertices at the volume boundary are set as *inactive* since deformations in the tomographic data acquired in a clinical practice are restricted to the central regions. In contrast, the positions of the inner vertices are free to change during the optimization of the similarity measure and are thus marked as *active*.

4.2.6 Optimization

In order to capture the non-linear deformation, at least three subdivision steps of the source image are required. This results in a large number of free vertices for the optimization. A possible optimizer is Powell's direction search method which has been already applied in the rigid registration procedure. The performance of this method, however, depends linearly on the number of optimized parameters. Three subdivision steps performed on the source image lead to 1029 degrees of freedom. This would result in significantly high computation times if Powell's optimization algorithm were employed.

Instead, another optimization strategy is followed to solve the non-linear registration problem. The *simultaneous perturbation stochastic approximation* (SPSA) [166] method for multivariate optimization problem is used. Let us denote the vectors of the translations t_i of the texture coordinates corresponding to the respective active vertices in the hexahedra structure by $\boldsymbol{t_k}$. At iteration $k+1$, the SPSA procedure updates these free parameters according to:

$$\boldsymbol{t_{k+1}} = \boldsymbol{t_k} - a_k \boldsymbol{g_k}(\boldsymbol{t_k}) \; . \tag{4.6}$$

Here, a_k is a scalar sequence defined later in Equation 4.8 and $\mathrm{g}_k(\boldsymbol{t_k})$ is the estimate of the gradient of the optimized NMI function for the parameter vector $\boldsymbol{t_k}$.

The gradient approximation involves two measurements of the NMI at the perturbed parameter vector $\boldsymbol{t_k}$. Specifically, the component i of this gradient is given by:

$$g_k^i(\boldsymbol{t_k}) = \frac{\boldsymbol{NMI}(\boldsymbol{t_k} + c_k \boldsymbol{\Delta_k}) - \boldsymbol{NMI}(\boldsymbol{t_k} - c_k \boldsymbol{\Delta_k})}{2 c_k \Delta_k^i} \; . \tag{4.7}$$

In this equation, $\Delta_k^i, \; i = 1, \ldots, n$ are elements of user-specified perturbation vector $\boldsymbol{\Delta}$. For this perturbation a Bernoulli distribution is used consisting of $+1$ and -1 both being generated with the same probability of 0.5, as suggested in [166]. a_k and c_k are sequences depending on the constants a, A, c, α and γ and defined as:

$$a_k = \frac{a}{(A+k)^\alpha} \; , \tag{4.8}$$

$$c_k = \frac{c}{k^\gamma} \; . \tag{4.9}$$

Additionally, these sequences have to fulfill the following conditions to ensure convergence of the iterative procedure:

$$\alpha - 2\gamma > 0, \; 3\gamma - \frac{\alpha}{2} \geq 0, \; 0 < \gamma < \alpha < 1 \; , \tag{4.10}$$

with $a > 0, \;\; c > 0, \;\; A \geq 0$. The essential strength of this iterative method is the underlying gradient approximation which requires only two measurements of the objective function independent of the dimension of the parameter domain. Contrary

to other methods where only one parameter at a time is changed and the corresponding partial derivative is calculated, in this approach all optimized parameters are simultaneously perturbed. This allows a significant reduction in the optimization cost.

Moreover, for estimation of the gradients direct measurements of the function are taken and no explicit gradient information has to be provided to the algorithm. This enables efficient application of this technique to the optimization of the normalized mutual information which is dependent on a discrete histogram. It thus has an analytically and computationally complex derivative. Since the optimal parameters do not necessarily have to be obtained in the last iteration of SPSA procedure, an additional third function measurement can be optionally conducted in each iteration to store the current optimal parameters. However, choosing appropriate configuration parameters for the registration problem results in an almost certain convergence. Therefore, after initial experiments we perform only two evaluations of the NMI at each iteration which further reduces the computational expense.

4.2.7 Multiresolution approach

Similar to the approach presented in Chapter 3, three resolution levels $\{({\mathbf{I_s}}^l, {\mathbf{I_t}}^l)\}_{l=0,\ldots,2}$ are considered in the non-linear registration procedure. This hierarchy is created by adapting the size of the rendering viewport to the level number. In comparison to the one-level approach, this technique showed a better quality of registration and at the same time an improved performance.

4.2.8 Results

The algorithm for non-linear registration was evaluated with pairs of 1.5 Tesla pre- and intraoperative MR data acquired with a T1-weighted gradient echo sequence. In this context, five pairs were considered which were identical to those used for the quality assessment of the rigid registration approach. In all cases, a significant soft tissue deformation occurred. Before non-linear registration, the considered pairs of datasets were rigidly aligned by the algorithm described in Chapter 3. In a further preprocessing step, the brains were extracted from the scans with the use of Brain Extraction Tool (BET) software [155] from the FMRIB Software Library (FSL). This segmentation is very important since otherwise already well registered rigid body structures (e.g. skull bone) would be considered for deformation by the intensity-based only non-linear algorithm, which is undesirable.

For each pair of brain images, non-linear registration was then conducted. Analogous to the experiments with the rigid registration algorithm, normalized mutual information was chosen as the similarity measure. In contrast to the rigid alignment, the SPSA procedure was employed in the optimization. Furthermore, in all experiments the intraoperative image was set as the target since this corresponds

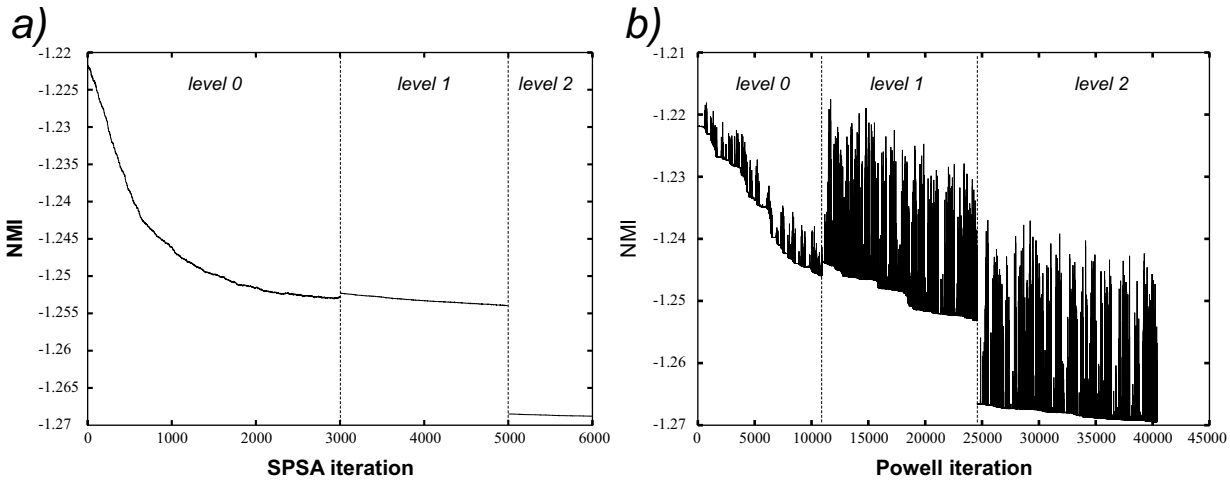

Figure 4.6: Comparison of the *a)* SPSA and *b)* Powell optimization runs. The plots show the value of the similarity function in dependence on the iteration number. Different hierarchy levels are separated with dashed vertical lines

to the clinical situation where preoperative information must be transferred into the intraoperative setting.

During the optimization process, the free parameters are the texture coordinates associated with the free vertices in the 3D hierarchical octree model. For each pair of images the registration was performed with a three-stage multiresolution approach run in a viewport of size 128×128, 256×256 and finally 512×512. According to the suggestions in [167], the configuration parameters for the SPSA procedure were set to be $\alpha = 0.101$ and $\gamma = 0.606$. Furthermore, our own investigations resulted in values of 100.0, 10.0 and 0.5 for the constants A, a and c, respectively.

For all experiments the computation times and information about convergence of the optimization procedure were recorded. The data collected from all the experiments is presented in Table 4.1. Additionally, for the data from the first patient a multiresolution registration was also performed with Powell's optimization method (using the values of 1.0, 10^{-1} and 10^{-2} for the TOL parameter at the hierarchy levels 0, 1 and 2, respectively). In Table 4.2, the respective computation times and the number of iterations required are compared to the performance of the SPSA algorithm. Supplementary plots showing the optimized NMI value in dependence on the iteration number are presented in Figure 4.6 for both optimization runs. The measurements were carried out using a P IV 3.0 GHz system equipped with Nvidia GeForce FX 6800 Ultra 256 MB graphics card running under Suse 8.2 Linux.

Subsequently, the quality of the non-linear registration method was statistically assessed based on measurements of the registration accuracy for some characteristic anatomical regions. For this purpose, six anatomical features were marked in the rigidly registered pre- and intraoperative images and in the preoperative image after non-linear registration. All the features were located in areas where significant

Table 4.1: Computation data of the 3D piecewise linear non-rigid registration algorithm. For each case the information about the hierarchy level, the corresponding viewport size and the achieved optimal value of NMI are specified. Additionally, the calculation time for one NMI evaluation (in one SPSA iteration two such evaluations must be performed) and the total registration time are shown

Patient No	Hierarchy	Viewport Size	Optimal NMI value	1 NMI evaluation Time [s]	Total Time [min]
1	level 0	128×128	1.24597	0.09	9.08
	level 1	256×256	1.25319	0.25	16.81
	level 2	512×512	1.26692	0.92	30.74
2	level 0	128×128	1.27941	0.09	9.12
	level 1	256×256	1.28774	0.25	16.92
	level 2	512×512	1.28817	0.92	30.68
3	level 0	128×128	1.26805	0.09	9.09
	level 1	256×256	1.27897	0.26	17.42
	level 2	512×512	1.28144	0.92	30.54
4	level 0	128×128	1.29208	0.09	9.17
	level 1	256×256	1.29743	0.25	16.80
	level 2	512×512	1.29803	0.94	31.44
5	level 0	128×128	1.27230	0.09	9.16
	level 1	256×256	1.28069	0.25	16.90
	level 2	512×512	1.28136	0.93	30.97

Table 4.2: Comparison of the performance of the SPSA and the Powell algorithm for non-linear registration. For both optimization techniques the number of iterations at a given hierarchy level, the overall number of iterations, the total computation time and the final NMI value are presented

	Iterations			Total number	Total	NMI
	level 0	level 1	level 2	of iterations	Time[min]	value
Powell	10920	13623	15833	40376	316.41	1.269525
SPSA	3000	2000	1000	6000	55.88	1.26922

Table 4.3: Point errors before piecewise linear non-rigid registration. RMS, VAR and STD denote the root-mean-square error, variance and standard deviation, respectively

Patient No	Point 1	Point 2	Point 3	Point 4	Point 5	Point 6	RMS error	VAR σ^2	STD σ
1	4.838	3.594	1.852	4.093	8.111	6.143	5.164	4.681	2.164
2	9.380	5.261	6.727	5.961	13.274	8.199	8.562	8.587	2.930
3	8.343	7.954	6.850	6.083	12.710	11.268	9.177	6.692	2.587
4	5.874	2.693	1.803	2.291	8.851	4.900	5.043	7.268	2.696
5	6.103	2.550	3.082	1.414	5.052	12.347	6.235	15.539	3.942

tissue shift occurred between the pre- and the intraoperative scans, which makes them relevant for the assessment of the algorithm. In this context, the following areas were considered: (P_1) the frontal lateral horn of one of the ventricles[3], (P_2,P_3) two points on the body[4] of the same shifted ventricle, (P_4) the junction between the longitudinal cerebral fissure[5] and the genu part of corpus callosum, (P_5) a significant point on the deformed part of the cortex and finally (P_6) a characteristic point on a sulcus lying next to the tumor border.

The calculated distances between the corresponding points after rigid registration are presented in Table 4.3. The values of the root-mean-square (RMS) error, the variance (VAR) and the standard deviation (STD) are also included in this table giving an idea of the magnitude of the deformation. The respective distances and the statistical measures calculated after non-linear registration are shown in Table 4.4.

In addition to the measurements a visual inspection of the registration quality was conducted retrospectively for all experiments. Figure 4.7 is a visualization of the registration results for Patient 1. Two-dimensional slices in the 3D data of the non-linearly deformed scan are compared with the corresponding slices through the intraoperative MR data to which the preoperative image was matched. These respective slices from the original preoperative MR data are also shown.

Difference images between the registered preoperative image and the original intraoperative data were also created and analyzed. An example of this visualization technique employed for assessment of the quality of registration for Patient 1 is presented in Figure 4.8.

[3]For a given case, either the left or right ventricle was chosen depending on which underwent the most significant change of shape due to brain shift.

[4]The central part (*pars centralis*) of the lateral ventricle extending from the splenium of the corpus callosum to the interventricular passage (*foramen*) connecting the lateral and the third ventricle.

[5]The deep fold between the cerebral hemispheres extending inferiorly to the corpus callosum.

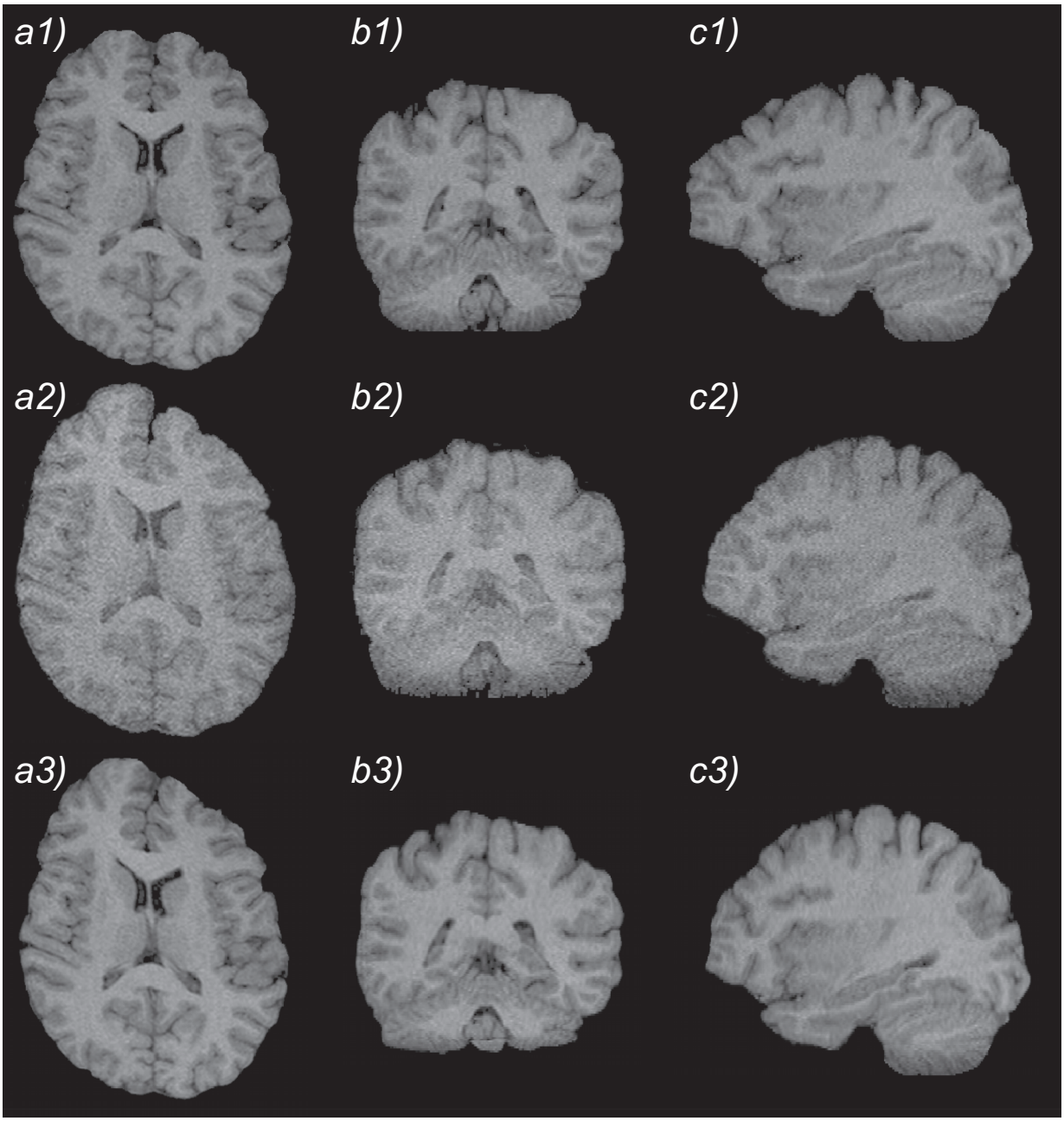

Figure 4.7: Pre- and intraoperative brain images after piecewise linear registration. *Top:* preoperative image registered rigidly to the intraoperative scan, *middle:* the intraoperative data and *bottom:* the corresponding slices of the non-linearly aligned preoperative image. *a)* Axial, *b)* coronal and *c)* sagittal views

Table 4.4: Point errors after piecewise linear non-rigid registration. RMS, VAR and STD denote the root-mean-square error, variance and standard deviation, respectively

Patient No	Point 1	Point 2	Point 3	Point 4	Point 5	Point 6	RMS error	VAR σ^2	STD σ
1	1.005	0.552	1.620	1.768	0.778	2.419	1.501	0.495	0.703
2	2.013	0.0	2.0	0.289	1.827	1.035	1.446	0.798	0.893
3	2.179	0.707	1.803	1.118	2.915	8.761	3.979	8.811	2.968
4	1.871	1.118	0.5	0.0	4.123	0.5	1.926	2.257	1.502
5	1.732	0.0	1.118	0.5	1.0	4.352	2.018	2.363	1.537

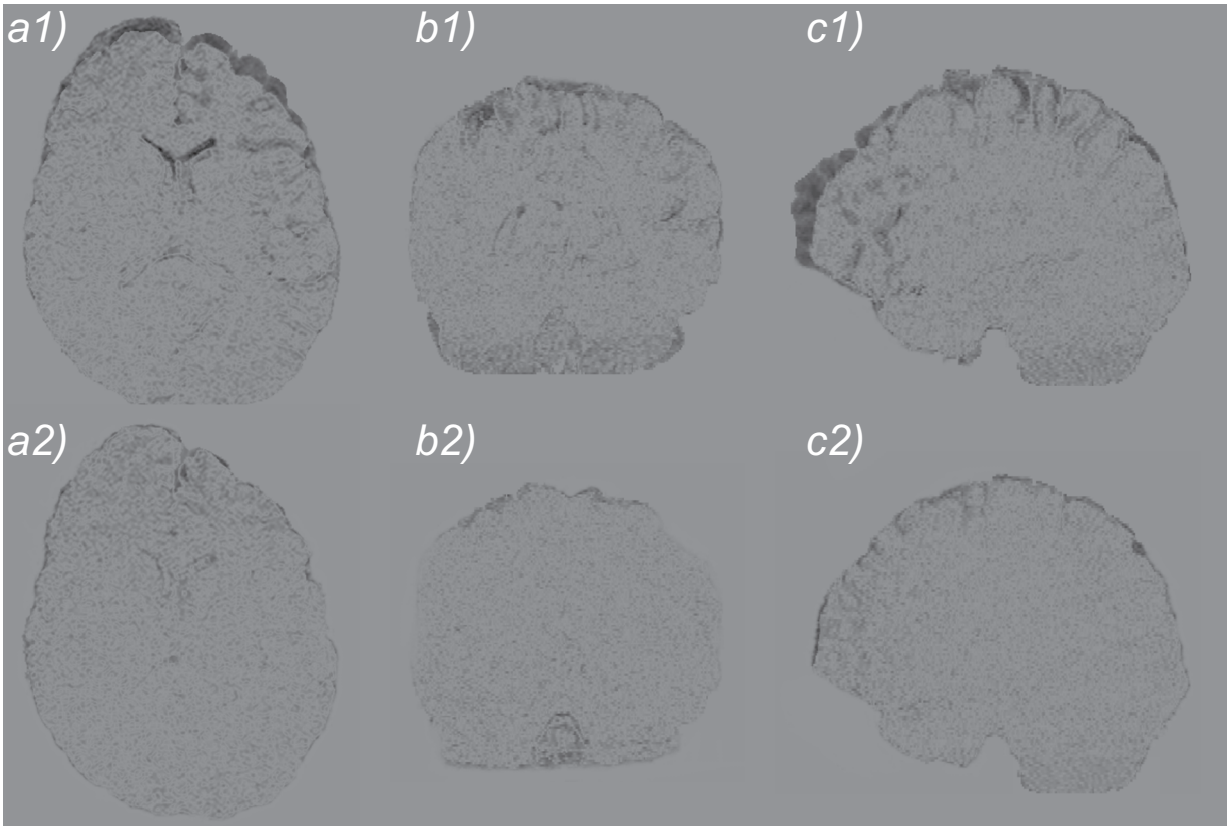

Figure 4.8: Difference images between the preoperative image and the intraoperative MR scan. *Top:* after rigid and *bottom:* after piecewise linear deformation. *a)* axial, *b)* coronal and *c)* sagittal slice views

4.2.9 Discussion

The presented results demonstrate the value of the introduced algorithm for non-linear registration of brain images. In terms of statistical analysis, the RMS error measured at six specified points of interest was significantly reduced in four cases from 5.04 - 8.56 mm after rigid registration by a factor of 2.62 - 5.92 to only 1.45 - 2.02 mm after the non-linear transformation (see Tables 4.3 and 4.4. In the worst case the RMS error was reduced from 9.18 mm to only 3.98 mm showing the limitations of the method.

For other cases the individual distances between the points prove that the algorithm was able to perform a good registration at the cortex region, which is usually represented in the similarity measure with a higher weight due to a significant amount of the air in the scans. Deeper structures from the preoperative scan were also non-linearly aligned to the respective structures in the intraoperative image within good precision. As a visual example supporting the numerical data, the anterior region of the right lateral ventricle in Figure 4.7a) can be considered.

The analysis of difference images confirmed the good quality of the registration. Only small intensity differences are visible in Figure 4.8. These are due in a big part to the intensity differences between the pre- and intraoperative imagery[6].

Furthermore, the presented method is characterized by relatively low computation times. Firstly, this was made possible by the use of graphics hardware for computational expensive interpolation operations. The number of necessary similarity measure evaluations was minimized with the use of a simultaneous perturbation stochastic approximation algorithm. This further reduced the overall computation times. The number of necessary NMI calculations was lowered to about 30% of those required by Powell's direction search method for Patient 1 (see Figure 4.6). Since the SPSA optimization method is characterized by a faster descent in the value of the similarity measure on the coarse hierarchy levels, the total computation time was even reduced to 13% of the time required by the Powell optimization. At the same time, the optimal NMI values found were almost equal for both optimization procedures.

Despite of the stochastic character of the optimization method, the same optimal results are approached each time with almost 100% surety. As can be seen from Table 4.1, for each case a full three-stage multiresolution registration was performed with a satisfactory accuracy within less than 60 minutes. This can be considered as a very good result with respect to the underlying complex problem. In four of five cases a very satisfactory non-linear registration of the pre- and intraoperative brain data was accomplished. This fact was confirmed by the numerical analysis of the experiments and also by the visual inspection of the results.

[6]Among others, these intensity artifacts are a result of the influence of the intraoperative environment such as use of a special coil for fixing the patient's head on the surgery table.

4.3 3D Free-Form Deformation

In this section, a free-form deformation (FFD) model and its implementation are presented. Similar to the piecewise linear model, the main idea is to warp the space surrounding an object that will then be warped implicitly. Now, to deform of the space we take 3D Bézier functions. This kind of FFD has many advantages. It contains inherent smoothness as well as elasticity, which makes it very suitable for describing the deformation of the soft tissue. Moreover, these functions are equipped with a simple mechanism for their modification and are characterized by intuitive behavior on alteration.

The disadvantage of 3D Bézier functions is the computational cost of their evaluation which is significantly higher compared to the simpler 3D piecewise linear model. Especially in the context of registration, many calculations of Bézier functions have to be performed. Therefore, to make the method more efficient this FFD approach has been implemented based on the hardware-accelerated 3D piecewise linear model presented in Section 4.2.

This section is divided into four parts. First, an introduction to the theory of Bézier transformations is given in Section 4.3.1. In the next section, our hardware-based FFD implementation is described and compared to the general FFD approach. In Section 4.3.3, the non-linear registration algorithm based on the accelerated FFD approach is described. Results of a series of experiments conducted with the presented algorithm applied to pre- and intraoperative MR images of the brain are summarized and discussed in Sections 4.3.4 and 4.3.5, respectively.

4.3.1 Deformation Model

Compared to Section 4.2.1, an additional parameter space $\mathcal{PS}$ is required to mathematically describe the deformation approach utilizing Bézier functions. The object space $\mathcal{OS}$ associated with the transformed dataset is considered to be parameterized with the function $\mathbf{P} : \mathcal{PS} \mapsto \mathcal{OS}$ leading from the parameter space $\mathcal{PS}$ being $[0,1]^3$ into this object space. For the purpose of deformation, the function $\mathbf{D} : \mathcal{PS} \mapsto \mathcal{TS}$ is introduced which is parameterized over the defined parameter space and leads to the texture space $\mathcal{TS}$. A lattice consisting of control points $b_{i,j,k}$ $(i = 0,\ldots,l, \quad j = 0,\ldots,m, \quad k = 0,\ldots,n)$ is placed in the texture space and defines the shape of function D in a unique way:

$$\mathbf{D}(s,t,u) = \sum_{i=0}^{l}\sum_{j=0}^{m}\sum_{k=0}^{n} B_i^l(s) B_j^m(t) B_k^n(u) b_{i,j,k} \; . \tag{4.11}$$

The function D is a trivariate tensor product of 1D Bézier functions where the basis functions B_i^l, B_j^m, B_k^n are Bernstein polynomials of order l, m and n, respectively (Figure 4.9). Changes in the form of the deformation function D can be achieved with movements of the lattice points $b_{i,j,k}$.

In the context of registration, an important property of any Bézier function is that it lies within the convex hull of the control points: $\mathbf{D}(s,t,u) \in \{\sum_{i,j,k} c_{i,j,k} \cdot b_{i,j,k}$, where $\sum_{i,j,k} c_{i,j,k} = 1$ and $i = 0,\ldots,l, \quad j = 0,\ldots,m, \quad k = 0,\ldots,n\}$. This property ensures that the Bézier function smoothly follows the control points without erratic oscillations. This is utilized in the registration algorithm by setting the border control points to be fixed. Therefore, we can prevent the deformation function from taking values from outside of the registered image volume.

4.3.2 Hardware-Accelerated Free-Form Deformation

Hardware is used to accelerate the FFD approach. Initially, the image data is loaded into the 3D texture memory of the graphics adapter since we want to employ the texture processing unit for performing the most expensive computations. The texture space $\mathcal{TS}$ being $[0,1]^3$ is associated with the texture memory in order to perform texture mapping.

The calculation of a single FFD consists of three steps. In the first step, the object is embedded in the lattice of control points. The control points lie in the texture space, while the object undergoing deformation is physically placed in the texture memory and in a logical sense in the object space. Initially, the control points in the lattice are uniformly displaced in the texture space.

To produce a deformation, in the second step control points are shifted to their new locations. This changes the structure of the control lattice, and this is expressed by the function:

$$\mathbf{M} : \mathcal{TS} \mapsto \mathcal{TS} \ ,$$
$$\mathbf{M}(b_x, b_y, b_z) = (t_x, t_y, t_z) \ . \tag{4.12}$$

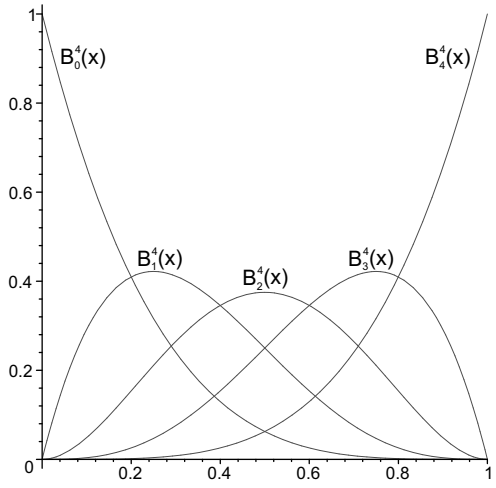

Figure 4.9: Bernstein polynomials of order 4

In our method, the changes in the control lattice are not directly based on the absolute coordinates of the control points. Instead, offset vectors (t_x, t_y, t_z) from the initial control points positions (b_x, b_y, b_z) are chosen as the free parameters of the Bézier transformation. This strategy allows us to treat the occurring free-form deformation as a change of a vector field deforming an object placed within it. Such a treatment is closely related to the physical nature of the phenomenon. In the initial stage, the vector field is set to 0 at each control point. In order to prevent illegal values of the deformation function (see the previous section), the offset vectors at the border are set to 0 during the whole registration procedure. Thus, only the inner control points of the Bézier function are free parameters for the optimization. This assumption is well motivated in practice since usually the interesting image information is contained in the interior of a 3D medical dataset and no deformation occurs at the boundaries of the volume.

In classical FFD approaches [142], after performing these two steps the new coordinates for every object point are explicitly calculated in the third and the final step based on the shifted coordinates of the control points. To circumvent this computation expense in our algorithm a discrete and uniform grid in the parameter space is considered. The function D incorporating the displacement function M is evaluated only on this sparse grid, defining the respective texture coordinates:

$$\mathbf{D}(s,t,u) = \sum_{i=0}^{l}\sum_{j=0}^{m}\sum_{k=0}^{n} B_i^l(s)B_j^m(t)B_k^n(u)(b_{i,j,k} + \mathbf{M}(b_{i,j,k})) \; . \tag{4.13}$$

In order to achieve a better approximation of the original shape of the deformation function the discrete grid can be set denser than the control lattice. After sampling the function D, the resulting texture coordinates on the sparse grid are used to propagate the deformation onto the whole volume using the trilinear interpolation. For this purpose, the 3D Bézier function is approximated with 3D piecewise linear model which was introduced in Section 4.2. Using this approach requires less computation expense since we do not need to process the whole 3D image voxel by voxel to obtain new intensity values, a necessary process for software approaches. Instead, graphics hardware is employed to reduce the execution time of these expensive operations. A 2D example presenting an approximation of FFD with piecewise linear patches is shown in Figure 4.10.

4.3.3 Registration

Similar to the approach that used only 3D piecewise linear functions, an initial estimation of the registration with Bézier functions was performed by rigid registration of the source and the target datasets. Having the datasets rigidly registered, the source dataset is loaded into the texture memory and is embedded in a lattice of control points. This structure allows intuitive changes of the volume. At the begin-

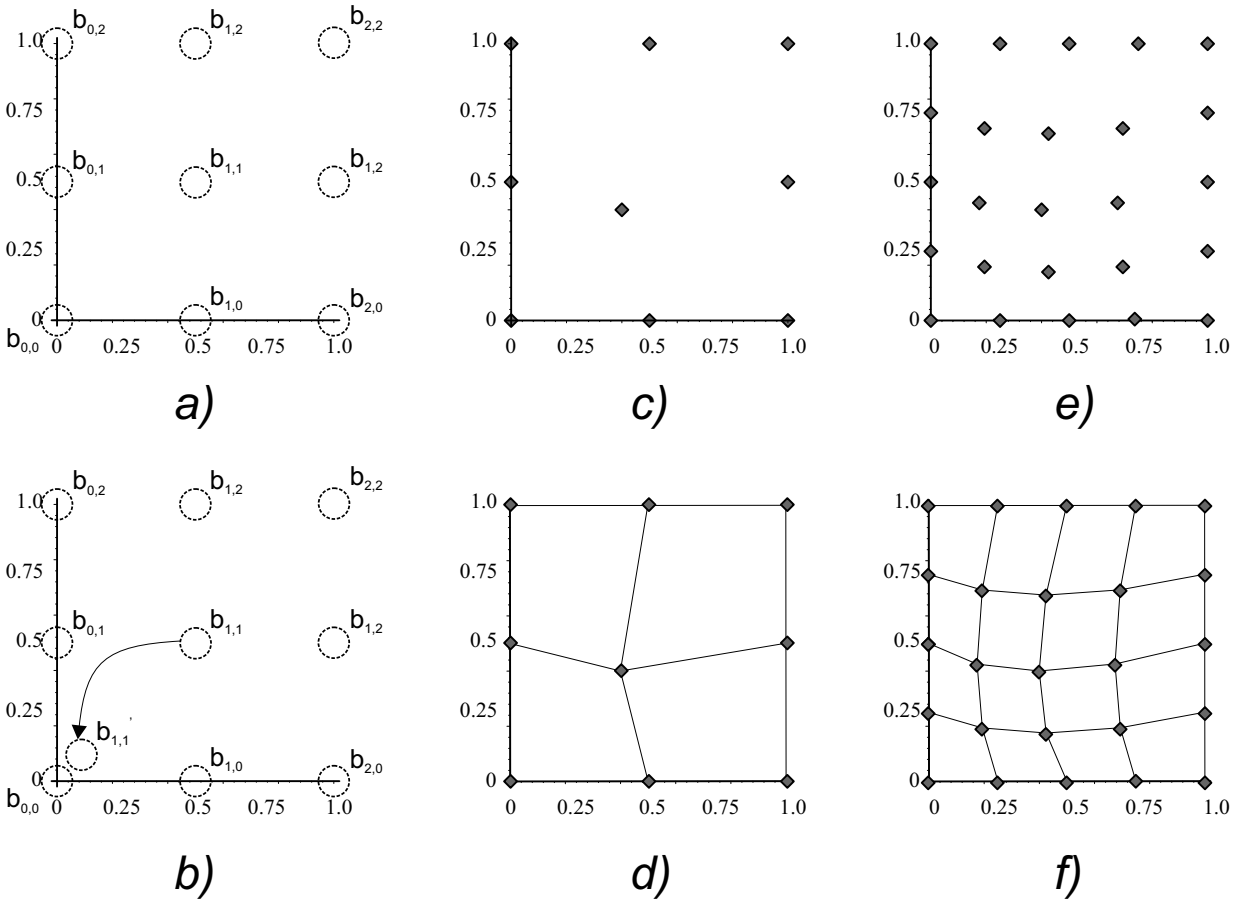

Figure 4.10: Subdivision of a slice into 2D piecewise linear patches. *a)* The Bézier function is defined over a 3×3 lattice. *b)* Control point $b_{1,1}$ was moved from its initial position $(0.5, 0.5)$ to $(0.1, 0.1)$. This resulted in $\mathrm{D}(0.5, 0.5) = (0.4, 0.4)$. *c)* Values of the image of function D on a uniform discrete grid 3×3. *d)* Resulting 2D piecewise linear subdivision of the slice. *e)* Values of the image of function D on a uniform 5×5 grid. *f)* Piecewise linear subdivision of the slice based on the values from *e)*

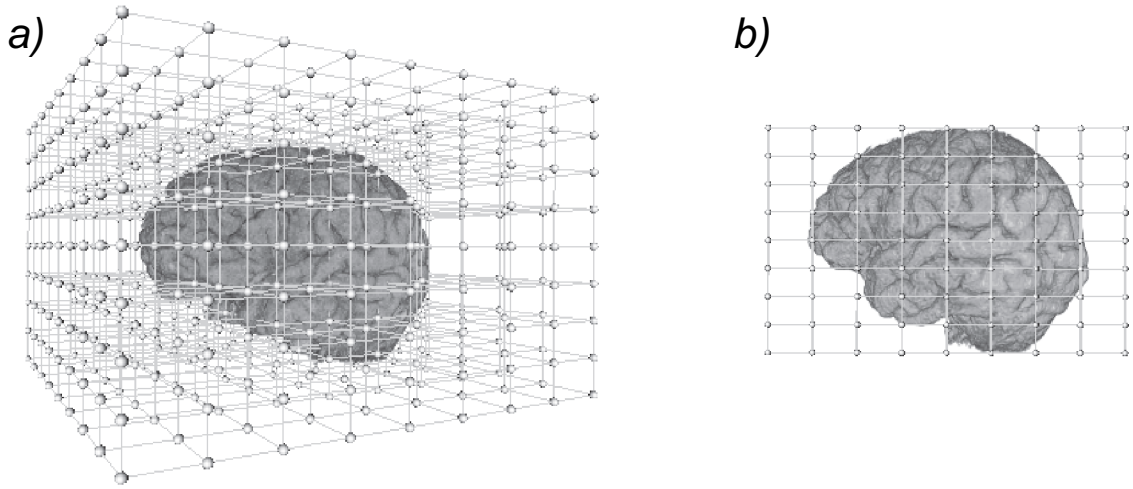

Figure 4.11: 3D free-form deformation. A dataset is initially embedded in a uniform lattice of control points. *a)* 3D view. *b)* A parallel projection

ning of the registration, this lattice has the form of a uniform parallelepiped (see Figure 4.11).

Then, during an iterative procedure, free control points in the lattice are manipulated such that the deformed volume aligns with the target volume. The optimal positions of the control points are estimated with the SPSA algorithm [166]. The quality of the registration is measured with the use of the normalized mutual information. Thereby, the degrees of freedom are the translations vectors from the initial positions of the inner control points in the lattice. In each iteration, the coordinates of all control points are stochastically perturbed and the new deformed volume is recalculated based on a piecewise linear approximation of the Bézier function. The algorithm optimizes the similarity measure between the recalculated and target volumes until a convergence is reached.

4.3.4 Results

The described hardware-accelerated FFD approach was assessed in registration experiments. Here, the same five pairs of datasets were considered as in Section 4.2.8 for the evaluation of the approach based on piecewise linear deformation only. Analogous to those previous experiments, each pair of the MR T1-weighted images was first rigidly registered and subsequently the brain volumes were extracted from all scans.

Each pair of brain images was then non-linearly registered with the use of the proposed approach. NMI was chosen as the similarity measure and the SPSA algorithm was employed for the optimization. The same parameter values controlling the behavior of the SPSA method were set as described in Section 4.2.8. Different to

Table 4.5: Computation data from experiments with the non-linear registration algorithm using 3D Bézier functions. For each case the hierarchy level, the corresponding viewport size, the final NMI value, the calculation time for one NMI evaluation and the total computation time are given

Patient No	Hierarchy	Viewport Size	Optimal NMI value	1 NMI evaluation Time [s]	Total Time[min]
1	level 0	128×128	1.25405	0.13	13.06
	level 1	256×256	1.25490	0.29	19.41
	level 2	512×512	1.26922	0.96	32.08
2	level 0	128×128	1.28869	0.13	13.12
	level 1	256×256	1.29179	0.29	19.28
	level 2	512×512	1.29005	0.97	32.38
3	level 0	128×128	1.28261	0.12	12.42
	level 1	256×256	1.28580	0.28	18.82
	level 2	512×512	1.28534	0.96	32.12
4	level 0	128×128	1.29582	0.13	13.07
	level 1	256×256	1.29875	0.29	19.33
	level 2	512×512	1.29846	0.96	32.08
5	level 0	128×128	1.28385	0.13	12.97
	level 1	256×256	1.28653	0.28	18.74
	level 2	512×512	1.28547	0.98	32.71

the 3D piecewise linear approach where the texture coordinates were directly optimized, in the FFD-based registration the optimal positions of the free control points in the lattice defining the shape of the related 3D Bézier function were iteratively calculated.

For each pair of the brain images, the FFD registration was performed in a three-stage multiresolution approach. All experiments were conducted with a control lattice consisting of $9 \times 9 \times 9$ control points. In order to better approximate the corresponding Bézier function, the function was sampled on a denser grid of $17 \times 17 \times 17$ points. This divided the object space uniformly into parallelepipedes of 1.56 cm $\times$ 1.56 cm $\times$ 1.0 cm.

The computation times and the achieved maximal values of the similarity measure for all experiments are presented in Table 4.5. As previously, the measurements were conducted on a P IV 3.0 GHz system running under Linux Suse 8.2 and equipped with Nvidia GeForce FX 6800 Ultra graphics card with 256 MB memory.

To assess the accuracy of the registration and to compare it to the results achieved with the 3D piecewise linear registration, the same anatomical features

Table 4.6: Point errors after registration with the use of Bézier functions. RMS, VAR and STD denote the root-mean-square error, variance and standard deviation, respectively

Patient No	Point 1	Point 2	Point 3	Point 4	Point 5	Point 6	RMS error	VAR σ^2	STD σ
1	1.122	0.483	0.427	1.386	0.637	2.180	1.208	0.455	0.675
2	2.647	1.0	1.871	1.022	1.512	0.524	1.587	0.571	0.755
3	0.707	0.5	1.118	1.581	0.5	1.414	1.061	0.221	0.470
4	1.803	1.118	0.5	0.0	1.581	0.0	1.099	0.616	0.785
5	0.707	0.0	0.707	0.5	0.5	4.686	1.977	3.011	1.735

were marked in the deformed images and the respective distances and statistical measures of the root-mean-square error, variance and standard deviation were calculated (see Table 4.6). The point-to-point distances after rigid registration only have already been provided in Table 4.3.

Finally, the registration quality was visually assessed. In Figure 4.12 2D views through the original pre- and intraoperative data and the corresponding views of the non-linearly aligned image are presented for Patient 1. Furthermore, the quality of the registration is revealed in the visualization of the intraoperative image fused with contours extracted from the deformed image. These images should be compared to the ones obtained after rigid registration which were presented already in Figure 3.8. Additionally, for Patient 3 the vicinity of the point (P_6) in the images obtained with both the 3D piecewise linear and the FFD registration approaches is comparatively shown in Figure 4.13.

4.3.5 Discussion

The described experiments exhibited the qualitative strength of the non-linear registration method based on hardware-accelerated 3D Bézier functions. The quantitative evaluation of the registration results showed that the RMS error measured at certain characteristic regions was reduced by a factor of 3.15 - 8.65 to 1.06 - 1.98 mm with the use of the proposed technique. Considering this statistical error as the reference, the achieved results were considerably better in comparison to the 3D piecewise linear approach.

Contrary to the results obtained with the use of the 3D piecewise linear registration, no misregistration was observed for Patient 3. The analysis of the images in Figure 4.13 suggests that for the piecewise linear deformation model misregistration is a result of violating the topology of the hexahedra structure during the registration process. This can be prevented by introducing an additional constraint

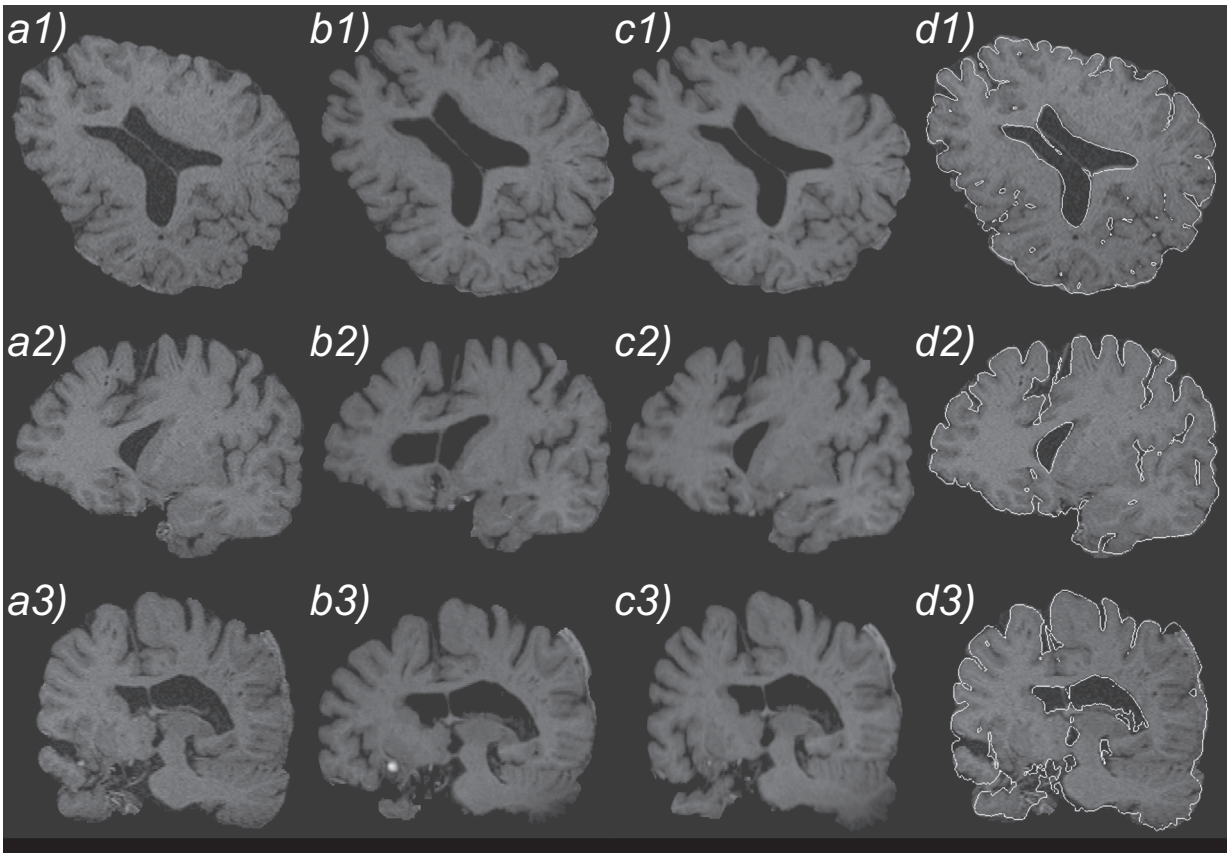

Figure 4.12: Results of the FFD-based non-linear registration of the pre- and intraoperative MR scans of the brain. *a)* An intraoperative slice image. *b)* The corresponding slice from the preoperative scan. *c)* Slice from the non-linearly deformed brain. *d)* The intraoperative image overlayed with the contours of the brain from the aligned image *c)*. *Top:* Axial, *middle:* sagittal and *bottom:* coronal views

for the shifting of the texture coordinates. This would, however, increase the computation expense. In images produced with the FFD approach such effects are not observed. This can be explained by the inherent smoothness of Bézier functions. Furthermore, the movements of the control points in the lattice during the optimization process have less impact on the deformed shape than changes of the same magnitude applied directly to the texture coordinates. Thus, injuries of the structure topology are less probable.

For other cases, visual comparison of the corresponding results for both the 3D piecewise linear and the FFD-based approaches revealed only small differences at first sight. However, a comprehensive visual analysis of the datasets with the use of interactive techniques such as a magic lens revealed the superiority of the latter approach in relation to the piecewise linear procedure. This advantage was also confirmed in the statistical analysis (see Table 4.6).

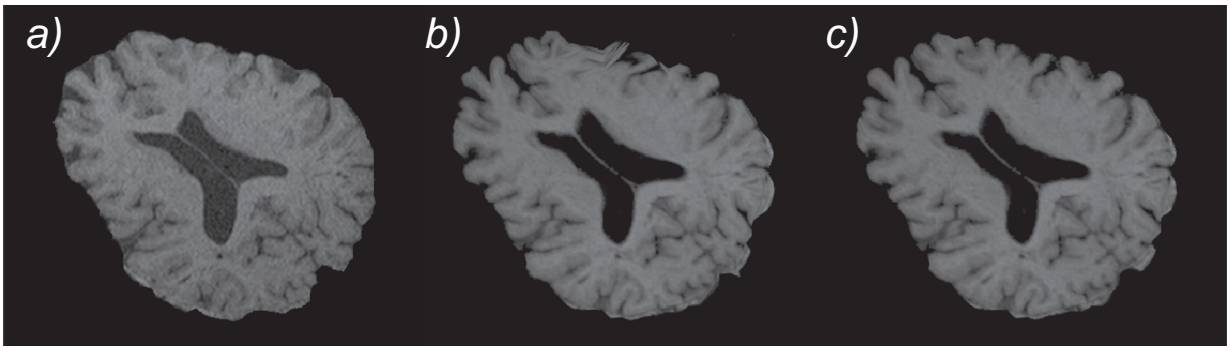

Figure 4.13: Example of registration artifacts. *a)* Slice from the intraoperative image. *b)* The corresponding slice from the preoperative image transformed with the 3D piecewise linear registration. Some artifacts are visible in the region of the anterior cortex. *c)* The corresponding view of image deformed with the use of 3D Bézier functions. No texture disturbances occur

For the optimization, the same parameters which were already used for the SPSA algorithm in the experiments with 3D piecewise linear registration proved also to be a reliable choice for the FFD registration approach. A good convergence behavior was observed after 3000, 2000 and 1000 iterations in the first, second and third hierarchy levels, respectively. This means a constant time was required for the whole registration procedure. The recorded computation times are a little bit higher in comparison to those from Section 4.2.8, which is caused by the double so fine ($17 \times 17 \times 17$) underlying hexahedra structure and by additional evaluations of the 3D Bézier function. However, this higher computational cost is balanced by a better registration quality in comparison to the 3D piecewise linear method (see Section 4.2.8).

Overall, use of an approach based on 3D Bézier functions provided very satisfying registrations for all five cases. A good alignment of the outer and inner structures was observed. In Figure 4.12 slice views through the non-linearly transformed images are accompanied by the corresponding pre- and intraoperative data. In this visualization, a good alignment of the data is exhibited. Especially, the misalignment both at the cortex region and at the ventricles has vanished. There is also a misalignment seen in the coronal view for the left lateral part of the brain. This is due to the tumor resection at this location. Nevertheless, the neighboring structures are well registered, which is seen in the numerical results for this experiment.

4.4 Conclusion

In this chapter, a 3D piecewise linear approach and a technique using 3D Bézier functions were presented for non-linear registration. Both hardware-accelerated methods are characterized by a high reduction in computation time in compari-

son to software-based registration implementations. In the second algorithm the flexibility of the Bézier transformation is combined with the performance achieved by using graphics hardware. Experiments conducted with data from real patients exhibited good results for both methods. However, the approach based on Bézier function was proven to be superior in a numerical analysis. At the same time, the required computational overhead related to the evaluation cost of the Bézier function is still acceptable in terms of the overall performance. Nevertheless, the 3D piecewise linear approach can also be sufficient for some applications where the speed of computation is more important than a high precision.

As an extension to both techniques. hardware-accelerated local illumination model has also been introduced in [135] which allows rendering of the deformed volumes for visualization purposes. This topic is, however, outside the main scope of this thesis and therefore not adressed here.

Part III

Simulation

Chapter 5

Survey of Soft Tissue Simulation

Previous parts of this thesis addressed registration methods. The algorithms described there focused on their application in the registration of head images. In that context, linear registration was considered for rigid alignment of structures (e.g. skull bone and undeformed parts of the brain) and non-rigid algorithms were applied and evaluated for registration of pre- and intraoperative brain images where highly non-linear tissue deformation occurred.

Non-linear registration is a complex problem due to the information difference between two images (e.g. the volume of brain differs significantly between pre- and intraoperative brain images). Despite the flexibility of the image registration methods developed within this thesis (where no additional *a priori* knowledge about the considered objects is required), this information difference is certainly a limiting factor for the use of such techniques. Another possibility to cope with and to compensate for the non-linear brain shift phenomenon is to predict the occurring soft tissue deformation with a physically-based computational model. Within such a model, the physics of a deformation is described by a system of partial differential equations. A variety of factors influencing the occurring deformation can be incorporated into such a model. Solving these equations leads to a *simulation* of a given process. Although taking into account all possible conditions (e.g. mechanical properties of the tissue or surgical events for the simulation of brain behavior during surgery) is a very intricate and complicated task. Following computational strategy contributes to a better understanding and analysis of the physical nature of an occurring phenomenon.

In this chapter, works on the physically-based modeling of soft tissue deformation are reviewed. This is instrumental in assessment of the own model for brain shift presented in the next chapter. Section 5.1 introduces applications of soft tissue simulation in the medical context. Section 5.2 is more specific and discusses existing works dedicated to the simulation and analysis of intraoperative brain deformation. The approaches are categorized according to different assumptions about the brain behavior and according to acting factors that are incorporated into the model. This provides a good basis for the presentation of own results.

5.1 Clinical Applications

Physically-based models are increasingly being used in the medical field, both for registration purposes and for simulating natural phenomena. The former applications have been comprehensively addressed in Section 2.4.1.3. In this section, simulation approaches are reviewed in the context of clinical applications.

There are numerous problems that can be approached with physically-based computational models: surgery simulation and practicing interventions on a virtual patient, supporting a clinical diagnosis of a real patient, surgery planning and finally performing the surgery under computer guidance, to name only the most important applications. Since simulation approaches differ drastically depending on the modeling process, in the following we present existing works categorized according to the simulated phenomenon. This list shows the extent of possible clinical applications of simulation techniques.

The most important groups of works on physical models for clinical assistance include:

A. Cranio-facial surgery simulation. This type of surgery aims at improvement of the appearance of the patient or at restoring normal body function (e.g. for patients with mouth tumors).

Fujioka *et al.* [60] reported very early a simple interactive surgical simulation system allowing the cutting and moving of facial bone blocks. As a further feature, prediction of skin surface movement was also possible. The first, however, to incorporate real modeling of soft tissue was Delingette *et al.* [50]. They used 3-simplex meshes to represent the fat tissue lying between the skull and the skin surface modeled by 2-simplex meshes. In [86], a multi-layer modeling of soft tissue incorporated in a mass-spring model was compared against a finite element implementation based on pseudoelasticity. A case study was performed there to examine the introduced deformation models.

A generic model for maxillofacial[1] surgery simulating the contraction of muscles with different mechanical properties (transverse isotropy, variable stiffness) was adapted in [31] to a specific patient case by registration. A volumetric finite element method was used in this work to perform the simulation.

B. Partial and total knee replacement surgery. For such surgeries, predicting kinematics and contact mechanics behavior of the implant is crucial. Mommersteeg *et al.* [116] developed and experimentally verified in a 3D model the knee and the surrounding soft tissue, considering multi-fiber ligament[2] bundles with non-uniform mechanical properties. A prototype for the simulation

[1] Of or relating to the upper jaw and face.
[2] A ligament is a tough band of fibrous tissue supporting a joint. It is located between the bones.

of arthroscopic[3] knee surgery was described by Gibson *et al.* [65]. As an alternative to FEM, the authors modeled the deformation of soft tissue with local disturbances and elastic relaxation. Haptic interaction was combined with real-time rendering in order to provide visual information to the user. The influence of kinematics on the internal stress distribution in the knee was investigated in [66]. The results were compared to experimental data acquired during a single gait cycle in a knee wear simulator. Chen *et al.* [32] computed passive knee kinematics by finding the contact state minimizing the internal strain energy in the ligaments surrounding the knee.

C. Hepatic interventions. A volumetric deformable liver model for simulating laparoscopic surgeries was presented by Cotin *et al.* [43]. In this model, the liver was treated as a linear elastic organ and a haptic device was used for the deformation, which could be performed interactively in simple models. An extended version of this model [42] combining the advantages of finite element and mass-spring models allowed simulation of cutting on complex liver structures. Miller *et al.* [113] analyzed the mechanical properties of liver and kidney tissue with the use of a non-linear viscoelastic model and successfully validated this model in physical experiments. Radio-frequency ablation methods were analyzed in [67] with the finite element method, where the heating of tissue during ablation was modeled with the Pennes bioheat equation. The results presented in that work has contributed to the improvement of the treatment of hepatic tumors.

D. Cardiac analysis. A general mathematical representation of ventricular geometry and muscle fiber structure of the heart was presented by Nielsen *et al.* [121]. A technique for the analysis of cardiac activation problems over a bidomain framework of interpenetrating finite elements and finite difference meshes was introduced in [27]. Sermesant *et al.* described a comprehensive system for biomechanical modeling of the heart based on a fusion of anatomical, functional and mechanical information from different imaging modalities (e.g. DTI imaging for mechanical elastic anisotropy of fiber directions). A process of FE modeling of a system including blood, myocardium and an ablation catheter was described in [183]. Different temperature-dependent mechanical myocardium properties were analyzed there as were their effects, which led to a better accuracy of the developed numerical model.

E. Ophthalmic[4] surgery. An anatomically detailed 3D computer graphic model of the eye and the surrounding facial tissue for use in a surgical simulator was

[3]The term "arthro" stands for joint, "scopic" for "looking inside". Arthroscopic surgery is performed with small surgical instruments and a tube-like instrument equipped with a camera and a light source.

[4]Relating to the eye.

presented by Sagar *et al.* [144]. The stress distributions in the cornea[5] were modeled with the use of a finite element theory for homogenous, orthotropic and non-linearly elastic materials. A quantitative model of corneal surface smoothing after laser refractive surgery was discussed in [80]. Growth, migration and loss of the corneal surface were there expressed with a partial differential equation system which was solved for the equilibrium state. A poroelastic finite element model was employed by Luboz *et al.* [96] to improve surgery planning by predicting the relationship between the decompressed volume of orbital soft tissues and the force exerted by the surgeon.

F. Other applications. There exist a number of other applications of simulation techniques for clinical problems. Azar *et al.* [5] investigated and validated a deformable finite element model of the breast for guiding a biopsy. They incorporated the properties of fatty tissue into the model and described the breast deformation with Lagrange's equations of motion. Deformation of the uterus[6] for diagnostic gynecological laparoscopy was modeled in [174] with a finite element technique incorporating collision detection. To accelerate the process, a parallel implementation was accomplished. A FEM simulation of needle insertion and radioactive seed implantation for the planning of brachytherapy for treating prostate cancer was presented by Alterovitz *et al.* [3].

Research on deformable models for the simulation of natural processes has been evolving over the last years towards more physically accurate models. At the same time, techniques to accelerate numerical computations for interactivity purposes have been widely investigated. Both issues are important from the clinical point of view. Accuracy is crucial in planning or conducting a computer-assisted surgery. The clinical requirements limit the available computation time and demand interactive updates. In this context, especially complex modeling problems (e.g. the simulation of brain shift) have to be further investigated.

5.2 Survey of Brain Shift Simulation

The phenomenon of intraoperative brain deformation is one of the most important clinical applications. A review of existing non-linear registration methods concerning this problem has been presented in Chapter 2. Another way to compensate for brain shift is to employ a physically-based deformation model where events associated with the surgical procedure are modeled. This section reviews such models.

A number of physical descriptions of the brain have been discussed in various works. Especially in recent years, there has been growing interest in this field.

[5]The clear, dome-shaped surface that covers the front of the eye.

[6]A hollow muscular organ in the pelvic cavity of females containing the developing fetus, the womb.

Introduction of image-guided neurosurgery has resulted in a high number of studies on the recognition and characterization of intraoperative brain motion during surgery. At the same time, image-guided procedures which are mainly based on preoperative planning have increased the need for a reliable estimation of tissue motion during surgical interventions.

In the last years, various approaches modeling intraoperative brain behavior have been presented. These approaches differ significantly from each other. The most characteristic differences are related to the assumptions of the mechanical properties of the brain tissue and to the governing equations describing the deformation. Another differentiating factor are the additional features incorporated into the model. In this section, important existing simulation algorithms are discussed. The works are categorized according to the mathematical description of the brain.

5.2.1 Linear Elasticity Models

Most of the proposed biomechanical models of brain deformation represent the brain as an elastic solid material. Hagemann *et al.* [69] developed a biomechanical model for brain deformation. The model assumed linear elastic behavior of the brain tissue. The presented approach was, however, validated only for two-dimensional examples and used an extremely computationally expensive domain discretization (image pixels were considered as elements of the finite element mesh). Another disadvantage of this method is that it relies on semi-manually determined correspondences between pre- and postoperative images. Thus, it does not allow the *prediction* of a brain deformation.

A somewhat improved version was presented by the same authors in [68]. This model considered two different physical descriptions for modeling incompressible fluids and rigid and elastic materials. It still required the manual definition of the image correspondences and, despite a modified discretization approach, allowed performing of the simulation only for small (61×61) two-dimensional images. Ferrant *et al.* [57] presented a linearly elastic FEM model which was also driven by surface correspondences. In comparison to the previous two works, this model could successfully overcome some limitations. First, in this approach the correspondences were identified automatically and second, the authors evaluated their method with three-dimensional brain data. In order to perform a deformation still pre- and intraoperative data were required which does not allow the method to be applied in an environment without intraoperative imaging or in the planning image-guided surgery.

Skrinjar *et al.* [153] proposed a model consisting of a set of interconnected nodes having assigned masses. The connection between the nodes was expressed as a Kelvin solid model[7]. The proposed framework also modeled the brain-skull interac-

[7]The Kelvin model is a parallel connection of a linear spring and dashpot. It assumes linear dependency between stress and strain and is suitable for modeling slow and small deformations.

tion with the use of numerical integration. Intraoperative measurements of some brain surface points were required during the surgery to reliably guide the model. An extension of this approach applied stereo cameras for tracking the exposed brain surface [154]. The reconstructed tissue surface was then used as a boundary condition for a deformable brain model. Additionally, the *falx cerebri*[8] was modeled by fixing the corresponding model nodes which also reduced computation time.

Another linear elastic model based on boundary extraction was presented by Wang *et al.* [188]. The corresponding shape points required in the model were estimated with the use of the Bayesian formulation of statistical information. Such a model, however, cannot be applied to the simulation of intraoperative brain deformation, due to the great variation in the extent and direction of the tissue shift between individual cases. Additionally, this method was formulated for 2D images and further investigations would be required before applying it to a 3D problem.

5.2.2 Viscoelastic Models

Mechanical stress in the brain was investigated in [147]. There, the brain was modeled as a three-domain structure with white and grey matter and ventricles each assumed to be isotropic and homogenous, though with different linear viscoelastic behavior. Unfortunately, this model was validated only for a section slice derived from a 3D brain model. Another viscoelastic model of brain tissue was presented by Brands *et al.* [21]. The soft tissue was treated there as nearly incompressible and the overall stresses was expressed as a sum of linear elastic volumetric stress and a deviatoric part depending on change of shape only. In experiments with a porcine brain, strain-dependent behavior of the tissue was observed to be well modeled. The authors scaled the simulation results by some empirically estimated factor to obtain optimal correspondence.

Two non-linear constitutive models of brain tissue were described by Darvish *et al.* [47]. There, a quasi-linear viscoelastic model was used to describe the spatial non-linearity and a non-linear viscoelastic model was taken to allow for both spatial and temporal non-linear behavior of the tissue. In experiments with bovine brain tissue the value of the presented model was confirmed. At the same time the authors observed that temporal non-linearity of the tissue could almost be neglected in the long-time responses, which demonstrates the effectiveness of modeling brain shift with a linear elastic model.

Miller *et al.* [114] proposed a non-linear viscoelastic model for modeling soft tissue. They showed non-linear stress-strain relations. In their experiments, porcine brain was used. This method is, however, computationally expensive and difficult to implement in finite element procedures. Therefore, in their next work a simpler, quasi-linear viscoelastic model was presented [115]. The authors validated this modified approach considering the elasticity parameters obtained *in vitro* for

[8] One of the curved folds of the dura matter which separates the left and the right hemisphere.

porcine brain tissue. During their simulation they could reproduce the forces at a level $31\ \%$ lower than those recorded in the experiments, which can be considered as good, bearing in mind the large variation of the mechanical properties of biological tissues. Nevertheless, despite these promising results, the approach should be more comprehensively validated with human brain data, especially with respect to the very specific conditions prevailing in intraoperative environment.

5.2.3 Biphasic Theory and Poroelastic Models

Biphasic theory was used very early to describe the mechanics of biological tissues [119]. In the last years, new applications and significant improvement of this theory have been seen. The biphasic description is similar to the porous-media models first comprehensively investigated in [17]. Both models are based on the assumption that the continuum consists of two phases. However, in pure biphasic theory two different continuum descriptions are taken for the two separate phases which interact within an entire volume. Poroelastic approaches consider a single continuum where the dynamics is influenced by changes in its hydration or pressure conditions [152].

Nagashima *et al.* [120] were first to apply a poroelastic model for the description of brain tissue. In that work, they examined the biomechanical properties of hydrocephalus. Specifying the material properties of the brain parenchyma, the loading characteristics and the boundary conditions, a two-dimensional FEM simulation of the hydrocephalic process was conducted and the subsequent analysis of the resulting stress and pressure distribution in brain structures was performed.

By means of a poroelastic theory, Miga *et. al* [111, 109] recently modeled the brain as an elastic body with intersticial fluid. In that work, the deformation characteristics of porcine brain was analyzed *in vivo*. Results demonstrated a $75\ \% - 80\ \%$ predictive capability and showed the significance of accounting for hydrodynamics in the simulation model. In numerous other works, the same group enhanced their framework incorporating the heterogeneity of the brain tissue into the model [108] and investigating the influence of the falx cerebri on the extent of the occurring deformation [110].

Taylor *et al.* [178] considered a two-dimensional finite element biphasic model of the hydrocephalus with fluid and solid phase for modeling the brain and ventricles, respectively. Although a linear elastic formulation was applied for describing the brain tissue, the value of the elasticity modulus for the brain was derived based on viscoelastic theory.

5.2.4 Conclusion

Miller *et al.* [114, 115] could show the viscoelastic nature of soft tissue, but the extent of this factor in the occurring deformation has still to be closely investigated

for various tissue types (e.g. the difference between diseased and healthy tissue or between animal and human brain should be addressed). What seems to be, however, the main limitation of a purely viscoelastic model is that it does not account for the hydrated nature of the brain. In contrast, this factor and also the coupling between hydrodynamics and occurring deformation are expressed by the poroelastic model, which suggests it as a reasonable choice for describing volumetric changes of saturated brain tissue under load. Thus, although in [112] some limits of the biphasic description for modeling of soft tissue are demonstrated, in this thesis the poroelastic model is considered for simulating brain shift.

There is no agreement as to the optimal model for soft tissue concerning its heterogeneity. Differences in the mechanical properties of gray and white matter were reported in [132]. Miga *et al.* [108] showed that the heterogeneity of the stiffness properties of tissue accounts only for $1\,\%-3\,\%$ of the tissue motion predicted with their model. Also in the work by Ozawa [126], no significant difference in the modulis of elasticity between the gray and white matter was observed. Thus, in our model homogeneity and isotropy are assumed for the entire brain volume.

Chapter 6

Computational Model of Brain Shift

In the previous section, possible clinical applications of physically-based simulation models has been comprehensively reviewed, and a separate section there was devoted to research on brain shift simulation. Various existing works on this subject have been enumerated and the important differences between the algorithms have been addressed. Even if some of those reviewed models are very promising, it is still difficult to perform a reliable simulation of brain deformation, especially with respect to time and the accuracy requirements of image-guided neurosurgery.

There are many factors justifying the development of a computational model: improved surgery planning, intraoperative updating of the preoperative imagery or a better understanding the physics of tissue deformation. One of a number of crucial issues regarding the development of a computational model of a physical process is the choice of the underlying equations describing the phenomenon. In our computational framework the brain is described in terms of a three-dimensional model of consolidation for a linearly elastic and porous solid with viscous fluid undergoing diffusion [152]. Since the incorporated fluid phase accounts for intracranial pressure and CSF drainage, this model is more realistic than models which treat brain as a single phase. At the same time, the tissue elasticity is assumed to be linear, which reduces the required computational expense in performing the model calculations.

The second important concern is the choice of an efficient and stable method for solving the physically-based equations. In this thesis, a finite element method based on unstructured tetrahedral grids has been considered. In order to ensure the stability of the numerical calculations, a customized approach for grid generation has been proposed. For a better prediction of soft tissue behavior, the intraoperative data has been used for the analysis and estimation of the optimal elastic properties of the tissue. This algorithm is addressed in Chapter 7. Furthermore, to reduce the computation time, a parallel implementation of the developed algorithm has been conducted which is presented in Chapter 8.

This chapter presents the main simulation framework for modeling the brain shift phenomenon. Within this system, it is possible to predict intraoperative brain deformation. In Section 6.1, a fundamental method for performing numerical simulations is discussed and the important mathematical properties of this technique are addressed. The constitutive model equations are the subject of Section 6.2. The discretization of the governing equation system and the resulting matrix and vector form are addressed in Section 6.3. In Section 6.4, a procedure for the generation of the 3D mesh geometry of our model is presented. Section 6.5 describes a technique for setting proper boundary conditions which reflect the intraoperative situation. The algorithm for volumetric image reconstruction after simulation is described in Section 6.6. Finally, the results of the application of the described framework for simulation of brain deformation are collected in Section 6.7. The analysis of the results and a conclusion are found in Sections 6.8 and 6.9, respectively.

6.1 Finite Element Theory

In most physically-based simulation models, the process of brain deformation is described by a set of partial differential equations (PDEs). Performing a numerical simulation is then equivalent to finding a solution of the given governing equation system. The usual general technique applied for determination of such numerical solutions is the *finite element method* (FEM). This technique has been widely used for conducting simulation in structural and continuum mechanics and has also found application in the biomedical field. This section shall give a general overview of the finite element method. The specific resulting linear system and other finite element considerations for the biomechanical model of brain shift are presented in the next chapter.

Modern finite element approaches are usually obtained by either *weighted residual* or *variational* integral formulations. Here, we will concentrate only on the former procedure since it has been followed in our model. Let us consider a general governing differential equation:

$$\mathbf{L}(\mathbf{v}) = c \; . \tag{6.1}$$

Here, L denotes a differential operator acting on an unknown function[1] v and c is the source term. In the method of weighted residuals, the desired continuous function v is replaced by a finite series approximation $\mathbf{v}_h$:

$$\mathbf{v}_h = \sum_{i=1}^{n} a_i \cdot \varphi_i \; . \tag{6.2}$$

The functions $\varphi_i, \; i = 1, \ldots, n$ are denoted as the *basis functions*. Replacing v with $\mathbf{v}_h$ in Equation 6.1 results in a residual term r:

$$\mathbf{L}(\mathbf{v}_h) - c = r \; . \tag{6.3}$$

[1]The function sought may take scalar values or may map a vector of several variables.

The approximated solution is optimal if the coefficients a_i are determined which minimize the residual r. The natural approach would be to set the integral of r to zero. This would, however, generate only one equation. Since we have n unknown coefficients a_i, weighting functions $\boldsymbol{\omega}_i,\ i = 1, \ldots, n$ are introduced. This yields:

$$\int_\Omega r \cdot \boldsymbol{\omega}_i \, dx = 0, \quad i = 1, \ldots, n \ , \tag{6.4}$$

where Ω is the domain of function v. According to Equation 6.3, the governing equation is modified accordingly, resulting in:

$$\int_\Omega \mathbf{L}(\mathbf{v}_h) \cdot \boldsymbol{\omega}_i \, dx = \int_\Omega c \cdot \boldsymbol{\omega}_i \, dx \ . \tag{6.5}$$

The choice of weighting functions defines the type of weighted residual technique. The *Galerkin method* arises when the basis functions φ_i are used as the weighting functions $\boldsymbol{\omega}_i$. This leads finally to the following integral form of the governing equation:

$$\int_\Omega \mathbf{L}(\mathbf{v}_h) \cdot \varphi_i \, dx = \int_\Omega c \cdot \varphi_i \, dx \ . \tag{6.6}$$

The basis functions are usually non-zero on a certain local subdomain and otherwise equal zero. Using this technique, global equations can be obtained by summation of the local contributions. This results in a significant simplification of the resulting set of equations and in a reduction of computation time required to solve the equation system.

To perform finite element calculations, the volume of interest (e.g the brain) has to be first divided into an interconnected set of geometric primitives called *finite elements* by Clough [38]. An example (in 2D) of a domain discretization is presented in Figure 6.1.

The subdivided geometry and discretization of the governing equations permit the approximation to the solution to be obtained element by element according to:

$$\int_\Omega \mathbf{L}(\mathbf{v}_h) \cdot \varphi_i \, dx = \sum_{e=1}^{m} \int_{\Omega_e} \mathbf{L}(\mathbf{v}_h) \cdot \varphi_i \, dx \ , \tag{6.7}$$

$$\int_\Omega c \cdot \varphi_i \, dx = \sum_{e=1}^{m} \int_{\Omega_e} c \cdot \varphi_i \, dx \ . \tag{6.8}$$

In these equations, the continuous partial differential equations defined over the whole computation domain are approximated by a discrete sum of integrals defined over each geometry primitive. Here, Ω_e is the domain of element e and m is the number of all elements ($\sum_{e=1}^{m} \Omega_e = \Omega$). Additionally, the basis functions usually have the interpolation property at the vertices defining the primitives.

After the governing equations have been expressed in terms of contributions from local elements, the remaining fundamental problem is to establish the matrix

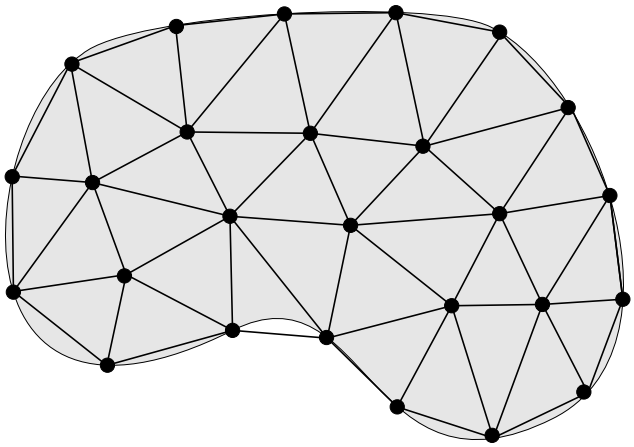

Figure 6.1: Example of finite element discretization in 2D. The domain is subdivided into two-dimensional triangular elements

form of the equations. This can be achieved if the differential equations (as in the form presented in Equation 6.3) are linear with respect to the coefficients a_i. Since the rearranging of the given equations into a linear system depends very strongly on the form of the governing equation, unnecessary and complicated considerations for a general case will not be presented here. Generally, the resulting system of linear equations will be of the form $\mathrm{A}\mathbf{x} = \mathrm{b}$. The local contributions A_e of each element to a discretized version of the PDEs are calculated based on values at the element's nodes only and then gathered into a global *stiffness matrix*[2] A. This process is called *assembling*. Additionally, boundary conditions are processed and the respective terms b_e are scattered into a global *load vector*[3] b. An example of matrix and vector assembly for a mesh consisting of four tetrahedra is schematically presented in Figure 6.2.

Finally, the linear system of equations is solved with use of a numerical method. For this purpose, iterative approaches are usually utilized. The stiffness matrix can be optionally preconditioned, which can lead to a better convergence and stability of the computations. As the solution, the coefficients of the basis functions are known at each node of the underlying geometry. Based on this, the solution can be propagated onto an arbitrary point of the computational domain. Having the solution of the constitutive equations, the numerical solution can be visualized and analyzed in a convenient representation, depending on the initial problem.

[2]The stiffness matrix is a historical notion originating from the time when the finite element technique was invented for studying stresses in complex airframe structures.

[3]The load vector is also a historical term having the same origin as stiffness matrix.

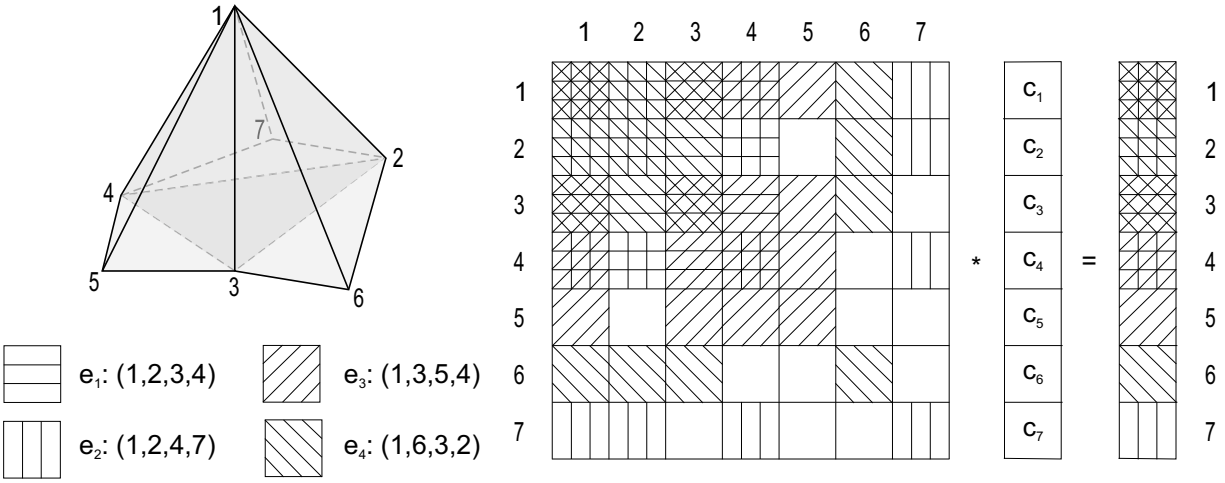

Figure 6.2: An example of matrix and vector assembly for a four-element mesh, with one parameter per node. *Left:* the topological connectivity between the mesh elements is shown. A different line pattern was assigned to each of the four tetrahedra in the depicted mesh to denote the source of the various coefficients occurring in the assembled matrix and vector. *Right:* the assembly of the stiffness matrix and the load vector is coded as a sum of the local contributions (with respect to the pattern coding, not to their values). The individual contributions can be reconstructed due to the distinct patterns chosen for the elements

6.2 Governing Equations

In our computational model the brain is considered to be a porous elastic medium saturated by a viscous fluid. The constituting relations are expressed in the form of coupled partial differential equations describing the diffusion process and the linear elasticity in a solid [152]. This dependence is essential. Compression of the medium results in increased pore pressure, thus affecting the internal flow of fluid. On the other side, an increased fluid pressure produces stress in the deformable solid matrix. This classical Biot formulation of this process relates the scalar field of the fluid pressure denoted by $p(\mathbf{x}, t)$ to the displacement vector field denoted by $\mathbf{u}(\mathbf{x}, t)$ at the position $\mathbf{x} \in \Omega$ at time $t \geq 0$.

In this section, the governing description of the poroelastic model is addressed. Thereby, the physical laws included in this model and their meaning are described. This provides an important and necessary insight into the theory which has been adapted within this thesis for the description of brain deformation.

6.2.1 Stress Distribution

In the first equation of the poroelasticity, the momentum balance equation is combined with Hooke's law for elastic deformation. We will establish the equations of

stress equilibrium for a quasi-static case, i.e. at some fixed time point. For simplicity, in the following considerations u stands for the time-dependent displacement vector $\mathbf{u}(\mathbf{x}, t)$.

By means of Hooke's law for an isotropic elastic body, the effective stresses are linearly related to the strains by the equation:

$$\sigma_{ij} = 2\mu e_{ij} + \lambda \delta_{ij} \nabla \cdot \mathbf{u} \ , \tag{6.9}$$

with e_{ij} denoting the strain tensor and δ_{ij} referring to the Kronecker delta.

For the linear elasticity, the strain components are related to the displacements by the equations:

$$\begin{aligned}
\epsilon_{xx} &= \frac{\partial u_x}{\partial x} \ , \\
\epsilon_{yy} &= \frac{\partial u_y}{\partial y} \ , \\
\epsilon_{zz} &= \frac{\partial u_z}{\partial z} \ , \\
\epsilon_{xy} &= \frac{1}{2}(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}) \ , \\
\epsilon_{yz} &= \frac{1}{2}(\frac{\partial u_y}{\partial z} + \frac{\partial u_z}{\partial y}) \ , \\
\epsilon_{zx} &= \frac{1}{2}(\frac{\partial u_z}{\partial x} + \frac{\partial u_x}{\partial z}) \ .
\end{aligned} \tag{6.10}$$

Let us consider an infinitesimally small elementary volume. In Figure 6.3, forces acting along the y-axis are presented. In the equilibrium state, all forces acting on the volume have to balance. Summing the respective values, we obtain:

$$\begin{aligned}
\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{yx}}{\partial y} + \frac{\partial \sigma_{zx}}{\partial z} - f_x &= 0 \ , \\
\frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{zy}}{\partial z} - f_y &= 0 \ , \\
\frac{\partial \sigma_{xz}}{\partial x} + \frac{\partial \sigma_{yz}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} - f_z &= 0 \ .
\end{aligned} \tag{6.11}$$

Combining Hooke's law from Equation 6.9, the relation between the strain and the displacements expressed in Equation 6.10 and the equilibrium condition from

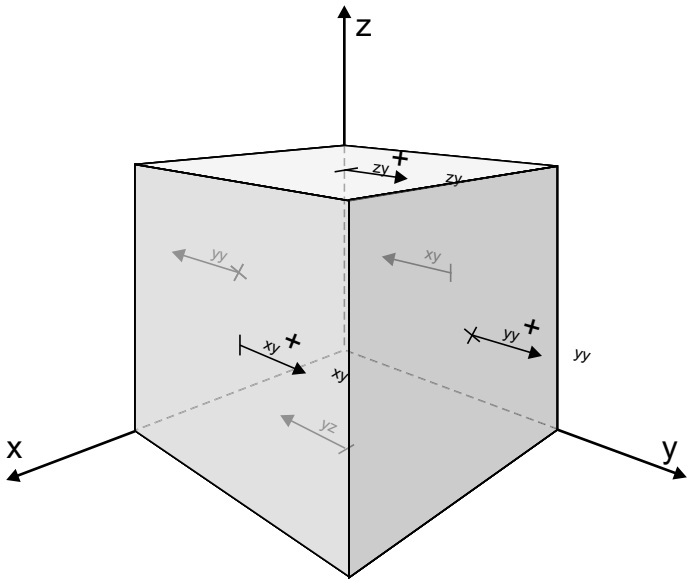

Figure 6.3: Stresses σ_{ij} acting upon the faces of an elementary volume, for an infinitesimally small volume $\Delta\sigma_{ij} = \frac{\partial\sigma_{ij}}{\partial j}$

the last equation, we find[4]:

$$\begin{aligned}
(\lambda+\mu)\frac{\partial\nabla\cdot\mathbf{u}}{\partial x} + \mu\nabla\cdot\frac{\partial\mathbf{u}}{\partial x} - f_x &= 0\,, \\
(\lambda+\mu)\frac{\partial\nabla\cdot\mathbf{u}}{\partial y} + \mu\nabla\cdot\frac{\partial\mathbf{u}}{\partial y} - f_y &= 0\,, \\
(\lambda+\mu)\frac{\partial\nabla\cdot\mathbf{u}}{\partial z} + \mu\nabla\cdot\frac{\partial\mathbf{u}}{\partial z} - f_z &= 0\,.
\end{aligned} \tag{6.12}$$

In the poroelastic model, the hydrated nature of the medium must be accounted for. The behavior of the solid is strongly influenced by the fluid pressure in the interstitial space. Let us denote by $p(\mathbf{x},t)$ the time-dependent pressure of the fluid in the pores and by $\alpha \geq 0$ a factor responsible for the mechanical coupling of the fluid pressure and the porous solid. The pressure can change dynamically with the time, however, for a fixed time point the gradient of the pressure can be treated as an acting force. Incorporating this hydraulic force into Equation 6.12 and considering the possible additional external volume-distributed forces $\mathbf{f}(\mathbf{x},t)$, we obtain the relation using a more compact vector notation, where $\Delta\mathbf{u}$ denotes the Laplace operator $\nabla\cdot\nabla\mathbf{u}$:

$$-(\lambda+\mu)\nabla(\nabla\cdot\mathbf{u}(\mathbf{x},t)) - \mu\Delta\mathbf{u}(\mathbf{x},t) + \alpha\nabla p(\mathbf{x},t) = \mathbf{f}(\mathbf{x},t)\,. \tag{6.13}$$

[4] $\nabla\cdot\mathbf{u}$ stands for $\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z}$.

6.2.2 Darcy's Law

The second equation of the poroelastic theory couples the volumetric strain to the fluid flow. According to Darcy's law, the volume of fluid flowing per second and unit area through a face of the elementary volume is related to the interstitial fluid pressure by the equations:

$$\begin{aligned} V_x &= -k\frac{\partial p}{\partial x} , \\ V_y &= -k\frac{\partial p}{\partial y} , \\ V_z &= -k\frac{\partial p}{\partial z} , \end{aligned} \tag{6.14}$$

where the physical constant $k \geq 0$ expresses the permeability of the medium. These three equations can be expressed with the use of a more compact vector notation as:

$$\nabla \mathbf{V} = -k\nabla\sigma \tag{6.15}$$

Let us consider a saturated solid. In such a case the total flux into the elementary volume has to be equal to zero (see Figure 6.4):

$$\nabla \cdot \mathbf{V} = 0 . \tag{6.16}$$

On the other hand, if we assume the incompressibility of the fluid, this fluid flow translates directly into a volumetric strain. Thus, as the pressure gradient decreases with the time (which is the immediate effect of drainage occurring during craniotomy), the displacement increases. Having a fully saturated tissue, this relation can be expressed by:

$$\frac{\partial \nabla \cdot \mathbf{u}}{\partial t} = \nabla \cdot k\nabla p . \tag{6.17}$$

In the general case, where the interstitial space is not fully saturated with the fluid, compression of the pores slows the pressure distribution. Additionally, possible fluid sources distributed within the volume should be considered. They are denoted by $\mathbf{h}(\mathbf{x}, t)$ in the general equation. These observations lead finally to the relation:

$$\frac{\partial}{\partial t}(c_0 p(\mathbf{x}, t) + \alpha\nabla \cdot \mathbf{u}(\mathbf{x}, t)) - \nabla \cdot k\nabla p(\mathbf{x}, t) = \mathbf{h}(\mathbf{x}, t) , \tag{6.18}$$

where the coefficient $c_0 \geq 0$ is proportional to the amount of fluid which can be forced into the medium by a pressure increment under conservation of the volume of the medium and $\alpha \geq 0$ is responsible for the mechanical coupling of the fluid pressure to the porous solid.

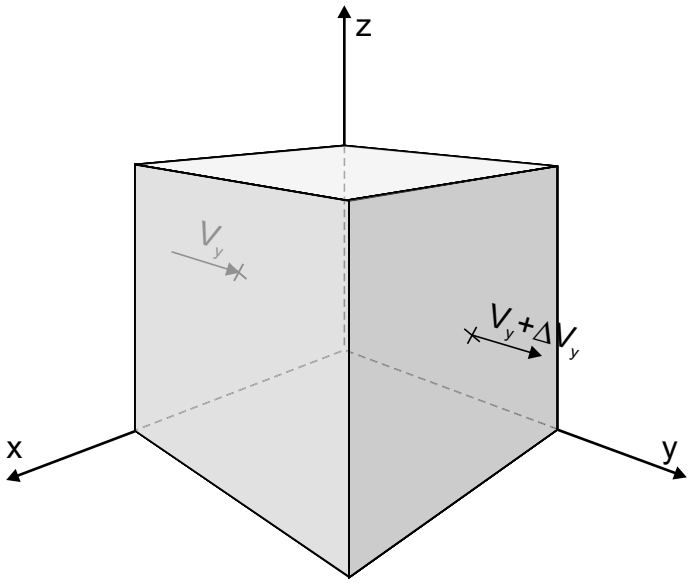

Figure 6.4: Conservation of the fluid volume V shown for the faces of an elementary volume perpendicular to the y-axis, for an infinitesimally small volume $\Delta V_y = \frac{\partial V_y}{\partial y}$

6.2.3 Boundary Conditions

Altogether, the governing system of poroelasticity differential equations consists of Equations 6.13 and 6.18. This equation system is time-dependent. Both equations are coupled and the unknowns are the three displacement components u_x, u_y, u_z and the interstitial pressure p. In order to solve the system, the boundary and initial conditions have to be specified. They can be of a simple type (Dirichlet boundary conditions), which results in portions of the boundary which are *clamped* ($\mathbf{u}(t) = 0$) or *drained* ($p(t) = 0$). Complementary boundary conditions involve the balance of forces on the *traction* boundary Γ_t and a balance of fluid mass on the *flux* boundary Γ_f.

It is possible to mix both types of conditions by prescribing two independent partitions of the boundary Γ: $\Gamma = \Gamma'_t \cup \Gamma_t,\ \Gamma'_t \cap \Gamma_t = \varnothing$ where displacement and tractions are defined and $\Gamma = \Gamma'_f \cup \Gamma_f,\ \Gamma'_f \cap \Gamma_f = \varnothing$ on which the pressure and flux are specified. This leads to four general boundary conditions:

$$\mathbf{u}(t) = 0 \ on\ \Gamma'_t\ , \qquad \sigma_{ij}(\mathbf{u}(t))n_j = g_i\ , 1 \leq i \leq 3\ on \Gamma_t\ , \tag{6.19a}$$

$$p(t) = 0 \ on\ \Gamma'_f\ , \qquad -\tfrac{\partial}{\partial t}(\mathbf{u}(t)\cdot\mathbf{n})\chi_s + k\tfrac{\partial p(t)}{\partial n} = h_1(t)\chi_s \ on\ \Gamma_f\ . \tag{6.19b}$$

The first boundary condition corresponds to a clamped portion Γ'_t of the boundary. The second condition expresses a balance of surface traction forces $\sigma_{ij}(\mathbf{u}(t))n_j$ determined by the unit outward normal vector $\mathbf{n}$ on Γ. The third condition means that at the drained boundary Γ'_f the pore pressure is equal zero. The last expression imposes a balance of fluid mass on the flow boundary Γ_f. Here, χ_s denotes

the characteristic function of the boundary $\Gamma_s = \Gamma_f \cap \Gamma_t$ which is neither drained nor clamped. Three of these boundary conditions refer to the solid medium where either the surface tractions or the displacement vector must be given. The fourth condition is related to the interstitial fluid where the flow along the direction normal to the boundary or the fluid pressure must be set.

6.3 Finite Element Discretization

For the specific problem of modeling brain tissue under deformation, additional assumptions have been made about the general model of poroelasticity. First, it is assumed that the brain is saturated with the interstitial fluid, which means that there is no empty space in the pores or, more intuitively, a change in pore pressure results in an immediate fluid flow. Second, the external forces have been incorporated into the boundary conditions. Under this conditions, the governing equations that have to be discretized can be formulated in the simplified form:

$$-(\lambda+\mu)\nabla(\nabla\cdot\mathbf{u}(\mathbf{x},t)) - \mu\Delta\mathbf{u}(\mathbf{x},t) + \nabla p(\mathbf{x},t) = 0\ , \tag{6.20a}$$

$$\frac{\partial}{\partial t}(\nabla\cdot\mathbf{u}(\mathbf{x},t)) - \nabla\cdot k\nabla p(\mathbf{x},t) = 0\ . \tag{6.20b}$$

In order to obtain a numerical solution for this system, the Galerkin version of the weighted residuals technique has to be applied to it. Furthermore, the derivative of time in the second equation is eliminated by integrating over the system of equations in time using the two-point weighting technique. The necessary transformations and calculations are technically complicated and therefore are detailed separately in Appendix A. According to those derivations, the governing equations can be expressed in a matrix form as:

$$\mathbf{A}^{\theta}\mathbf{x}_{n+1} - \mathbf{A}^{1-\theta}\mathbf{x}_n = \mathbf{b}^{\theta}_{n+1} - \mathbf{b}^{1-\theta}_{n}\ , \tag{6.21}$$

with $0 \leq \theta \leq 1$ being a given weight.

For a fixed time point, four unknowns (u_x, u_y, u_z and p) are related to each node of the finite element mesh. Therefore, the matrix entries a^{θ}_{ij} are submatrices of the size 4×4 and are given by the formula:

$$a^{\theta}_{ij} = \begin{bmatrix} \theta(\mu m + (\lambda+\mu)m_{xx}) & \theta(\lambda m_{xy} + \mu m_{yx}) & \theta(\lambda m_{xz} + \mu m_{zx}) & \theta m_x \\ \theta(\lambda m_{yx} + \mu m_{xy}) & \theta(\mu m + (\lambda+\mu)m_{yy}) & \theta(\lambda m_{yz} + \mu m_{zy}) & \theta m_y \\ \theta(\lambda m_{zx} + \mu m_{xz}) & \theta(\lambda m_{zy} + \mu m_{yz}) & \theta(\mu m + (\lambda+\mu)m_{zz}) & \theta m_z \\ m_x & m_y & m_z & \theta\Delta t k m \end{bmatrix} \tag{6.22}$$

$$m_{kl} = \int_{\Omega} \frac{\partial\varphi_i}{\partial k}\frac{\partial\varphi_j}{\partial l} d\Omega.\quad k \in \{x,y,z\},\ l \in \{x,y,z\}\ ,$$

$$m_k = \int_{\Omega} \varphi_i \frac{\partial\varphi_j}{\partial k} d\Omega,\ k \in \{x,y,z\}\ ,$$

$$m = m_{xx} + m_{yy} + m_{zz}\ .$$

The coefficients $b_i^\theta \in \mathbb{R}^{3\times 1}$ of the load vector $\mathbf{b}^\theta$ are expressed as:

$$b_i^\theta = \begin{bmatrix} (\oint_{\partial\Omega} \theta\sigma_s(t_{n+1}) \cdot \varphi_i d\vec{\Omega})_x \\ (\oint_{\partial\Omega} \theta\sigma_s(t_{n+1}) \cdot \varphi_i d\vec{\Omega})_y \\ (\oint_{\partial\Omega} \theta\sigma_s(t_{n+1}) \cdot \varphi_i d\vec{\Omega})_z \\ \Delta t \oint_{\partial\Omega} k\nabla p(t_{n+1}) \cdot \varphi_i d\vec{\Omega} \end{bmatrix} \tag{6.23}$$

The matrix coefficients $a_{ij}^{1-\theta}$ and the vector elements $b_{ij}^{1-\theta}$ have analogous forms.

6.4 Geometry Generation

The governing equations describe brain tissue under load. The unknowns are the displacement vector $\mathbf{u}$ and the interstitial pressure p. In order to solve these equations with the finite element method, a volumetric computational domain has to be provided corresponding to the already established discretized form of the equation system. In our approach, tetrahedral meshes are considered as the geometric primitives. Ensuring regularity here is very important for the convergence and stability of the FEM computations. For this purpose, a special technique for the mesh construction has been applied. In this section, the complete framework for tetrahedral mesh generation is described and the results of the application of the proposed scheme to volumetric discretization of complex brain geometry are discussed.

6.4.1 Extraction of Brain Volume from MR Data

The data for generation of a discrete geometry structure required for the brain shift simulation is established from MR head images for a given case. In the very first step, the brain volume is segmented from the scan in the form of a binary mask. For this purpose, a segmentation software Brain Extraction Tool (BET) [155] available as part of the FMRIB Software Library (FSL) is utilized. After performing this step, some additional manual correction of the segmentation mask is usually necessary. This task is accomplished with a customized software. In Figure 6.5, an exemplary output of the segmentation process is presented.

6.4.2 Triangular Surface Extraction

The brain segmentation mask resulting from the previous step divides the volume data into two regions. The background voxels are assigned a zero value and the voxels belonging to the extracted object have some non-zero value. Based on this binary information, a triangle surface is generated based on the strategy introduced by Labsik *et al.* [90]. First, the initial mesh is calculated with the Marching Cubes algorithm [94]. Applying the Marching Cubes algorithm to a dataset of the resolution typical for MR data ($512 \times 512 \times 160$ voxels) produces a triangle mesh

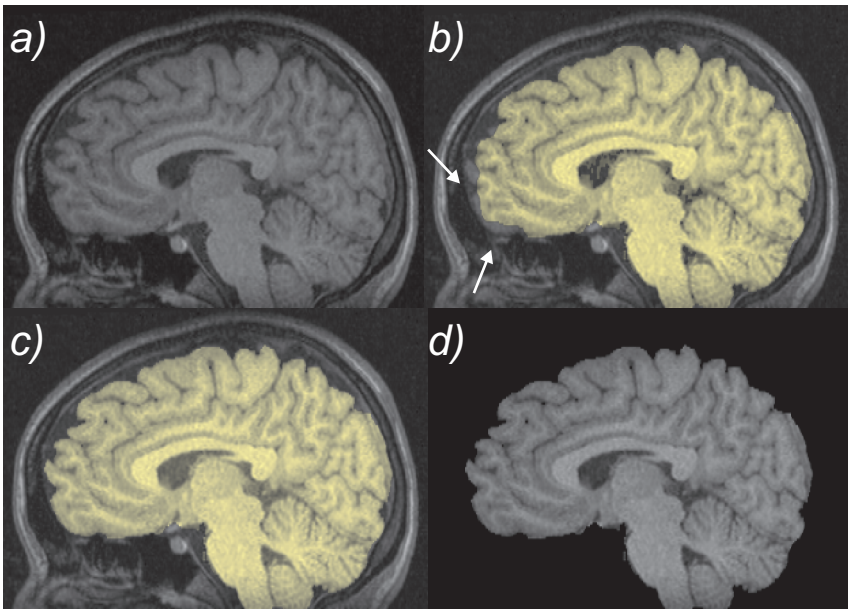

Figure 6.5: The brain segmentation process. *a)* A sagittal slice from an exemplary dataset. *b)* An overlayed view of the slice and the corresponding part of the segmentation mask obtained with the BET software (the mask is shown with brighter intensity). The arrows point to the locations, where the automatic segmentation did not yield satisfactory results. *c)* The segmentation mask after manual correction. *d)* The extracted brain by applying the corrected mask

consisting of millions of triangles. Thus, the initial mesh must be further processed. To decrease the number of triangles to the desired amount a mesh reduction is performed based on the algorithm by Garland and Heckbert [63]. In this method, so called *error quadrics* are associated with the vertices of the mesh. These quadrics store the approximation error between the original fine mesh and the reduced mesh.

A mesh reduction algorithm conducted directly on a huge mesh would be a very time consuming task. Instead, based on the segmentation mask, a hierarchy of downsampled volumes is created using dilation filtering. In this context, experiments showed that usually three hierarchy levels are sufficient. Subsequently, the initial mesh is extracted with the use of the Marching Cubes method from the volume on the coarsest level and only then is the mesh reduction performed.

Since the dilation filter has a growing effect, the base mesh is not identical to the surfaces obtained at the finer volume levels. Therefore, an optimization of the extracted mesh is required. First, all vertices with a valence of three are removed. Considering the volume hierarchy, an isosurface at level l is then projected onto the impliced isosurface at the next, finer level $l + 1$. The isosurface is found by trilinear interpolation in each voxel of the volume dataset. After this projection, an optimization is required since the initial mesh takes into account only the geometry

extracted at the coarser level. The complete procedure consists of three main steps (see Sections 6.4.2.1 - 6.4.2.3) which are executed several times in iteration:

- projection of the vertices onto the respective isosurface,
- relaxation of the vertices,
- hierarchical subdivision of the mesh.

6.4.2.1 Projection

Both relaxation of the vertices and surface subdivision produce vertices that no longer lie on the original isosurface. Also the dilation filter is responsible for the differences between isosurfaces calculated for the same isovalue, but on different hierarchy levels. Therefore, there is a need for a projection operator which would bring the incorrectly placed vertices back onto the respective isosurface. In this context, the normals to the surface going through the respective vertices are considered and the vertices are moved along these lines onto the surface. To establish the correct intersection with the isosurface, a distance gradient function is calculated to give the shortest signed distance between a vertex and the isosurface. Depending on the sign of this distance, it is decided in which direction of the surface normal the intersection point should be moved.

6.4.2.2 Relaxation

The relaxation operator improves the shape of the triangles. This is a very important prerequisite for the next step, where based on the triangular surface well shaped tetrahedra of approximately the same size are generated. The relaxation force leads to an even distribution of the vertices over the isosurface and therefore to triangles of good quality. Commonly, a discrete version of the Laplacian is taken for smoothing, however, it has a shrinking effect on the mesh thus moving vertices off the isosurface. Instead, we use only the tangential part $T(\mathbf{v})$ of the Laplacian $L(\mathbf{v})$:

$$T(\mathbf{v}) = L(\mathbf{v}) - <L(\mathbf{v})|\mathbf{n}_{mathbfv}> \mathbf{n}_{\mathbf{v}} \ , \tag{6.24}$$

where $\mathbf{n_v}$ is the surface normal vector at $\mathbf{v}$ computed by normalizing the curvature normal vector:

$$\kappa(\mathbf{v}) = \sum_{\mathbf{w} \in N_v} (\cot \alpha_w + \cot \beta_w)(\mathbf{v} - \mathbf{w}) \ , \tag{6.25}$$

with α_w and β_w are the angles opposite to the edge $\overline{\mathbf{vw}}$ in the adjacent triangles.

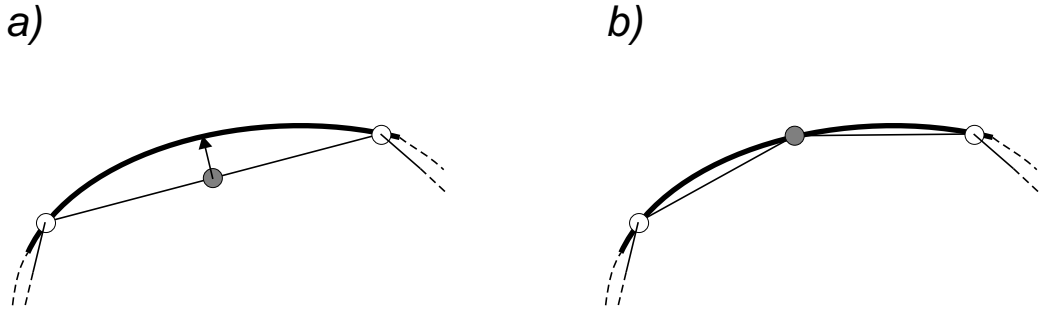

Figure 6.6: Projection onto an isosurface after refinement. *a)* A new vertex is introduced at the midpoint of an edge. *b)* After projection of the new vertex along surface normal a better approximation is achieved

6.4.2.3 Hierarchical Refinement

In order to capture all the details of the isosurface at the finest hierarchy level, the meshes extracted and processed at the coarser levels have to be refined. By applying uniform subdivision, three new vertices are introduced for a triangle at the midpoints of the edges and a hierarchy of subsequently refined meshes is constructed. One or more subdivision steps can be applied at each level of the downsampled volume hierarchy. Newly generated vertices usually do not lie on the isosurface. To overcome this problem, the previously described projection operator is employed to ensure the correct location of the vertices. In Figure 6.6 a 2D example of a projection after the refinement is presented.

With the use of the hierarchy of meshes, the boundary surface of the tetrahedral grid is approached with an increasing precision. In Figure 6.7 the mesh generation process is depicted. In this example, five uniform subdivision steps were performed in order to obtain the final geometric representation. It can be observed how the details of the complex brain geometry are progressively added with the increasing mesh resolution.

The final surface is characterized by a very high level of detail. The important anatomical structures at the cortex are well captured, thus establishing a good correspondence between the model and the anatomy of the patient (see Figure 6.8).

6.4.3 Tetrahedrization

In Section 6.4.2, a method for constructing triangular surfaces from the volumetric discrete datasets has been presented. There, the emphasis in the method was on ensuring that the resulting triangle mesh consists of well shaped primitives. To create a volumetric computational domain for the finite element simulation, in a second step a tetrahedrization of the region enclosed by the generated surface is accomplished. For this purpose, the commercially available software AMIRA (Mercury

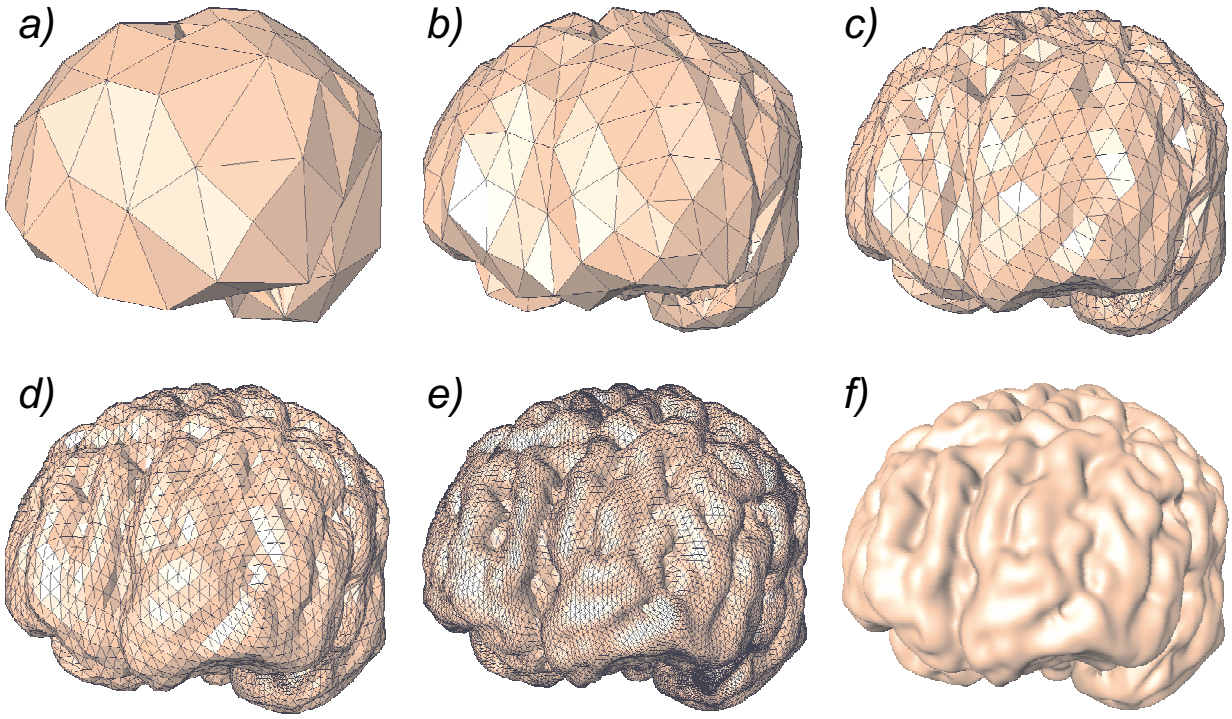

Figure 6.7: Extraction of a semi-regular mesh of the brain. *a)* Triangular mesh extracted and processed at the coarsest level. *b)-e)* Hierarchy of uniformly subdivided meshes. *f)* The final mesh after five subdivision steps. In the last image the edges of the triangle mesh are not shown for clarity

Computer Systems) is used. having a regular triangle mesh as its input is crucial to the generation of the grids consisting of well shaped tetrahedra.

In Figure 6.9a), a histogram corresponding to the value distribution of the dihedral angles[5] in the tetrahedral mesh is exposed, proving a high quality of the generated grids. The optimal value for this angle (for an equilateral tetrahedron) is about 70 degrees. The next distribution plot refers to the solid angle[6] (this angle has a value of about 30 degrees for an equilateral tetrahedron). In both plots, a very high regularity in both angles was observed. Finally, the distribution of the edge length in the grid is presented in the histogram of Figure 6.9c). A significant peak was observed around the value of 5 mm which proves the high quality of the generated grid. Such regularity of the geometric primitives in the finite element method is essential for performing stable numerical computations.

[5]For each tetrahedron edge the dihedral angles are defined as the angles between the two adjacent faces.

[6]For each tetrahedron vertex the solid angle is defined as the part of the unit sphere centered at this vertex which is occupied by the tetrahedron.

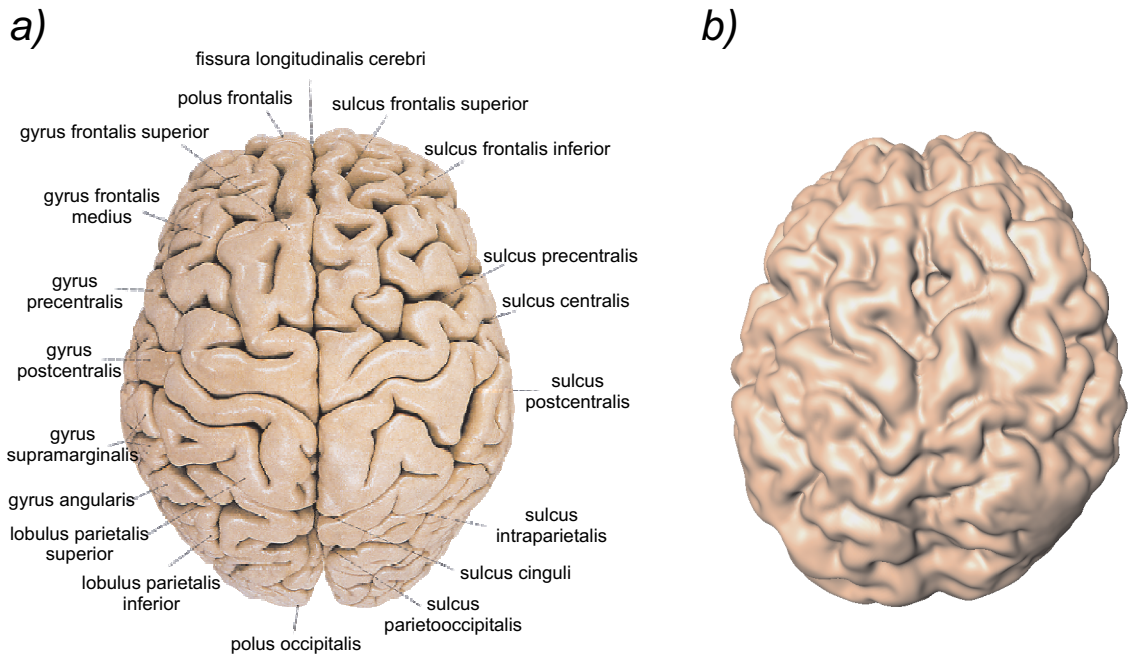

Figure 6.8: Anatomical details of the brain surface. *a)* A description of the important cortical structures taken from an anatomy atlas [156]. *b)* The brain surface captured with the proposed framework. A very good correspondence between extracted surface and the image from the atlas is visible

6.5 Boundary Conditions

In order to incorporate the intraoperative settings into the numerical model of brain shift, the forces which affect the brain during surgery have to be defined with respect to the generated geometry. In [176] non-rigid registration of MR images acquired before and after compressing a volunteer's breast was employed to derive image-based forces. In our model, the load vector is set according to the location of the skull opening. Since this information is only included in the intraoperative dataset, a rigid registration of the pre- and intraoperative MR scan is performed with the approach presented in Chapter 3. Then, in a 3D editor supporting the fusion of the preoperative brain surface and the intraoperative MR data, the respective outer part of the generated volumetric grid is manually marked (see Figure 6.10). Subsequently, a separate boundary condition is specified for the defined surface region.

6.6 Volume Reconstruction

The direct result of the numerical simulation of brain shift is a vector field of displacements which are defined only at the vertices of the tetrahedral grid describing

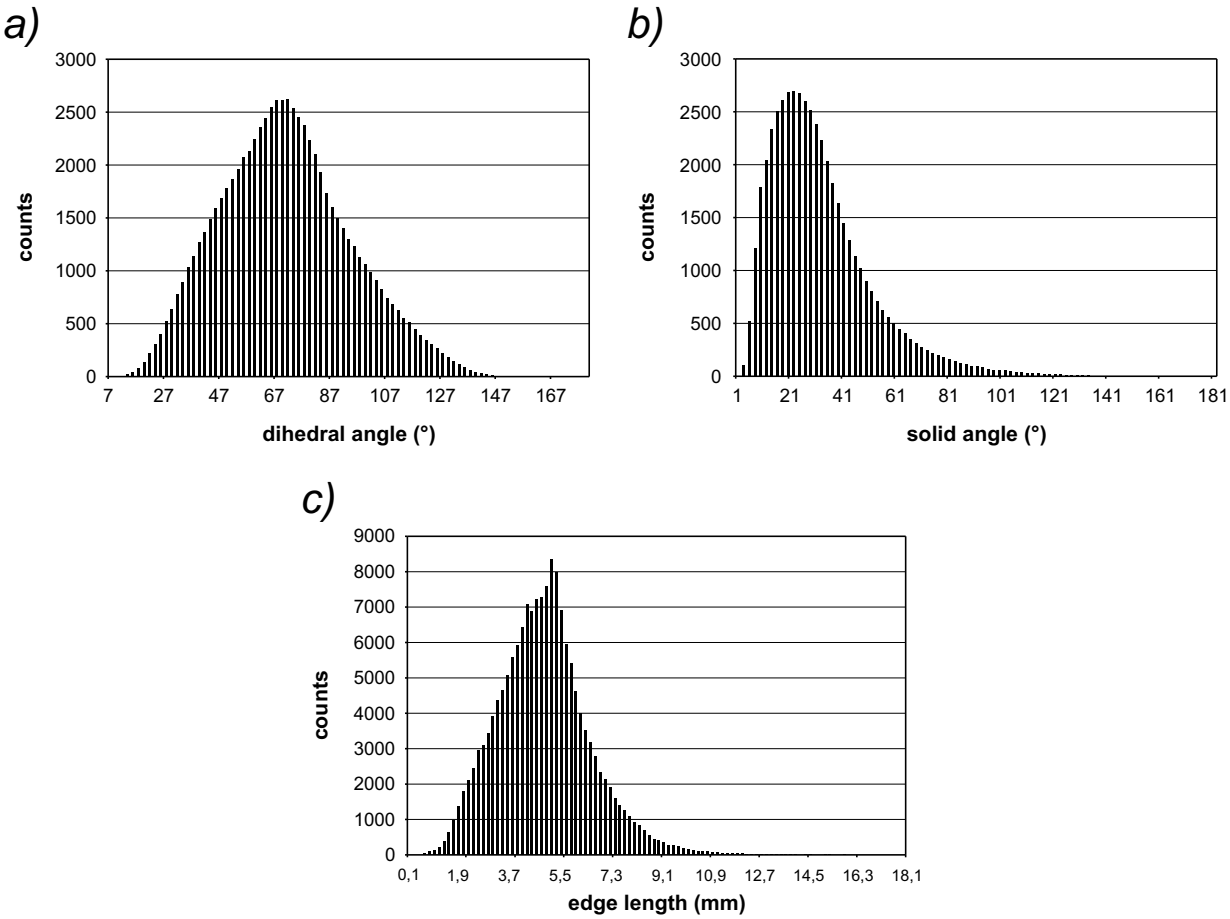

Figure 6.9: Histograms showing the quality of the generated tetrahedral grids. The plots were generated for a mesh consisting of 123496 elements

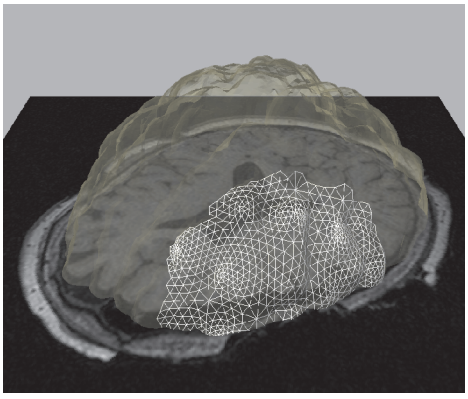

Figure 6.10: The region of the boundary condition is manually marked in a 3D editor according to the location of the skull opening. The respective and defined part of the surface is visualized by displaying the underlying grid. The remaining brain surface is shown in a semi-transparent representation

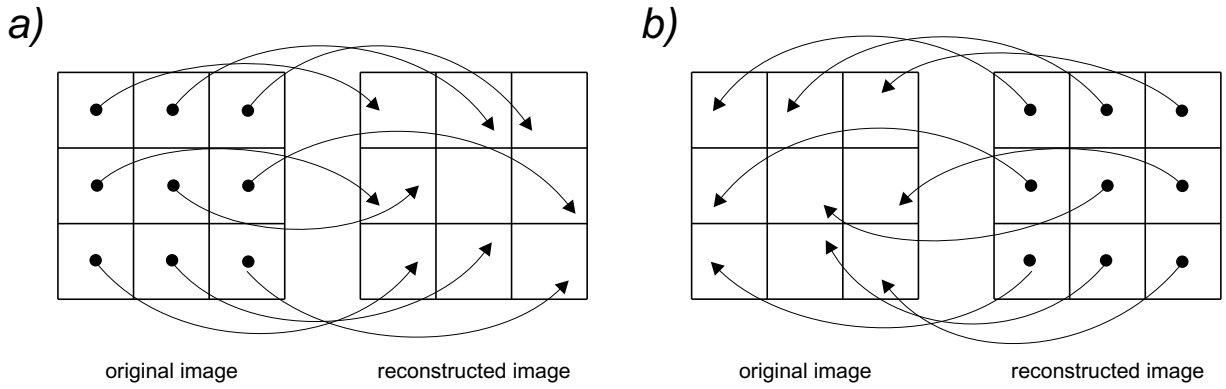

Figure 6.11: 2D example of image reconstruction techniques. *a)* Forward propagation produces holes in the reconstructed image. *b)* Backward propagation does not exhibit this effect

the preoperative brain geometry. In order to obtain a simulated 3D image of the brain after deformation, intensities in the preoperative dataset have to be reconstructed according to the calculated grid deformation. For this purpose, all tetrahedra in the deformed grid are traversed and for each element its inner voxels are determined. The displacement for each voxel is calculated as a scalar product of its barycentric coordinates and of the displacements given in the four nodes of the tetrahedron the voxel belongs to. Finally, a new intensity value is computed by moving back the voxel location after the deformation and by trilinear interpolation of the intensities in the preoperative MR volume. This back-projection technique, contrary to forward propagation, prevents the formation of holes in the reconstructed dataset (see Figure 6.11).

6.7 Results

6.7.1 Synthetic Dataset

The behavior of the proposed poroelastic model was analyzed with a synthetic data. Here, an isotropic cube of size 100 mm $\times$ 100 mm $\times$ 100 mm and consisting of 196608 tetrahedra was considered. The object was assumed to be a linearly elastic medium with pores saturated with fluid.

In the experiments, the compression of the cube was simulated with respect to different values of Young's modulus E and Poisson's ratio ν. As a boundary condition, a constant force was applied downwards in the direction perpendicular to the top surface of the cube. The top surface was set to be drained. Further, the cube was clamped on its bottom face where no fluid flow was allowed.

A visualization of the results from the experiments is shown in Figure 6.12. The original undeformed cube overlayed with the outer surface of the underlying tetra-

hedral mesh is presented in Figure 6.12a). Figure 6.12b) shows the deformed cube for $E = 8000$ Pa and $\nu = 0.46$. An expansion in the direction perpendicular to that of the applied force can be observed which is due to a chosen, small compressibility of the material $\nu = 0.46$. In comparison, in Figure 6.12c) a stronger deformation of the object is visible, which is due to a smaller value of Young's modulus ($E = 6000$ Pa). Finally, Figure 6.12d) includes a visualization of the simulation for a highly compressible material ($\nu = 0.05$). Young's modulus was here set to $E = 8000$ Pa. A linear compression of the cube towards its bottom can be observed while the cube walls are still contained within the bounding box of the undeformed object. Such behavior is characteristic for very compressible media.

During these experiments with synthetic data, an experience about the behavior of the described poroelastic model was gained. This experience was crucial for setting model parameters appropriately for experiments with clinical data of the human brain.

6.7.2 Clinical experiments

The described biomechanical model was validated in experiments conducted with real data from patients. Five pairs of MR T1-weighted scans of the human head taken before and after craniotomy were considered. All scans were obtained with a Siemens Magnetom Sonata Maestro Class 1.5 Tesla scanner and had a resolution of 512 $\times$ 512 $\times$ 160 voxels of size 0.49 mm $\times$ 0.49 mm $\times$ 1.0 mm. Significant brain shift was observed in all patients. The selected datasets were the same as those used in Chapters 3 and 4 for the evaluation of the intensity-based image registration techniques. Thus, a direct comparison of the quality of different methods of image alignment can be conducted.

In a preprocessing step, each pair of the datasets was first rigidly registered. Then, brain extraction, surface generation and tetrahedrization of the preoperative brain volumes were performed for each experiment. The generated tetrahedral meshes consisted of about 500 000 elements each. This resulted in a mean edge length of approximately 1 mm, which is very fine with respect to the underlying voxel resolution. Alignment of the pre- and intraoperative datasets allowed retrospective modeling of the forces which had acted upon the brain after craniotomy. Appropriate regions were marked according to the size and position of the skull opening and the respective boundary conditions were set. This ensured a good correspondence between the intraoperative situation and the model.

Then, matrix and vector assembly was conducted. For elasticity parameters values of $\nu = 0.458$ for Poisson's ratio and $E = 9921$ Pa for Young's modulus were employed, based on the findings of Chapter 7. The linear equation system was preconditioned with an incomplete LU (ILU) factorization scheme [8] and subsequently solved with the Generalized Minimum Residual Method (GMRES) [143] since this technique is very efficient for square, large and sparse matrices. The maximum

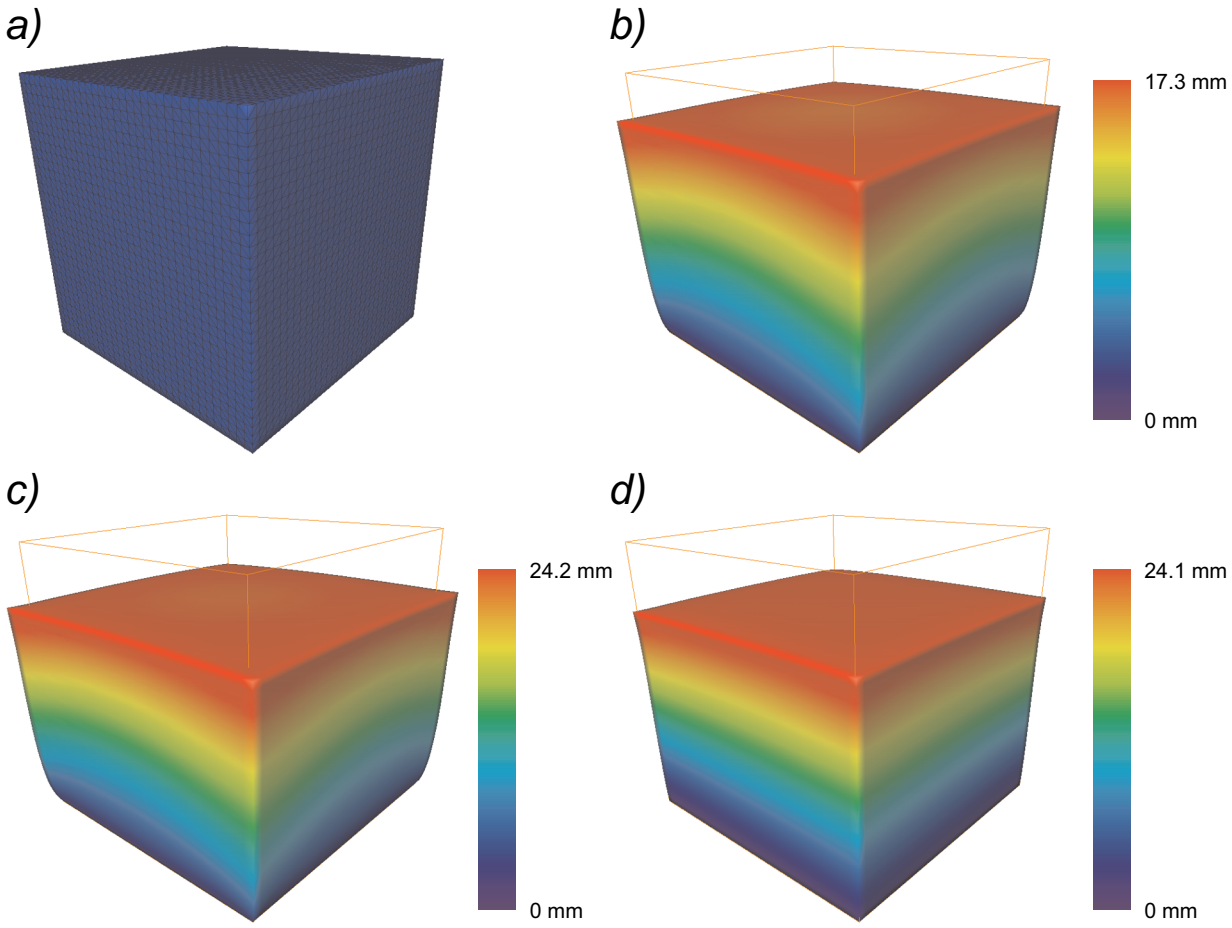

Figure 6.12: Deformation of synthetic data. *a)* An undeformed cube superimposed with the underlying geometry structure. *b)* View of the deformed cube for $E = 8000$ Pa and $\nu = 0.46$. *c)* Deformation for $E = 6000$ Pa and $\nu = 0.46$. *d)* Compression of the cube for $E = 8000$ Pa and $\nu = 0.05$. The displacements resulting from the compression are color-coded. The bounding box of the undeformed state is depicted for comparison

Table 6.1: Point errors after performing biomechanical simulation of intraoperative brain deformation. RMS, VAR and STD denote the root-mean-square error, the variance and the standard deviation, respectively

Patient No	Point 1	Point 2	Point 3	Point 4	Point 5	Point 6	RMS error	VAR σ^2	STD σ
1	2.5	1.803	1.581	2.739	1.871	2.062	2.131	0.196	0.443
2	2.915	1.581	3.202	2.345	3.841	2.958	2.894	0.595	0.771
3	2.915	3.354	2.828	3.742	3.202	3.354	3.123	0.179	0.424
4	1.581	1.5	1.118	1.581	1.0	0.707	1.291	0.131	0.362
5	3.536	2.121	2.550	1.5	1.0	4.637	2.836	1.803	1.343

number of outer iterations was set to 5000, the *restart* parameter limiting the number of inner iterations was chosen to be 330 and the convergence tolerance *tol* was defined as 10^{-4}. For the time stepping a semi-implicit scheme ($\theta = 0.8$) was used.

After simulation, the geometry of the brain was updated according to the resulting displacement vector field and volumetric image reconstruction was performed. Based on the reconstructed images and on the available intraoperative images, a subsequent quantitative analysis of the results from the experiments was conducted. For this purpose, several anatomical features were selected in the intraoperative data and in the corresponding image reconstructed from the simulation. In order to compare the quality of the alignment to the results obtained with the image registration methods, the marked features were identical to those chosen in Section 4.2.8. The distances between the selected points before simulation and the statistical terms of root-mean-square (RMS) error, variance (VAR) and standard deviation (STD) have been already presented in Table 4.3. The corresponding values established after biomechanical deformation are collected in Table 6.1.

In addition to the measurements, the results from the simulation were inspected visually. Contours extracted from the simulated image were superimposed with two-dimensional views through real intraoperative data. In Figure 6.13 such a visualization is presented for Patient 1. Corresponding views of the preoperative data and of simulated intensity images are also included.

Furthermore, in order to assess the overall extent and direction of the brain shift, for each experiment the calculated displacements were mapped onto the respective surface representations of the brain using a color-coding technique. An example of this visualization is shown in Figure 6.14 with a respective scale expressed in mm.

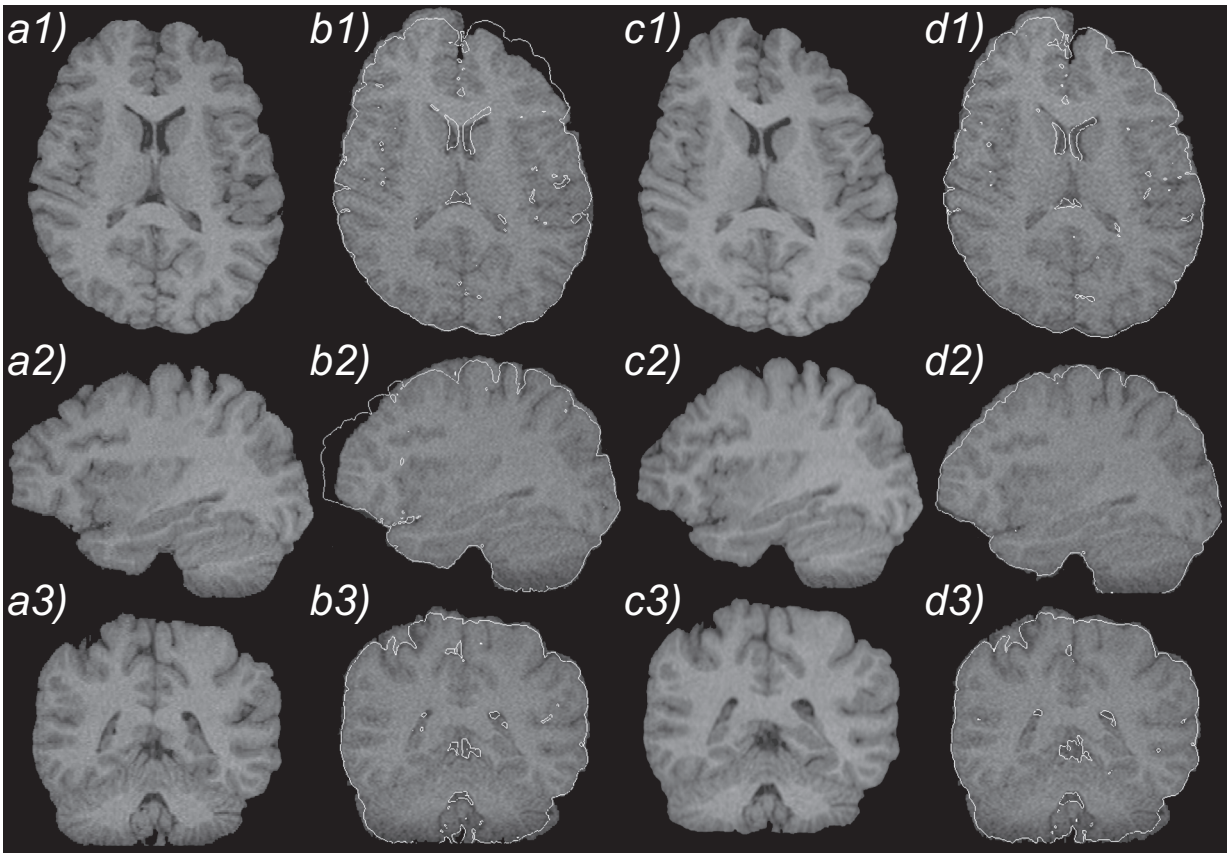

Figure 6.13: Two-dimensional visualization of the simulation results. *a)* A preoperative slice image. *b)* The corresponding intraoperative slice overlayed with the contours extracted from *a)*. *c)* The corresponding slice from the image reconstructed from the simulation. *d)* The intraoperative image superimposed with the contours of the simulated image. *Top:* Axial, *middle:* sagittal and *bottom:* coronal views

6.8 Discussion

The experiments conducted showed the quality of the proposed approach. The computations exhibited numerical stability. In all experiments, the iterative solution method converged within the specified tolerance. On one hand, this is due to a special technique which ensured the regularity of the computational domain in the form of a tetrahedral grid. On the other hand, the appropriate choice of values of 9921 Pa for Young's modulus and 0.458 for Poisson's ratio ensured the stability of the system.

Visual comparison of the simulated images to the real intraoperative data confirmed the good quality of the simulation. In all cases, the model behaved as expected. The direction and extent of the brain shift correlated well with the values from the intraoperative image. The most significant displacements occurred at

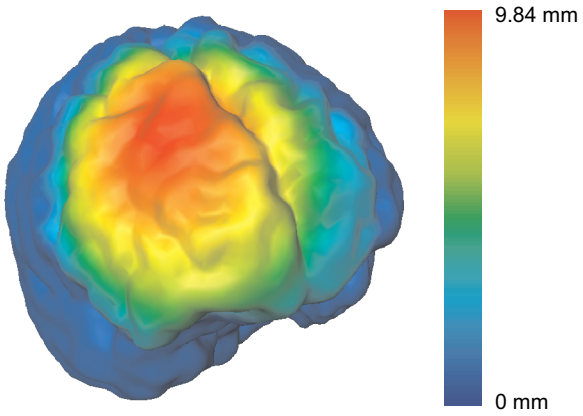

Figure 6.14: Visualization of the simulation results. The magnitude of the displacement is projected onto the brain surface using a color-coded technique. In the regions colored red the highest deformation occurred, blue color means no deformation

the regions where the respective skull had been opened, decreasing smoothly until reaching the clamped boundary (see Figure 6.14).

In all cases, the preoperative data was modified and the registration error was measured after the numerical calculations. At best, this error was lowered to 1.29 mm (for Patient 4) compared to the value of 5.04 mm after only rigid transformation. In the worst case (Patient 3), the RMS error amounted to 3.12 mm which still means an improvement of 6.05 mm with respect to the error measured for the rigidly aligned but undeformed preoperative image. In comparison to the results measured for the non-linear registration approach using Bézier functions, the alignment accuracy achieved with the simulation model was about 0.19 - 2.06 mm less in terms of the RMS error. Nevertheless, simulation reduced the misalignment by a factor of 2.20 - 3.91. Specifically, visual analysis of the results revealed that although the shape of the ventricles could be significantly improved after the simulation, there is a potential for further improvement (see Figure 6.13b1) and d1) in that region. This can be accomplished by an explicit physical modeling of the ventricles in the biomechanical model. In Figure 6.13d1), a misalignment is visible in the left frontal brain lobe. This observation indicates a need to incorporate the falx cerebri[7] into the biomechanical model. As expected, this would help to avoid such undesirable effects.

Overall, considering the fact that, contrary to direct registration methods, the simulation is based only on the preoperative data, the results are very satisfactory. In all cases brain shift could be successfully predicted and a good correspondence between the real intraoperative image and the image produced with the simula-

[7]The scythe-shaped fold of dura matter in the longitudinal fissure between the two cerebral hemispheres.

tion technique was observed. In the simulation, the intraoperative data was only employed to define appropriate boundary conditions reflecting the real surgical situation. No explicit image information from the intraoperative data was utilized as in the case of intensity-based registration where intraoperative imagery was defined *a priori* as the target for the deformed preoperative data. This is an important strength of the simulation framework, which allows its application in environments without an intraoperative MR scanner.

6.9 Conclusion

In this chapter, a comprehensive simulation framework for modeling the intraoperative brain deformation has been introduced. Within this model, the brain is treated as a linearly elastic medium saturated with an interstitial fluid. This three-dimensional finite element model for predicting intraoperative brain deformation has been quantified using an experimental protocol considering pre- and intraoperative MR head data acquired before and during an open-skull surgery. Appropriate boundary conditions reflecting the intraoperative environment were set with the help of the algorithm for rigid registration presented in Chapter 3. The evaluation showed that the considered poroelasticity model describes the behavior of the brain under load well and is able to predict the intraoperative brain movement with good precision.

Overall, the conducted experiments showed that the approach is capable of being employed for imagery update during an image-guided neurosurgery. This application is especially important if no intraoperative imaging is available. In such environments, the surgery procedure can be significantly improved with the use of the proposed model. Furthermore, the model can be used for preoperative prediction of the direction and the magnitude of brain shift, thus supporting the surgeon during the planning of the surgery.

There is still some potential for improvement of the presented model. However, incorporation of all factors into a physically-based simulation is a very complex task. Apart from the biomechanical properties of the brain tissue and the setting of appropriate boundary forces, all surgical events (e.g. tissue resection) having an influence on the shape of the brain could be modeled. This would require recording these events during a surgery. Altogether, the model presented here is very encouraging fundamental research that can be extended and applied to improve the process of image-guided surgery.

Chapter 7

Determination of the Mechanical Properties of the Brain

In the last chapters, a framework for the simulation of intraoperative brain deformation has been presented and validated in a series of experiments. The physics of the deformation process has been modeled in terms of partial differential equations describing tissue as a poroelastic medium. Generally, existing algorithms of physically-based brain shift simulation can be categorized according to the underlying mechanical model of the tissue. Various approaches were discussed in Chapter 5.

Independent of the general mathematical description of the properties of brain tissue considered, all simulation methods strongly rely on the choice of some constants related to the stiffness of the brain. Even the most sophisticated algorithms are limited without precise information about the brain elasticity. Especially in the field of neurosurgery, precision of the applied computer systems is one of the most important requirements. Choosing reliable values correctly describing the behavior of the brain is thus an important issue in brain shift simulation.

In this chapter, a framework for the automatic determination of the mechanical properties of brain tissue is introduced. The approach is based on registering the images obtained as a result of a 3D biomechanical simulation of brain shift and the corresponding intraoperative MR data. Since the intraoperative data provides images of the tissue after movement, validation is an integral part of the method.

A review of related work dealing with the estimation of the elasticity of tissue is the subject of Section 7.1. Section 7.2 presents the main components of our algorithm. In this context, a similarity function is addressed in Section 7.2.1 and the biomechanical model and the optimization procedure are described in Sections 7.2.2 and 7.2.3, respectively. A statistical analysis of the numerical results of the algorithm is presented in Section 7.3. This chapters closes with Section 7.4 extensively discussing the value of the presented method and concluding remarks are made in Section 7.5.

7.1 Previous Work

Biomechanical characterization of soft tissue has been a subject of intensive investigation in recent years. Miller *et al.* validated in a series of model *in vivo* experiments the mechanical properties of porcine brain tissue previously obtained *in vitro* [115]. The forces reproduced within their model differed from those actually recorded in the experiments by on average 31 %. This can be explained, among others, by the fact that living tissue exhibits different stiffness properties than deceased tissue.

In recent years, diagnostic imaging modalities have attracted significant attention for measuring the response of tissue *in vivo* to various types of loadings. In this context, above all magnetic resonance (MR) and ultrasound (US) imaging have been extensively investigated. US-based tissue elasticity imaging methods can be split into two main groups. Static methods compare images of the tissue before and after displacement. As an example for this technique, US waves were considered for detecting the internal motion of normal and pathological breast and prostate tissues resulting from a static mechanical stimulus [87] (here *in vitro*). Dynamic approaches use the Doppler effect to determine the tissue motion resulting from vibration. Here, researchers applied US to observe the behavior of tissue subjected to a low-frequency vibration [146].

In the case of US, however, poor resolution in the off-axial directions is one of the limiting factors. Therefore, MR was applied in investigations of the shear modulus of the brain *in vivo* [100]. A very comprehensive review of similar works on measuring tissue stiffness was given in [124]. Low availability and difficulties with the integration into clinical practice are, however, the main disadvantages of such techniques.

At the same time, methods for the implicit determination of the mechanical properties of the tissue have been the focus of numerous investigations. Dependencies between the Lamé elastic constants which were reported in various works were exhibited and discussed by Hagemann *et al.* [69]. The influence of different elastic values describing the stiffness of breast tissue was inspected by Tanner *et al.* [176] with respect to the accuracy of their proposed biomechanical breast model. Also, within the context of mammography, Young's moduli for 2D cross sectional MR slices of breast tissue were reconstructed by Miga *et al.* in a function minimization framework using regional measures of image similarity.

It is, however, crucial for the simulation quality to incorporate both Young's modulus E and Poisson's ratio ν in the reconstructed elastic parameters of tissue since their combination is very important for the physical behavior of the tissue. The own method presented in this chapter accounts for this issue and calculates both elastic moduli.

7.2 Method

7.2.1 Similarity Function

The method for the determination of Young's modulus E and Poisson's ratio ν for the brain tissue is based on an iterative process. A biomechanical model of brain shift interacts with the image information contained in the MR data of the head acquired before and after craniotomy. During optimization, in each iteration a volumetric dataset is reconstructed according to the displacement vector field resulting from the numerical simulation of the intraoperative brain movement (see Section 7.2.2). This image reconstruction is thereby parametrized by both of the elastic moduli of tissue E and ν. The main idea of the presented framework is to automatically vary these elastic constants of the brain in such a way that the correspondence between the reconstructed deformed 3D image of the brain and the brain image in the intraoperatively acquired scan is maximized.

The measure of this correspondence is defined based on the normalized mutual information (NMI) [172] which relates the grey value distributions in both datasets in terms of entropy. The similarity measure denoted by $\mathrm{S}_{sim,intra}$ used in our framework extends the NMI and is expressed in dependence on Young's modulus E and Poisson's ratio ν. Both of these elastic moduli are set as free parameters in an optimization procedure maximizing the value of the constructed similarity function which is formulated as:

$$\begin{aligned} \mathbf{S}_{sim,intra}(E,\nu) &= NMI_{sim,intra}(E,\nu) + \alpha_0 < E_{low}, E_{high} > (E) + \\ &\quad \alpha_1 < \nu_{low}, \nu_{high} > (\nu) \,, \end{aligned} \tag{7.1}$$

with $\alpha_0 < E_{low}, E_{high} > (E)$ representing a penalty term restricting the allowed values for E and defined by:

$$\alpha_0 < E_{low}, E_{high} > (E) = \begin{cases} c_0 & : \quad E \notin < E_{low}, E_{high} > \,, \; c_0 \leq 0 \\ 0 & : \quad E \in < E_{low}, E_{high} > \end{cases} \,. \tag{7.2}$$

This term constrains with the use of the penalty constant c_0 the search space for Young's modulus E to be between the upper and lower limits E_{high} and E_{low}. The constant c_0 is thereby set to a sufficiently low negative value to ensure that the final solution takes a value from the specified interval. Analogously, the reasonable limits for Poisson's ratio ν can be specified with the term $\alpha_1 < \nu_{low}, \nu_{high} > (\nu)$. These constraints allow the incorporation of expert knowledge about the reasonable intervals for the elastic parameters in the optimization process. This prevents the finding of unstable simulation configurations and reduces the number of iterations required for the determination of the optimal values for E and ν. Thus, the computational expense for the optimization is shortened. Note, that both restriction terms can be skipped by setting $c_0 = 0$, which allows optimization over the whole parameter space.

7.2.2 Biomechanical Model and Volume Reconstruction

During the optimization procedure, the value of the function $\mathbf{S}_{sim,intra}$ is computed in each iteration according to Equation 7.1. This value corresponds to the level of similarity between the volume reconstructed from the physically-based simulation and the corresponding intraoperative MRI image. Generally, the underlying physical model of the soft tissue deformation can be chosen arbitrarily. Here, according to our previous works, a simplified set of equations describing the behavior of poroelastic materials under load was considered. The detailed description of this model and the setting of proper boundary conditions has been addressed in Chapter 6.

In order to calculate the value of the function $\mathrm{S}_{sim,intra}$, the normalized mutual information between the reconstructed image and the intraoperative dataset has to be evaluated. For this purpose, intensities in the deformed volume have to be recomputed based on the displacement field resulting from the simulation. This step is performed by the volume reconstruction algorithm described in Section 6.6.

7.2.3 Minimization Framework

Similar to the strategy presented in Section 6.5, appropriate boundary conditions are defined according to the location of the skull opening. This step is accomplished based on the rigid registration of the pre- and intraoperative images. After the boundary conditions have been set, the parameters E and ν are initialized and the optimization is started. In each iteration only one parameter is changed at a time. Then, a biomechanical simulation is performed using the current values of the elastic moduli and the new deformed volume resulting from the simulation is computed. Subsequently, the value of the function $\mathrm{S}_{sim,intra}$ is calculated. This iterative procedure terminates when the similarity function $\mathrm{S}_{sim,intra}$ reaches its maximum within a given precision. The parameter search is performed automatically using Powell's direction search method [133].

7.3 Results

To assess the value of the proposed method and to determine the elastic parameters for the brain tissue, a series of computational experiments was conducted. In the experiments, six pairs of T1-weighted pre- and intraoperative MR scans were processed. All scans were obtained with a Siemens Magnetom Sonata Maestro Class 1.5 Tesla scanner and had a resolution of $512 \times 512 \times 160$ voxels and a voxel size of 0.49 mm $\times$ 0.49 mm $\times$ 1.0 mm. Significant changes in brain shape between the pre- and the corresponding intraoperative scan were observed in all patients.

Prior to any optimization iteration, in all experiments the corresponding pre- and intraoperative datasets were firstly rigidly registered. Then, the brain volumes were extracted from both images and a tetrahedral grid was generated for the preop-

Table 7.1: The optimal values of the elastic parameters recorded in six optimization runs. For each experiment the resolution of the mesh is provided

Patient No	# of tetrahedra	Young's modulus E	Poisson's ratio ν
1	9952	8863.01 Pa	0.452
2	123496	8196.21 Pa	0.461
3	10919	9921.20 Pa	0.459
4	32698	8780.61 Pa	0.461
5	112322	10231.05 Pa	0.457
6	545235	9116.24 Pa	0.460

Table 7.2: A statistical analysis of the calculated elastic moduli. The root-mean-square error (RMS), the variance (VAR) and the standard deviation (STD) were computed

	Root-mean-square error (RMS)	Variance (VAR)	Standard deviation (STD)
Young's modulus E	9210.87 Pa	577170.60 Pa	759.72 Pa
Poisson's ratio ν	0.458344	0.000012	0.003445

erative brain volume. Subsequently, appropriate boundary conditions were defined on the generated grids according to the location of the craniotomy.

For each pair of scans, the optimization algorithm was then executed. The initial values of Young's modulus E and Poisson's ratio ν were identical throughout all experiments and were set as 6000 Pa and 0.40, respectively. The search space for Young's modulus and for Poisson's ratio was restricted to the intervals $< 1500, 12000 >$ and $< 0.3, 0.49 >$, respectively. The penalty constant c_0 was set in all cases to -1.0.

For each optimization run the optimal elastic parameters were recorded. The values collected from all experiments are presented in Table 7.1. The obtained values were also analyzed by means of statistics. The results of this analysis are collected in Table 7.2.

Furthermore, the maxima of the similarity function $\mathrm{S}_{sim,intra}$ were investigated. On the one hand, the similarity function was plotted in the form of a projection on one of the optimal parameters, one at a time. Figure 7.1 shows exemplary plots (for Patient 3). Additionally, the similarity measure was equidistantly sampled on part of the two-dimensional domain corresponding to the two elasticity parameters. The recorded values are presented in Figure 7.2 in the form of a color-coded diagram

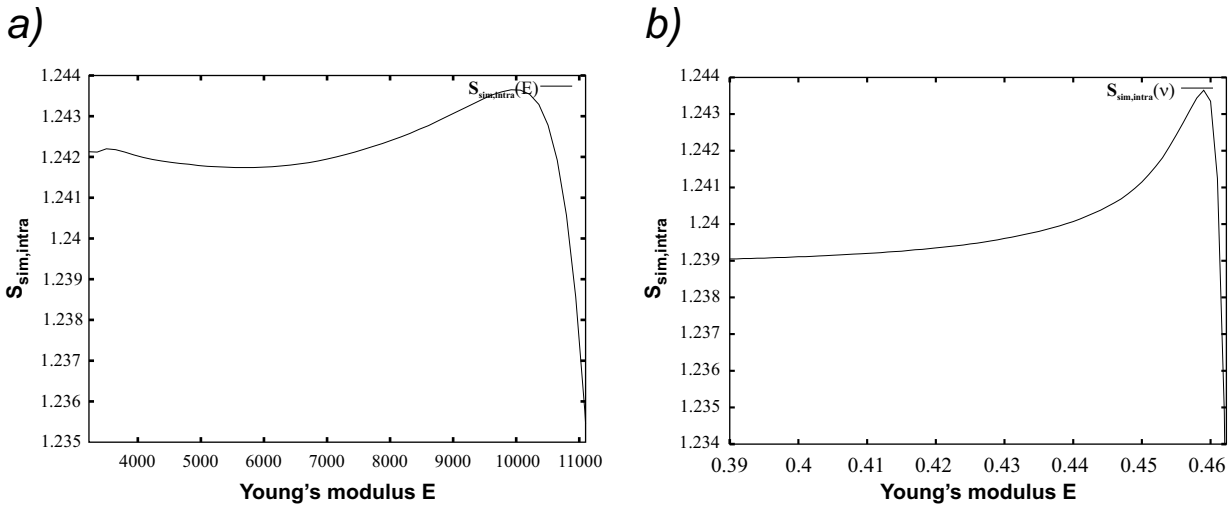

Figure 7.1: Plots of the similarity measure with one coordinate held constant: *a)* $\nu = 0.459$. *b)* $E = 9921$ Pa

spanned over the computational domain.

7.4 Discussion

For all experiments, the optimal elastic moduli E and ν maximizing the previously defined similarity measure $S_{sim,intra}$ were successfully estimated. In the experiments, the computations exhibited a stable numerical behavior, which could be achieved, among others, by the high quality tetrahedral meshes generated. Only in close vicinity to $\nu = 0.5$ for Poisson's ratio were some problems with the convergence of the system observed. This led to local fluctuations in the plot of the similarity function. Therefore, the search space for Poisson's ratio was limited to the interval $< 0.3, 0.49 >$.

Throughout the experiments, different mesh resolutions were considered in order to evaluate their influence on the behavior of the model and on the optimal values computed. Overall, the simulation showed good convergence properties, independent of the grid size. Even for the coarsest grid consisting of 9952 tetrahedra no significant instabilities occurred. Furthermore, no specific relation between the computed optimal elastic parameters and the number of grid elements was observed. This is encouraging since it suggests that the calculated optimal values can be applied to arbitrary model calculations.

The root-mean-square value computed for Young's modulus was found to be 9210.87 Pa. In absolute terms, the statistical analysis revealed some deviation between the optimal Young's moduli calculated in the individual experiments and the RMS value. However, the percentile relation between the standard deviation

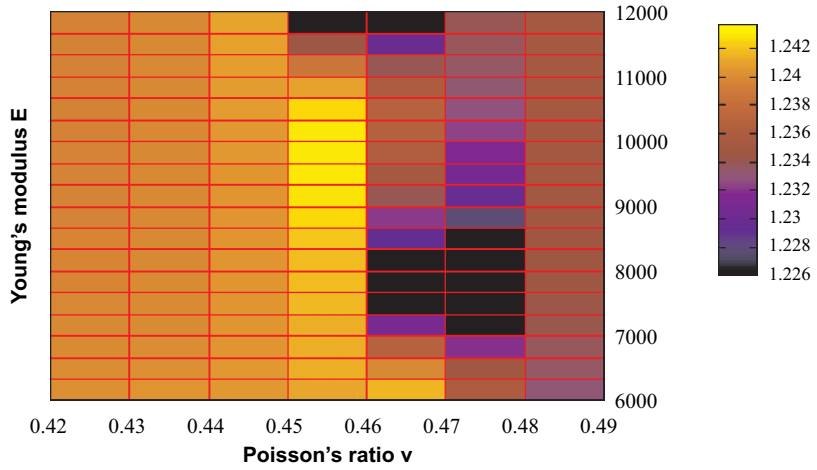

Figure 7.2: Color-coded visualization of the similarity measure sampled over a two-dimensional domain corresponding to Young's modulus E and Poisson's ratio ν

and the achieved values of E was within the interval $< 7.4\ \%, 9.3\ \% >$, which is an acceptable result and indicates only some statistical discrepancy in the calculated elastic values.

Furthermore, the RMS value of 9210.87 Pa for Young's modulus is about 4.5 times bigger than the value used by Miga *et al.* in their experiments [107]. On the other hand, this RMS value is relatively close to the value of 7425 Pa reported in an *in vivo* experiment for the swine brain [114].

Considering Poisson's ratio, the optimal values found lie only within the interval $< 0.452, 0.461 >$. The standard deviation for these numbers was calculated to be 0.003445, which confirms a very good correlation for this elastic parameter in terms of statistical analysis. Since the values of Poisson's ratio are generally very small, the bias-corrected variance was only at the level of $1.2 \cdot 10^{-6}$. The corresponding RMS value was computed to be $\nu = 0.458344$. This result is very close to the value $v = 0.45$ used in the aforementioned work [107] and in the work by Ferrant *et al.* [57].

In addition to the statistical analysis, the quality of the algorithm and the provided optimal values were inspected retrospectively for each optimization experiment, whereby the existence and the uniqueness of the optima were investigated. Let us consider Figure 7.1 which shows the projections of the similarity measure for Patient 3. The maxima in the function plots correlate to the values listed in Table 7.1. For the same experiment, the additional visualization of the function $S_{sim,intra}$ sampled over a two-dimensional domain of both elastic parameters confirmed that the localized values are at the global optima (see Figure 7.2).

Furthermore, a visual inspection of the brain deformation resulting from simulations with the optimal values of Young's modulus E and Poisson's ratio ν was performed for all cases. This visualization was accomplished with customized soft-

ware based on the Open Inventor (TGS) and AMIRA (Mercury Computer Systems) visualization packages. The comparative assessment of the simulated and the intraoperative MR images showed an improved correspondence between the model and the real situation, in comparison to the results for non-optimized values. A visual examples confirming the quality of the simulation results have already been presented in Figures 6.13 and 6.14 in Chapter 6.

7.5 Conclusion

In this chapter, an algorithm using the techniques of simulation and registration for the determination of the mechanical properties of the the brain tissue was presented. The value of this technique was shown in a series of computational experiments and their subsequent statistical analysis. The elastic parameters obtained with this method improve the correspondence between a physically-based simulation model and the real intraoperative data. The numerical results achieved correspond well with the values reported in the literature [114, 107]. This proves the strength of the proposed strategy and shows that it is a promising alternative to approaches where the mechanical properties of tissue are directly measured.

Although the algorithm was validated with a poroelastic tissue deformation model, the main concept of this technique is general, and the statistical results can be applied to perform an arbitrary biomechanical brain shift simulation, possibly based on a different physical description. This work contributes to the difficult problem of defining appropriate elasticity parameters of living soft tissue.

Chapter 8

High Performance Simulation

In the last chapters, a physically-based for modeling the brain deformation during open skull surgery has been introduced. Application of this model enables the prediction of the intraoperative movement of the soft brain tissue and thus an improved planning of the surgical intervention. Furthermore, during surgery model calculations accounting for the current conditions in the operating room can be employed to improve intraoperative navigation by incorporating updated 3D images into the respective visualization subsystems.

In neurosurgical applications an important requirement on any computer-based system is accuracy. However, performing a precise model-based simulation which includes real physical factors is a complex and computationally expensive task. Standard implementations of algorithms simulating brain shift are characterized by high execution times. In order to integrate such a simulation system into the operating room, the calculation of the brain deformation must be performed efficiently. Only then can the analysis system respond to the surgeon at a rate reasonable for use during a neurosurgical intervention.

To achieve clinically acceptable execution times, within this thesis a parallel simulation approach has been developed. This high performance implementation allows a significant reduction in computation time in comparison to a standard single processor implementation. At the same time, it preserves accuracy, which is an important issue for neurosurgery. Moreover, due to the distributed processing of the data, very high resolution geometry that does not fit into the address space of a single workstation can be processed. Thus, more complex and more detailed brain structures can be modeled, which makes the presented parallel approach even more superior to a one-processor implementation.

This chapter begins with Section 8.1 giving an overview of related work concerning techniques for simulation acceleration. Both standard and parallel techniques are addressed. Section 8.2 reveals the details of our own implementation. In Section 8.3, performance measurements of the introduced parallel framework are presented. Discussion of the achieved improvement in computation times with respect to the underlying cluster architecture is included in Section 8.4. This chapter is

concluded in Section 8.5.

8.1 Previous Work

Clinical applications of finite element simulations with an emphasis on brain shift simulation have been comprehensively reviewed in Chapter 5. However, even the best simulation algorithms are useless for real neurosurgical application if they are computationally too expensive. In this section, an overview of the existing techniques employed for the acceleration of FEM biomedical calculations is presented.

In [22], the algebraic system of equations resulting from FEM discretization is reduced through a condensation to a system where only the surface nodes are the unknowns. As a disadvantage this technique has a static character, which means that topological changes, such as cutting, cannot be modeled. The same authors introduced an extension of this method to the modeling of dynamic systems [23]. However, similar to the previous technique, depending on the mesh size, a precomputation of up to several hours is required in order to perform a simulation.

A similar technique using a banded matrix was applied by Berkley *et al.* [12] to simulate suturing wounds for models with up to 14000 nodes. The authors could improve performance in comparison to the previous implementations, but still an expensive preprocessing step was required. A hybrid approach combining FEM and a mass-spring model demonstrated by Cotin *et al.* [42] does not have this limitation. In this model deformation at interactive rates is possible but only for low-resolution meshes, which excludes it from employment for complex brain geometries. A boundary element method (BEM) was applied by Monserrat *et al.* [117] to model the liver as an elastic object. Instead of discretizing the entire volumetric domain, an appropriate geometric representation has to be constructed for the liver surface. Precalculation times are, however, still a limiting issue for this method.

A design for a parallel computation framework for a real-time laparoscopic surgery simulator was presented in [174]. However, this work concentrated mainly on the design of a new parallel architecture rather than on the acceleration of the software implementation using existing standard hardware resources. In [25], a simulation system for abdominal surgery was introduced. In this framework, software processes responsible for haptics and collision detection were separated from the rest of the simulation tasks. The communication between the processes was established via the Parallel Virtual Machine (PVM) protocol running on a shared memory architecture with four processors. Although a parallel execution of different components was accomplished with this system, this method does not account for the imbalance in load between the processors and does not allow parallel distribution and processing of grid elements.

An object-oriented parallel framework for the FEM simulation of the human knee joint motion was described by Wawro [191]. The authors minimize the data volume exchanged during the synchronization phases by applying a multilevel al-

gorithm. A minor point here is that hexahedral elements were used as the geometric primitives for FEM. Due to this choice, a large amount of manual work is required to correct of the generated meshes. A parallel implementation accelerating brain shift simulation was presented in [189]. This approach, however, suffers from an imbalance in the matrix assembly between the processors, which prevents efficient use of the available parallel resources.

In this chapter own parallel framework is proposed that is based on a standard PC cluster architecture. It is a very efficient implementation where a balanced distribution of the finite element computations over the processors is accomplished.

8.2 Parallel Framework

The FEM formulation is very well suited to parallelization since the assembly process is only locally dependent and can be carried out separately for each element. We make use of this property and introduce a strategy for conducting a finite element simulation of brain shift in parallel. In this context, our computational scheme comprises four main units discussed in more detail in the following sections:

- grid partitioning,
- distribution of the elements to the processors,
- matrix and vector assembly,
- linear system solving.

8.2.1 Grid Partitioning

A good partitioning of the computational domain prior to the actual calculations is critical to the efficiency of large-scale numerical simulations performed in parallel. There are two criteria which should be satisfied. First, it is important that the number of elements assigned to each processor is approximately equal. The goal of this requirement is to optimize the computational balance between the available CPU resources. Second, the number of elements adjacent in the mesh but assigned to different processors should be as small as possible. This condition minimizes the costs of network communication between the individual computing nodes.

A finite element mesh can be represented as a graph with the vertices corresponding to the mesh nodes. Thus, fulfilling the both enumerated conditions is tantamount to the problem of optimal graph partitioning. Traditional algorithms establish such a decomposition based on the original graph. However, the quality of the resulting partitioning is usually not satisfactory. A better alternative would be a multilevel approach, where the partitioning is conducted on the graph of reduced size and then successively adapted to the original resolution.

A very efficient multilevel k-way partitioning algorithm is provided in the METIS library [84]. This software package is embedded in our framework since it produces high quality partitionings. As the assembly process in our implementation proceeds on the element by element basis, the dual graph of the finite element mesh must be considered for the partitioning. In this representation, each vertex of the graph corresponds to one element in the mesh and the edges indicate the adjacency of the respective elements. This dual graph is automatically generated from the finite element mesh by METIS software before the actual partitioning.

There also exists a parallel version of this partitioning software called ParMETIS [85] which enables execution of the algorithm in parallel. However, this alternative is more difficult to handle and increases the management overhead. Since the complete serial partitioning procedure takes only some seconds, performing this step in parallel would not much improve the overall performance. Therefore, in our system the partitioning of the FEM mesh is conducted in a serial preprocessing step before parallel computations are started.

In Listing 8.1, a sample output of a partitioning session is presented. Specifically, the number of edges being cut by the partitioning and the balance[1] of the partitioning with respect to the number of the elements are shown. It can be seen that the achieved partitioning balance is nearly optimal. As a result of the partitioning, a unique number corresponding to one of the available CPUs is assigned to each mesh element. These numbers are then used in the next step for distributing the elements to the individual processors.

```
soza@opus2:~/metis # ../metis-4.0/partdmesh brain.mesh 8
**********************************************************************
  METIS 4.0.1 Copyright 1998, Regents of the University of Minnesota

Mesh Information ----------------------------------------------------
  Name: brain.mesh, #Elements: 545235, #Nodes: 147158, Etype: TET

Partitioning Dual Graph... ------------------------------------------
  8-way Edge-Cut:    7584, Balance:  1.03

Timing Information --------------------------------------------------
  I/O:                          0.990
  Partitioning:                 2.870
**********************************************************************
```

Listing 8.1: Sample output of the mesh partitioning program for mesh `brain.mesh` and an 8-way partitioning

[1]The balance of a k-way partitioning is defined as $\frac{km}{n}$, where m is the size of the largest partitioning part and n is the number of the elements in the mesh. For an ideally balanced partitioning this ratio would be equal to 1.0.

8.2.2 Element Distribution and Management

According to the partitioning scheme, the data is distributed to the respective destination processors. Contrary to the partitioning itself, this step is already performed in parallel. On each processor, a read function parses the input mesh file and reads in only the elements with a number from the set assigned to this processor by the partitioning procedure. These elements and the corresponding vertices are stored locally on the respective processors. The vertices located at the boundary of the neighboring partitioning regions are processed separately and stored on two or more processors. These special nodes are called *ghost vertices*.

In a further step, the global stiffness matrix and the global load vector are created which are collective and thus distributed over all processors. For this purpose, the Portable, Extensible Toolkit for Scientific Computation (PETSc) library is employed [7]. Since stiffness matrices for a finite element simulation are generally characterized by a very sparse structure, in our implementation the matrix is stored in the compressed row format to optimally use the available memory resources. Then, local parts of the global matrix and the boundary conditions expressed in the global vector are assembled element by element by each processor based on the previously stored mesh parts.

After the mesh distribution the elements and vertices are stored in local data structures and are therefore addressed in a local ordering scheme. Thus, a local-to-global mapping has to be provided in order to efficiently manage the global calculations. Specifically, accounting for the ghost vertices in the defined mapping implies that the constructed mapping is a many-to-one function (see Figure 8.1). Such mapping is essential for the assembly process since it enables gathering, summing and finally synchronizing the local information from all processors into one global structure.

Distribution of the elements in parallel allows a radical reduction of the computing and the communication time in comparison to the common strategy where only one process is responsible for reading and delivering the data to all processors. Moreover, this technique allows processing very high resolution meshes that would not fit into the address space of a single processor workstation.

8.2.3 Matrix Storage

The distribution of the stiffness matrix over given processors is performed in such a way that complete rows are assigned to the respective processors. The assignment of the rows is thus performed sequentially, e.g. in Figure 8.2 the first three rows are assigned to the first processor, the next three rows are stored on the second processor and the last three rows on the third processor. Distribution of the load and the solution vectors is conducted analogously.

This matrix distribution pattern is managed by PETSc and usually differs from the distribution scheme for the elements and the vertices resulting from the mesh

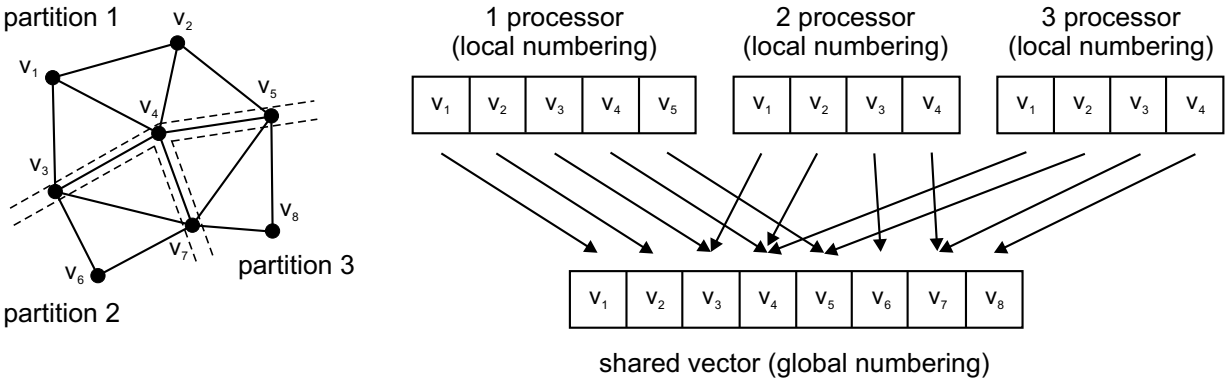

Figure 8.1: An exemplary triangle mesh partitioned over three processors and the corresponding local-to-global mapping. On each processor, the vertices are accessed in a local ordering. The vertices with global numbers v_3, v_4, v_5 and v_7 are the ghost vertices shared by the respective processors

partitioning. Since no influence on the assignment of the rows to the processors is possible, a correspondence between the matrix and the node numbering must be established in order to perform an element-based matrix assembly. For this purpose, a global-to-global mapping is introduced which maps the global numbers of the vertices onto the global numbers of the rows in the matrix.

A crucial issue for achieving a good computational performance is the preallocation of the memory required for the matrix storage during instantiation of the matrix object. Good initial estimation of the pattern of non-zero matrix entries created during the assembly process reduces the number of later memory allocations which are usually very time expensive operations. To achieve this, in a preprocessing step each processor iterates over its local elements. By applying the already mentioned global-to-global mapping, for each vertex the corresponding entries that

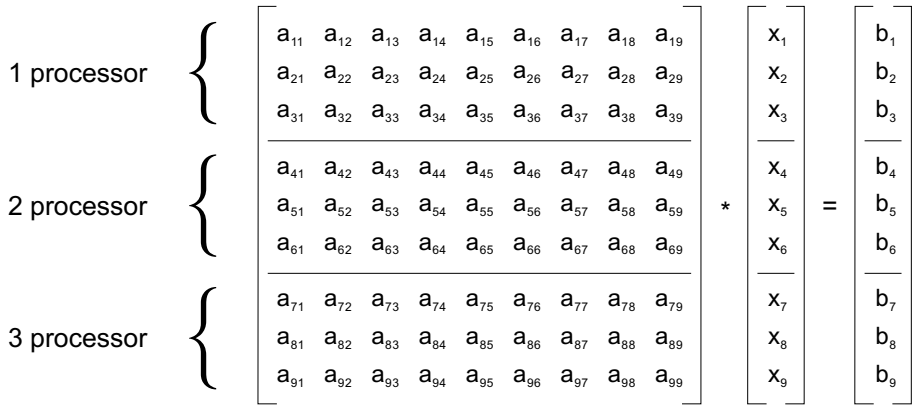

Figure 8.2: Parallel matrix and vector distribution for a linear system of equations

will be produced during the assembly process in the stiffness matrix are counted. Subsequently, the numbers of these entries from all processors are summed. In this process, the entries coming from neighboring vertices lying on the boundary between partitioning regions are counted on each of the processors storing a given vertex. This results in some overestimation of the non-zero entries but is good enough for preallocation purposes.

8.2.4 Solving the Linear Equation System

The matrix and vector assembly results in a linear equation system. In this context, PETSc library provides access to a variety of linear system solvers, including parallel and sequential, direct and iterative methods. In our framework, a parallel implementation of the Generalized Minimum Residual Method (GMRES) is employed to solve the generated linear equation system. For preconditioning, the block Jacobi method is applied. Here, each block is assigned to one process and is preconditioned with the incomplete LU (ILU) factorization.

Similar to the serial implementation, the parallel GMRES method requires the specification of convergence criteria. The convergence is based on the relative decrease $rtol$, the absolute size $atol$ and the relative increase $dtol$ of the residual norm. In iteration k, convergence is achieved if:

$$\|r_k\|_2 < \max\{atol, rtol * \|r_0\|_2\} \,, \tag{8.1}$$

with $r_k = B - Ax_k$ being the residual at iteration k. Divergence is detected if:

$$\|r_k\|_2 > dtol * \|r_0\|_2 \,. \tag{8.2}$$

Further, the GMRES algorithm requires specyfication of the maximal number of outer iterations and the value of the *restart* parameter limiting the number of inner iterations.

8.3 Results

The performance of the described framework was evaluated in a series of biomechanical simulations of brain shift performed in parallel. For the simulation runs, three pairs of pre- and intraoperative T1-weighted scans from patients undergoing craniotomy consisting of 512 $\times$ 512 $\times$ 160 voxels of the size 0.49 mm $\times$ 0.49 mm $\times$ 1.0 mm were considered. In a preliminary step, tetrahedral meshes of the brain with a resolution varying from 123496 to 1691150 elements were generated from the preoperative images previously rigidly registered to the corresponding intraoperative data. Further, the grids were partitioned (for an example see Figure 8.3) and the respective input files were generated.

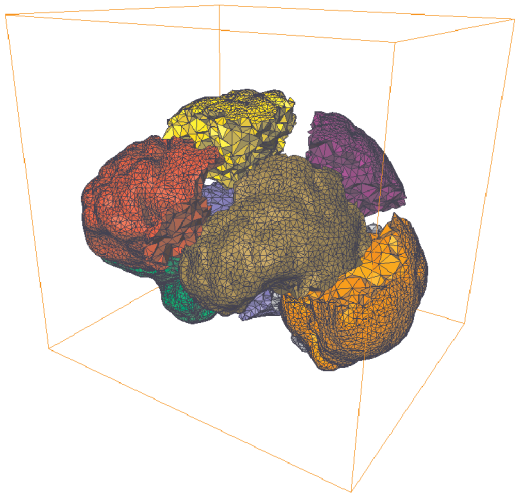

Figure 8.3: Brain geometry partitioned over eight processors

Then, simulation runs with the presented implementation were conducted and the computation times were recorded. The parameters *rtol*, *atol* and *dtol* for the GMRES algorithm were set throughout the experiments to 10^{-5}, 10^{-10} and 10^{-5}, respectively. The maximum number of iterations was limited to 5000 and the $restart$ parameter was set to 330. The measurements of the computation times from the experiments were conducted on a IA32 cluster. The parallel computer architecture consisted of 4 blade servers, each equipped with two processors and networked with 1000 Mbps Gigabit Ethernet (see Table 8.1).

Table 8.1: Architecture of the computational cluster

Component	Description
4 nodes, each with:	
CPU	2 $\times$ 2.66 GHz Intel Xeon, 533 MHz, 512 KB Cache
Memory	3 GB ECC DDR-RAM
Network card	onboard dual port 10/100/1000 BASE-T Gigabit Ethernet
Hard disk	hotplug ST336607LC 36.7 GB Ultra320 SCSI
Operating system	Linux Suse 8.2
Network switch	3Com Layer 2-Switch Baseline 2816 16 $\times$ Gigabit Ethernet RJ45 10/100/1000

There are many factors that influence the measured computation time of an application executed in parallel, e.g. the latency of the communications between the CPUs during the synchronization stages or simply the current cluster work-

load. Therefore, in order to obtain reliable measurements of the computation times, for each mesh 5 simulation runs under identical system settings were performed on one, two, four and eight processors. Altogether, 20 simulations were conducted for each pair of datasets giving a total of 60 runs, which provided sufficient data for a significant statistical evaluation.

The performance of the system was assessed relative to the times required for the main simulation tasks: data reading and distribution, preprocessing (e.g. memory allocation), assembly and solving. The additionally performed volume reconstruction and the visualization were conducted afterwards in serial and thus not considered in the performance analysis. In Figure 8.4, the mean values of the measured computation times with respect to the number of the active processors are presented separately for each mesh. In the case of the highest mesh resolution, the experiments could only be performed on four and eight processors due to the large amount of memory for storage of the mesh and the additional data structures.

In addition to these function plots, the acceleration in computation times was measured for each case as the ratio of execution time on one CPU and on n CPUs. These values are presented in Table 8.2.

8.4 Discussion

In the conducted experiments, our implementation showed a good scaling behavior with respect to the number of nodes used for the computation. In comparison to a sequential version of the simulation algorithm, the times required for the matrix assembly and solving of the linear system of equations were drastically reduced. Specifically, for a tetrahedral grid of moderate (in reference to the underlying complex brain deformation) resolution, one time step for the corresponding sparse linear system consisting of 107820 equations was performed within 11-12 seconds on eight CPUs (see Figure 8.4a). The same simulation conducted on a single-processor

Table 8.2: Acceleration in computation times for the parallel framework. The presented numbers express the improvement of the computation performance in reference to the total execution times on one CPU (Mesh 1 and Mesh 2) or four CPUs (Mesh 3)

Number of processors	Mesh 1 (123496 elements)	Mesh 2 (545235 elements)	Mesh 3 (1691150 elements)
1	1.000	1.000	–
2	1.185	1.396	–
4	2.351	2.547	1.000
8	4.160	4.660	1.944

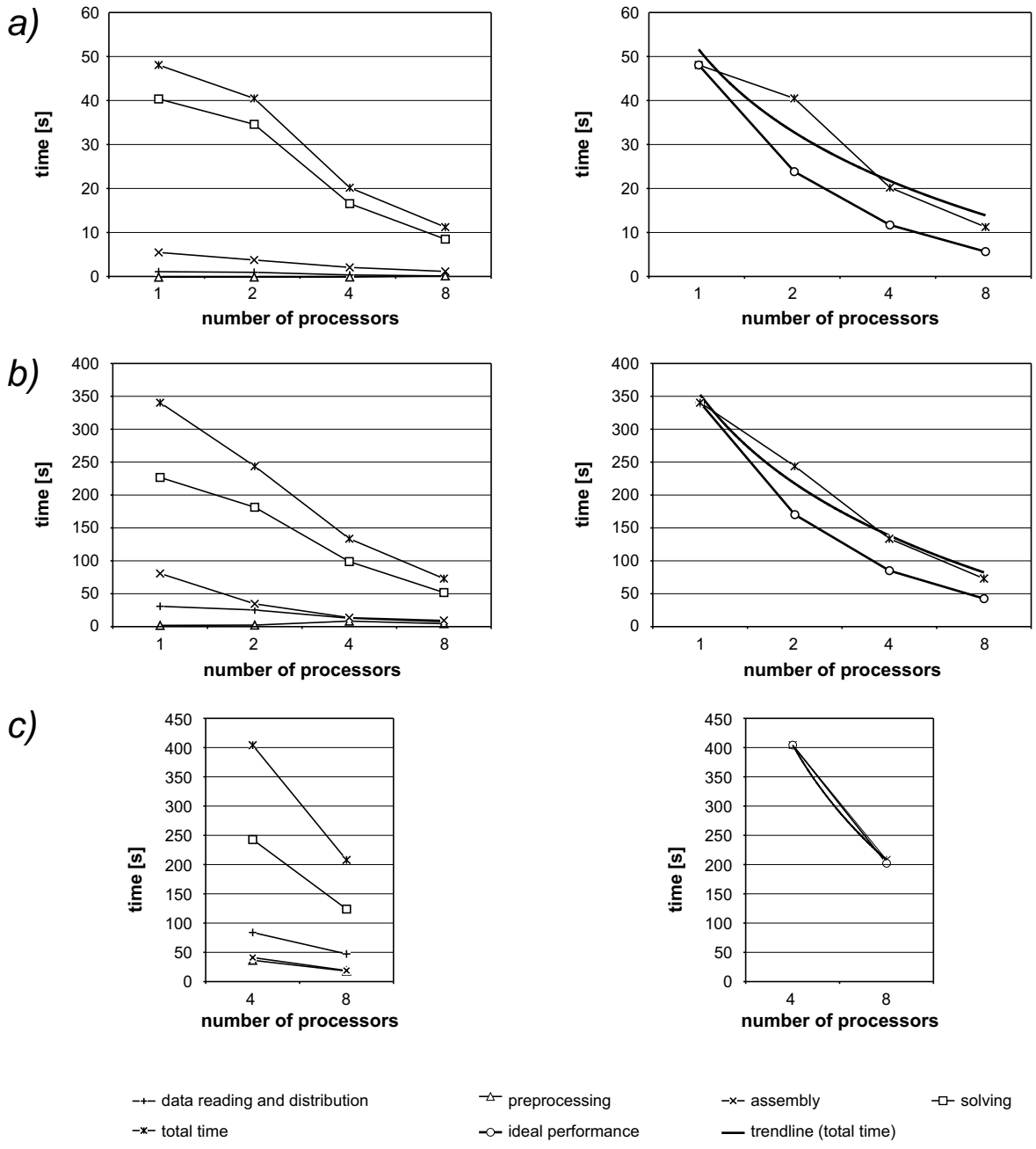

Figure 8.4: Timing results of the parallel implementation (the x-axis is logarithmically scaled). Data reading and distribution, preprocessing, assembly and solving times are plotted for a system of *a)* 107820, *b)* 588632 and *c)* 1357332 equations corresponding to the respective mesh resolutions from Table 8.2. Additionally, an ideal peak performance and a trendline for the measured total execution times are presented for each simulation series

workstation required about 50 seconds.

The parallel strategy was also the only approach which allowed conducting a simulation on a very high resolution mesh consisting of 1691150 tetrahedra and 339333 vertices. This resulted in a linear system consisting of 1357332 equations. A huge amount of data was required for managing such meshes and the related data structures did not fit into the address space of a single-processor workstation. Using the parallel framework, the simulation was successfully accomplished within about 3.5 minutes on eight processors (see Figure 8.4b).

It is a well known fact that a parallel implementation of an algorithm does not scale linearly with the time. This is explained with additional time costs related to the design of the parallel program and program execution. Specifically, creating and accessing auxiliary data structures such as the mappings requires an additional time expense. Furthermore, as introduced in the LogP model [45], the overall parallel performance depends on the architecture specific values of latency[2], bandwidth[3], number of processors and processor communication overhead[4].

In a parallel architecture different components run at different speeds. Even the assembly process performed for the same number of elements takes different amounts of time on different processors because no two tasks are exactly the same. This necessitates a synchronization between processors, e.g. in order to allow all processors to finish the assembly step and to transfer the required data before further calculations are started. For example, the entries in the global matrix related to the ghost vertices are created on different processors and therefore have to be added prior to solving the linear system. This synchronization step has a significant impact on the performance.

All these limiting factors explain why the ratios presented in Table 8.2 do not increase by a factor of 2 with the doubling of the number of processors used. Overall, the acceleration ratio of about 4.5 can be considered as a good result which can be further scaled on an architecture with a greater number of processors. The influence of the parallel overhead is most visible when comparing the running times measured for one and two processors. Whilst in this case a significant decrease in performance scalability was observed, comparing only simulation runs on two or a higher number of processors did not exhibit such behavior.

Furthermore, a better performance was observed for bigger meshes where the benefit coming from distributed computations was more significant than the communication costs. For Mesh 3, even an almost ideal factor of 1.944 could be achieved for the performance of the algorithm on four and on eight processors. Contrarily, simulations performed with smaller meshes suffered from relatively high parallel overhead (for Mesh 1 only a factor of 1.185 could be reached for simulations performed on two processors).

[2]Delay incurred on communicating a message from its source to the target processor.

[3]The reciprocal of the minimum time interval between consecutive message transmissions or receptions at a processor.

[4]The length of time that a processor is engaged in the reception or transmission of a message.

The entire time required to perform a simulation also comprises brain segmentation and mesh generation timings. These tasks can be, however, conducted in a preprocessing step and are therefore not a limitation for a practical application during surgery. Furthermore, after parallel simulation and computation of the resulting deformation field, the corresponding image data has to be recalculated accordingly. Currently, this step is performed in a sequential approach and only takes a few seconds.

8.5 Conclusion

In this chapter a parallel framework for biomechanical simulation of intraoperative brain deformation has been presented. The approach was evaluated in a series of computational experiments with clinical data to show its value. Efficient use of the available parallel architecture allowed a significant reduction in computation times in comparison to a one-processor implementation. With this strategy, simulation can be performed in a time acceptable for a clinical application. Furthermore, due to a distributed storage of the data a high level of anatomic detail can be modeled in the brain can be modeled, which is essential for neurosurgery. These facts make the presented parallel approach superior to the standard single-processor implementation.

Moreover, the described system architecture could be further exploited to achieve even better performance. In the cluster of workstations, information was exchanged using Message Passing Interface (MPI) [51] protocol. In this protocol, a processor transmitting data sends a message to a peer, which issues a receive signal before it uses the received data. In the parallel architecture used throughout the experiments, each workstation contained two processors. Therefore, the data exchange process was no longer symmetric. Depending on whether the communicating processors were located within the same workstation or not, the shared memory or the ethernet network was used to transmit data, respectively. This fact could be used to improve the performance of the parallel simulation. However, it would restrict the flexibility and the portability of the programm. Thus, in the presented implementation this very specific architecture detail is hidden by the message passing layer.

Part IV

Application of the Non-linear Registration

Chapter 9

Non-linear Correction of DTI Data

In Chapter 4, two powerful methods for the non-linear registration of 3D medical images have been introduced. Although the presented techniques have been applied to the analysis of intraoperative brain deformation, they are general and can be used for registration of arbitrary, also multimodal, three-dimensional voxel data. This chapter presents a specific application of the registration algorithm described in Section 4.3. Here, non-linear registration is employed for the correction of distortions occurring in the MR diffusion tensor imaging (DTI) data. This relatively new type of imaging provides extremely important information about neurological connections within the human brain, reflecting their organization in bundles of fibers [16]. Therefore, DTI has rapidly become an essential source of data for neurosurgery in recent years.

This chapter begins with an introduction, Section 9.1, giving the necessary medical background and providing motivation for the presented works. Furthermore, the respective steps of the algorithm for the correction of artifacts occurring in the DTI sequence are introduced in Section 9.2. Specifically, Section 9.2.1 describes the technical details related to the acquisition of the DTI data. The steps required for the extraction of the neural pathways are addressed in Section 9.2.2 and the employment of the non-linear registration in removing distortions from the preprocessed data is the subject of Section 9.2.3. Results from the experiments conducted with clinical data from the human brain are presented in Section 9.3 and subsequently discussed in Section 9.4. Finally, this chapter is concluded in Section 9.5.

9.1 Introduction

Neural pathways are anatomical connections within the human brain which are responsible for transmitting sensory, motor and a number of other neurological signals. During neurosurgery, it is essential to protect these structures from injury in order to avoid undesirable posturgical deficits. Anatomically, neural fibers consist of cylindrical cells filled with fluid bounded by a cell membrane. The diffusion of

water (commonly known as Brownian motion) within such cells has an anisotropic character since the less permeable longitudinal cell walls restrict the flow of fluid in this direction. Only very recently it has become possible to use imaging techniques to determine the location and the architecture of the anisotropic neural pathways. For this purpose, MR diffusion tensor imaging (DTI) has been employed [177, 15].

The principle of this imaging sequence is based on the acquisition of a series of diffusion images reflecting the molecular Brownian motion within tissue cells along a specified direction. These directional measurements form a basis for the computation of diffusion tensors. In order to extract meaningful information from this data, the dominant eigenvector is calculated for each tensor and a vector field containing this information is created. Subsequently, a tracking procedure is applied to the computed vector field, which results in a curvilinear path between automatically or manually specified source and target points. Such paths reveal the location and the course of the fiber tract systems within the brain.

Visualization of the so computed fiber tracts is of essential importance for neurosurgery since the fibers are visually almost indistinguishable from the surrounding tissue. Moreover, a general knowledge about the location of these tracts is not sufficient as the course of the tracts differs strongly between individuals and can be additionally influenced by a neighboring tumor [123].

However, the anatomical structures cannot be detailedly represented in the DTI data due to the coarse resolution of the underlying images. In contrary, the standard MR sequences, such as the T1-weighted sequence are able to capture the brain anatomy in a high detail. Although this data may contain some distortions, these are negligible and the image can be taken as an anatomical reference during a neurosurgical intervention. Combining information from both image modalities by integration of the computed fiber tracts into an MR T1-weighted dataset would allow visualization of the neural pathways with respect to the neighboring diseased anatomical structures to be resected (e.g. those of a tumor). This strategy can be used together with complementary techniques allowing the mapping of cortical functions in areas adjacent to tumors. Thus, a better surgery planning would be possible which would significantly minimize the probability of permanent postoperative neurological deficits.

The diffusion tensor data, however, suffers from distortions which are inherent to this imaging technique. These distortions occur above all at the skull base and in the vicinity of the transition between air and other materials, e.g. at the brain stem or at the frontal brain lobe (as shown in Figure 9.1). These artifacts can be explained by the small bandwidth of the echo planar imaging sequence in the direction of the phase encoding and by the inhomogeneity of the magnetic field related to a varying susceptibility of the tissue [9].

Several strategies have been followed to address this issue. Modified MR gradient sequences were applied in [1]. Other postprocessing methods include correcting the field maps using a computational model [81]. Also, non-linear registration of DTI and MR data was proposed to cope with these artifacts [82]. We propose to

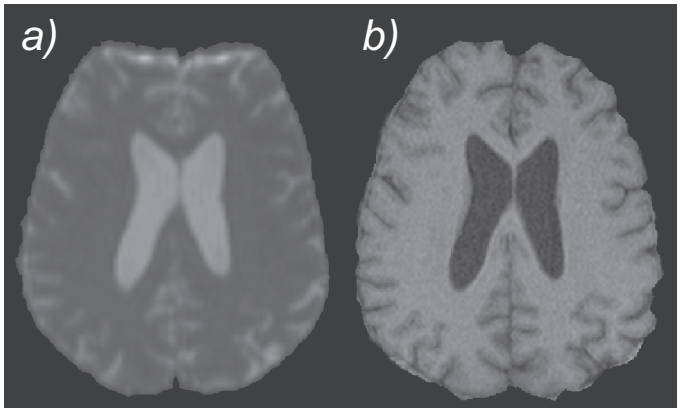

Figure 9.1: Artifacts typical of DTI data. *a)* Distorted DTI data after brain segmentation, a sagittal view. The artifacts occur primarily in the frontal lobe. *b)* The corresponding standard MR T1-weighted data does not exhibit such effects

correct the artifacts in the fiber tracts extracted from DTI data by the use of the hardware-accelerated approach using 3D Bézier functions introduced in Chapter 4.

9.2 Algorithm

To correct the neural tracts calculated from the distorted DTI images, the fibers are first extracted with a tracking algorithm operating on the precomputed diffusion tensor vector field. Then, all voxels in the DTI dataset containing a part of at least one fiber are marked. Finally, the diffusion tensor and the MR image are non-linearly registered. These steps are described in more detail in the following. Additionally, technical information about the image acquisition is provided.

9.2.1 Image Acquisition

All images were acquired at the Department of Neurosurgery at the University of Erlangen-Nuremberg with a Siemens Magnetom Sonata Maestro Class 1.5 Tesla scanner comprising a gradient system with a field strength of up to 40 mT/m (effective 69 mT/m) and a slew rate of up to 200 mT/m (effective 346 mT/m).

9.2.1.1 DTI Image Data

The data necessary for the calculation of the tensor vector field is a series of directional diffusion images. Typically 6, 12 or 18 directions are considered. In our experiments, the data consisted of an image measured without any gradient direction and six directional images acquired in different gradient directions. The gradient directions corresponded to the diffusion directions measured at a given

time. Altogether, this resulted in a total of seven volumes, in the following referred to as $B_0, \ldots, B_6$ where B_0 denotes the MR image with no gradient. The B_0 image reflects the anatomical structures best, however, in coarse resolution. The other images show mainly the information about the diffusion for a respective direction. All $B_i,\ i = 0, \ldots, 6$ datasets exhibit distortion artifacts.

Since the DTI sequence parameters have a significant influence on the quality of the images and on the occurring artifacts, in the following all important image acquisition parameters are detailed.

DTI imaging parameters: TR=9200 ms, TE=86 ms, $b_{high} = 100 s/mm^2$, $b_{low} = 0 s/mm^2$, voxel size 1.875 $\times$ 1.875 $\times$ 1.9 mm^3, 1502 Hz/Px bandwidth, acquisition matrix 128 $\times$ 128, 60 slices with no intersection gap, the diffusion-encoding gradient for the six diffusion weighted images was directed along the main coordinate axis in both directions. The acquisition time for 5 runs which were averaged was 5 min and 31 sec.

9.2.1.2 MR T1-weighted Image Data

The advantage of conventional MR imaging in comparison to DTI are a high signal to noise ratio, thin coherent slices and the possibility of multiplanar reconstruction. MR T1-weighted sequence was acquired with an acquisition matrix 256 $\times$ 256 interpolated 512 $\times$ 512, 160 slices and a voxel size of 0.488281 $\times$ 0.488281 $\times$ 1.0 mm^3. The average time required for the acquisition was 8 min 39 sec.

9.2.2 Fiber Tracking

The term "fibers" here refers to streamlines which are assumed to represent the most likely pathways through the tensor field. The extracted lines do not exactly represent the neural fibers but provide an approximated computational model of these anatomical structures. Prior to the fiber tracking, second order diffusion tensors have to be calculated for each voxel. This step is performed by solving the Stejskal-Tanner equation system [169]:

$$S_i = S_0 e^{-b\hat{\mathbf{g}}_i^T \mathbf{D} \hat{\mathbf{g}}_i} \ . \tag{9.1}$$

D is the diffusion tensor to be computed, S_i is the measured diffusion value at a given voxel for dataset B_i, S_0 is the value at this voxel in the dataset measured without a gradient, $\hat{\mathrm{g}}_i$ refers to the gradient direction used in dataset B_i and b is LeBihan's b-factor which depends on several of the acquisition parameters. The equation systems is solved with the method described in [195].

After calculation of the diffusion tensors, the fiber tracking algorithm is started, which is an iterative procedure. At the beginning, voxels with a high fractional anisotropy (FA) [11] value are selected as seed points. If the FA falls below a certain value, the tracking algorithm stops. A single tracking step thus consists of several

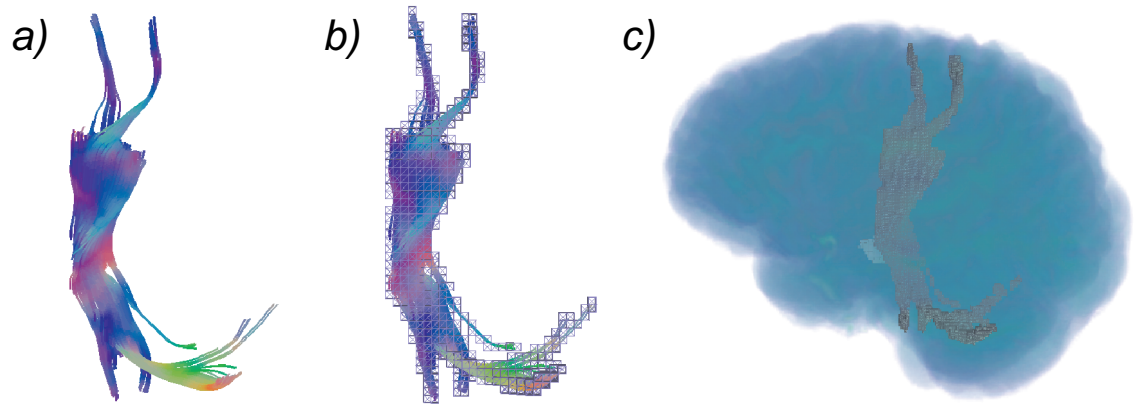

Figure 9.2: The left pyramidal tract extracted from DTI data. *a)* Fiber tracking representation. *b)* Respective voxels containing the fiber tract. *c)* 3D visualization of the marked voxels within the corresponding semi-transparently rendered B_0 volume

substeps. First, the tensor at a current point is trilinearly interpolated and the eigenvector corresponding to the highest eigenvalue is determined. This eigenvector is called the principal eigenvector and corresponds to the direction of the highest diffusion [10]. In order to track fibers with a high accuracy, the Runge-Kutta integration scheme of order four is applied which requires the computation of the principal eigenvector at four positions in the vicinity of the current position. The final direction of the streamline propagation is calculated based on these measurements.

The step size in the determined direction is fixed to a quarter of the voxel size. This strategy has proved to prevent missing sudden changes in fiber direction during tracking. Apart from FA threshold, the length of a streamline and its curvature are considered as stop criteria. Additionally, the user may choose to track only fibers running through a defined region. This enables the extraction of fiber tracts which are of special interest for a given medical application.

For further processing, the voxels containing the extracted fiber tracts are marked within the B_0 dataset by setting the respective intensity to a predefined, unambiguous value. An example of the visualization of voxels representing the left pyramidal tract is shown in Figure 9.2. The so prepared B_0 dataset is then taken for registration with the MR T1-weighted image.

9.2.3 Registration

In a first step, the B_0 image containing the marked fiber tracts is rigidly registered to the MR T1-weighted data with the use of the algorithm presented in Chapter 3. The result of this registration is taken as an initial estimation for a non-linear alignment which is performed with the use of the hardware-accelerated free-form deformation approach which has been described in Chapter 4. In this context, the B_0 image was set as the source and MR T1-weighted data was chosen as the target.

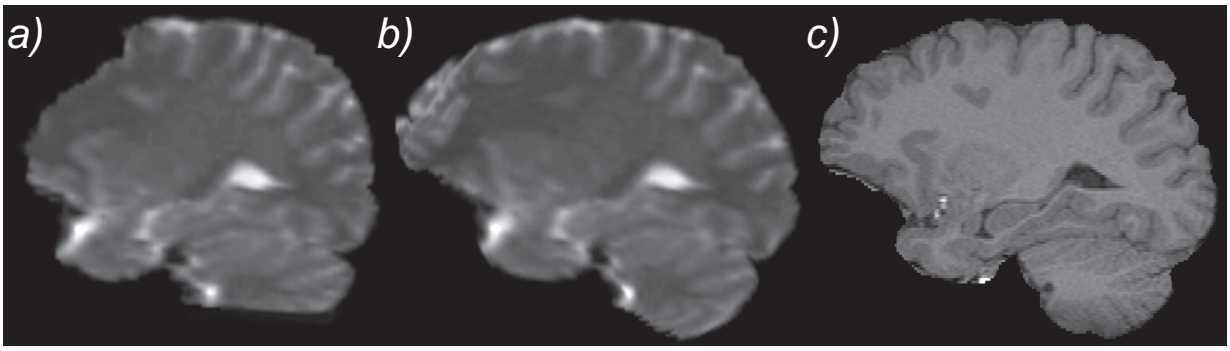

Figure 9.3: Non-linear correction of DTI images. *a)* A sagittal view of a B_0 image. *b)* The corresponding slice of the B_0 dataset after non-linear registration to a reference MR image. *c)* The corresponding sagittal view of the MR image

9.3 Results

The algorithmic steps described in the previous section were applied to correct the fiber tracts extracted from DTI images of seven patients. The mean number of voxels belonging to the left pyramidal tract marked in all B_0 datasets was 1774. Since the total number of voxels in a volume was 983040 ($128 \times 128 \times 60$), the marked part amounted to only 0.12 % of the whole volume. The total computation time was about 1.25 h for one patient. Approximately 30 minutes were required for data preprocessing and brain segmentation, whilst the rest of the time was consumed by the non-linear registration. All computations were performed on a P IV 3.0 GHz system equipped with a Nvidia GeForce FX 6800 Ultra graphics card with 256 MB memory.

Using the registration of the preprocessed B_0 image to the MR T1-weighted dataset, the distortion artifacts were non-linearly corrected and the marked left pyramidal tract was accordingly modified. In Figure 9.3, a sagittal slice from the original B_0, the corresponding slice from the non-linearly transformed B_0 and the respective slice from the target MR dataset are presented.

The corresponding axial slices of the original and the modified B_0 dataset were analyzed in order to inspect the extent of the distortion in the DTI data and to evaluate the results of the registration. For this purpose, the maximum displacement between the original and the modified left pyramidal fiber tract was measured within every second slice in the left-right and in the anterior-posterior directions. The results of these measurements are collected in Table 9.1. Additionally, a histogram showing all measured displacements for one registered pair of datasets is presented in Figure 9.4. Since the left pyramidal tract was not included in the axial slices 0-32 and 128-160, the respective entries in the histogram are set to zero.

Table 9.1: Distortions of the left pyramidal tract for seven patients. The displacements were measured for every second axial slice of a B_0 dataset with the marked left pyramidal tract before and after non-linear registration. The mean values of the displacements in the left-right and in the anterior-posterior direction are presented

Patient No	Mean distortion [mm]	
	left-right	anterior-posterior
1	0.54	1.85
2	1.16	3.31
3	0.36	5.72
4	0.94	5.35
5	0.04	1.96
6	0.61	4.70
7	0.97	2.20

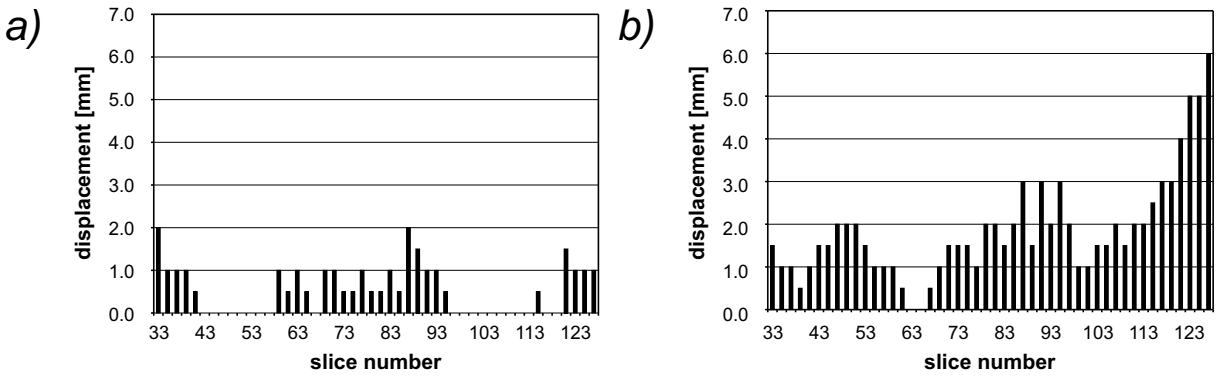

Figure 9.4: Distortion of the left pyramidal tract in the corresponding slices of a B_0 and the non-linearly corrected B_0 image measured for every second axial slice (Patient 1). *a)* The histogram of the displacements in the left-right direction. *b)* The histogram of the displacements in the anterior-posterior direction. The distortions are significantly larger in *b)* which corresponds to the phase-encoding direction

9.4 Discussion

Since the marked voxels amounted to only 0.12 % of the volume, the influence of these artificially changed intensities had a negligible impact on the registration process and thus the preprocessed B_0 image could be safely used for registration to the MR T1-weighted dataset. The validity of this assumption was proved during a comprehensive analysis of the registration results.

Overall, the benefit of the non-linear registration approach was visible. A detailed visual inspection of the registered data confirmed the good quality of the

alignment, e.g. in the central region of the gyri, the internal capsule and the brainstem. Furthermore, distortion artifacts of up to 28 mm in the frontal lobes were successfully corrected. The corresponding sagittal slices of the B_0, the non-linearly transformed B_0 and the target MR T1-weighted datasets presented in Figure 9.3 reveal the high quality of the registration.

The analysis of the fibers' displacements resulting from the non-linear correction was consistent with expectations. A small distortion was observed at the brain surface (in the medial region) and in the region of the internal capsule, whilst bigger artifacts occurred at the brainstem due to nearby air-containing structures such as the clivus[1] and the sphenoidal sinus[2]. Generally, only minor displacements were measured in the left-right direction (see Figure 9.4). For this case, the maximal mean value obtained in the experiments was 0.97 mm. The main differences were observed in the anterior-posterior direction which corresponds to the phase-encoding direction. Here, a mean displacement of 5.72 mm was determined for Patient 3.

An alternative strategy for correcting the shape of the fiber tracts would be to employ the transformation resulting from the non-linear registration of B_0 and MR T1-weighted image to the $B_1, \ldots, B_6$ datasets and to subsequently compute the fiber tracts from the resultant corrected images. However, with the approach presented here the volumetric representation of fibers is corrected which is of the main importance for the neurosurgery.

9.5 Conclusion

In this section, the non-linear algorithm using 3D Bézier functions was employed for the correction of distortions in the diffusion tensor images resulting from the underlying fast MR imaging protocol. The information contained in the DTI data is crucial since the so measured neural pathways are responsible for transmitting signals emitted during all neurological functions and have to be preserved during surgery to avoid undesirable postoperative deficits. The presented technique provides accurate and reliable data, thus enabling spatially correct preoperative neurosurgical planning. Furthermore, the corrected fiber tracts combined with information from conventional and functional MR imaging can be integrated into a neuronavigation system giving a powerful surgical support.

[1] A downward slopping surface continuous with the groove on the basilar portion of the occipital bone. It supports the upper part of the pons.

[2] Either of a pair of cavities in the body of the sphenoid bone (a compound bone of winglike shape, situated at the base of the skull) that communicate with the nasal cavity.

Part V

Conclusion

Chapter 10

Summary

In recent years, surgical procedures have been revolutionized by the invention of navigation hardware which application has led to the introduction of image-guided surgery. This new surgical strategy brings the possibility of tracking surgical tools during the intervention, which enables simultaneous navigation both in the physical space associated with the operating room and in a virtual image space. Especially in the context of neurosurgery, this has resulted in an enormous improvement in the quality and in a considerable simplification of surgical procedures. As an example, projection of tumor boundaries into the semi-transparent mirror of an operating microscope is possible with the registration transformation between the physical and image spaces. This allows conducting the resection of diseased tissue more precisely and effectively.

Since image-guided neurosurgery is strongly related to computer technology and the availability of mathematical models, there are still many needs to meet and open challenges to solve. One of them is the development of fast and yet reliable methods for the registration of medical images. In this context, rigid image registration algorithms can be employed for finding the transformation between the pre- and intraoperative MR data. This transformation can then be utilized during image-guided surgery to transfer preoperative planning into the intraoperative setting or to perform an image update within the navigation system. Furthermore, there is a need for the development of computationally efficient and at the same time accurate non-linear registration techniques for the calculation of highly complex soft brain tissue deformation which takes place during an open-skull surgery. Another challenging issue is a reliable simulation of the occurring brain shift for surgery planning and also for performing an image update in the operating setting where no intraoperative imaging is available.

In this doctoral thesis, new methods for linear and non-linear registration of medical image data and for a physically-based simulation of the intraoperative brain deformation have been developed. First, a hardware-based algorithm for three-dimensional rigid registration has been implemented. Since the intraoperative environment is characterized by extremely high requirements on the compu-

tation times, modern graphics hardware has been employed for conducting time consuming calculations. Thereby, no external features such as fiducial markers are required as this algorithm is based on the voxel intensities of medical images only. Various experiments based on MR data acquired before and during human brain surgery proved the value of the algorithm for neurosurgical procedures and showed its suitability for the intraoperative setting in terms of performance.

Further, new methods for non-linear registration have been proposed and implemented. In this context, an approach based on a volumetric hexahedral structure and an algorithm using 3D Bézier functions have been introduced and comprehensively evaluated for neurosurgical applications. In both methods, the interpolation capabilities of 3D texture mapping hardware have been exploited for the acceleration of the computations. In experiments conducted with real data, the former approach has been characterized by a little bit better computational efficiency, but the latter technique has been proven to provide qualitatively better results. Generally, both approaches were capable to cope with large non-linear deformations of the brain in most of the cases and could find satisfactory solution within a reasonable time interval.

In order to predict the intraoperative brain deformation, a simulation framework has been introduced where the occurring phenomenon has been modeled by means of physically-based equations. Based on the preoperative MR data, first an appropriate 3D geometry is generated with a sophisticated method ensuring the regularity of the constructed grid. Then, the governing equations are solved with a finite element technique. Subsequently, a three-dimensional image is reconstructed according to the resulting displacement vector field. This image can be integrated into a navigation system, thus providing the surgeon with information about the direction and the magnitude of the tissue shift. Furthermore, a high performance implementation has been introduced for accelerating the computations. Within this extension to the simulation framework, the underlying geometry is partitioned into a number of topologically optimal parts and the computations are distributed among several processors grouped in a PC cluster. This technique leads to a drastic reduction in computation times, whilst the accuracy of the solution is preserved. Moreover, the proposed system is further scalable and has a potential for many other general applications.

Finally, the methods developed for registration have been coupled with the simulation framework in order to estimate the optimal elastic properties of brain tissue for the use within a physically-based model. A statistical analysis of the results obtained in a series of experiments with pre- and intraoperative head MR images allowed the physical constants describing the brain tissue to be calculated. This enables a realistic prediction of brain shift before surgery (e.g. for planning purposes) assuming the location of the skull opening is known.

All the algorithms introduced here have been designed to support image-guided neurosurgical intervention. This represents a considerable contribution to the difficult problem of compensation for and analysis of intraoperative brain deformation.

Another application for the developed techniques is the removal of distortions from diffusion tensor image (DTI) data. Here, the subimage representing functional information has been non-linearly registered to the anatomical MR data. The resulting transformation has been then used to correct the position of the fiber tracks calculated from the directional gradient images. Although the non-linear algorithm had been initially developed for the registration of pre- and intraoperative MR images, it has been also successfully utilized in this context emphasizing the strength and the flexibility of the presented systems.

10.1 Future Challenges

The rapid progress in computer technology and the enormous growth of medical knowledge in recent years has increased demand for the introduction of new and improved applications of computer-assisted interventions and medical image computing. The activity in related research fields suggests that image-guided neurosurgery will become more and more important, implying an increasing need for better and faster techniques assisting diagnosis and therapy. Thereby, an extremely important issue is the real-time requirement imposed by the intraoperative restrictions.

The methods and their applications introduced within this work provide a flexible and powerful basis for the development of further algorithms and for their implementation in intraoperative assistance systems. Future challenges focus on further reduction of computation times, especially for non-rigid registration. Although the methods introduced in this work enable a complete three-stage non-linear registration in less than an hour, processing times under 10 minutes are necessary in order to make this technique intraoperatively applicable.

Graphics hardware shows a significant potential that can be exploited for this purpose. The new generation of programmable graphics adapters allows further acceleration of non-rigid registration calculations. The methods introduced in this work provide a comprehensive basis for the use of graphics hardware in medical image computing and show the promise of further research in this direction.

Apart from these computational issues, visualization and augmented reality strategies are gaining increasingly in importance. In presently available commercial systems, the surgeon is supported by visual incorporation of preoperative planning data in optical systems. This information, however, is not truly three-dimensional. Integration of images obtained with the developed simulation and registration algorithms with a 3D image overlay system fusing the real and virtual space would significantly improve the quality of surgical interventions.

Part VI

Appendix

Appendix A

Finite Element Discretization

A.1 Weighted Residuals Form

According to the Galerkin version of the weighted residuals technique, to obtain a numerical solution to the poroelastic equation system, Equations 6.20a and 6.20b have to be multiplied by the basis functions φ_i and subsequently integrated over the computational domain Ω (see Equation 6.6). This yields the following:

$$-(\lambda+\mu)\int_\Omega \nabla(\nabla\cdot\mathbf{u}(\mathbf{x},t))\cdot\varphi_i d\Omega - \mu\int_\Omega \Delta\mathbf{u}(\mathbf{x},t)\cdot\varphi_i d\Omega + \int_\Omega \nabla p(\mathbf{x},t)\cdot\varphi_i d\Omega = 0\ , \tag{A.1a}$$

$$\int_\Omega \frac{\partial}{\partial t}(\nabla\cdot\mathbf{u}(\mathbf{x},t))\cdot\varphi_i d\Omega - \int_\Omega (\nabla\cdot k\nabla p(\mathbf{x},t))\cdot\varphi_i d\Omega = 0\ . \tag{A.1b}$$

In order to establish the discrete form of the equations, some terms have to be rearranged. For this purpose, let us first consider a general formula for the divergence of the product of a scalar field C and a vector field $\vec{V}$:

$$\nabla\cdot(C\vec{\mathbf{V}}) = C(\nabla\cdot\vec{\mathbf{V}}) + (\nabla C)\cdot\vec{\mathbf{V}}\ . \tag{A.2}$$

Applying this identity to the Equations A.1a and A.1b results in the following relations:

1. For the first term of Equation A.1a, we have:

$$\nabla((\nabla\cdot\mathbf{u})\cdot\varphi_i) = \nabla\varphi_i\cdot(\nabla\cdot\mathbf{u}) + \nabla(\nabla\cdot\mathbf{u})\cdot\varphi_i \Rightarrow$$
$$\int_\Omega \nabla(\nabla\cdot\mathbf{u})\cdot\varphi_i d\Omega = \int_\Omega \nabla((\nabla\cdot\mathbf{u})\cdot\varphi_i)d\Omega - \int_\Omega (\nabla\cdot\mathbf{u})\cdot\nabla\varphi_i d\Omega\ . \tag{A.3}$$

   From the simplified version of Gauss's integral theorem follows the formula for the integral of the gradient of a function:

$$\int_\Omega \nabla((\nabla\cdot\mathbf{u})\cdot\varphi_i)d\Omega = \oint_{\partial\Omega} (\nabla\cdot\mathbf{u})\cdot\varphi_i d\vec{\Omega}\ . \tag{A.4}$$

Hence, the following equality holds:

$$(\lambda+\mu)\int_\Omega \nabla(\nabla\cdot\mathbf{u}(\mathbf{x},t))\cdot\varphi_i d\Omega = (\lambda+\mu)\oint_{\partial\Omega}(\nabla\cdot\mathbf{u})\cdot\varphi_i d\vec{\Omega} - (\lambda+\mu)\int_\Omega(\nabla\cdot\mathbf{u})\cdot\nabla\varphi_i d\Omega\ . \tag{A.5}$$

2. Considering the second term of Equation A.1a and the relation $\Delta\mathbf{u}\cdot\varphi_i = \varphi_i\nabla\cdot(\nabla\mathbf{u})$ results in:

$$\nabla\cdot(\varphi_i\nabla\mathbf{u}) = \varphi_i\nabla\cdot(\nabla\mathbf{u}) + (\nabla\varphi_i)\cdot(\nabla\mathbf{u}) \Rightarrow$$
$$\mu\int_\Omega\Delta\mathbf{u}\cdot\varphi_i d\Omega = \mu\int_\Omega\nabla\cdot(\varphi_i\nabla\mathbf{u})d\Omega - \mu\int_\Omega(\nabla\varphi_i)\cdot(\nabla\mathbf{u})d\Omega\ . \tag{A.6}$$

The integral theorem by Gauss relates the volume integral of the divergence to the surface integral by:

$$\int_\Omega\nabla\cdot(\varphi_i\nabla\mathbf{u})d\Omega = \oint_{\partial\Omega}\varphi_i\nabla\mathbf{u}d\vec{\Omega}\ . \tag{A.7}$$

Combining this and Equation A.6 the following equality is obtained:

$$\mu\int_\Omega\Delta\mathbf{u}(\mathbf{x},t)\cdot\varphi_i d\Omega = \mu\oint_{\partial\Omega}\varphi_i\nabla\mathbf{x},t)d\vec{\Omega} - \mu\int_\Omega\nabla\varphi_i\cdot\nabla\mathbf{u}(\mathbf{x},t)d\Omega\ . \tag{A.8}$$

3. Finally, for the last term of Equation A.1b the formula for the divergence of a product is considered:

$$\nabla\cdot(k\varphi_i\nabla p) = k\varphi_i(\nabla\cdot\nabla p) + k\nabla p\cdot\nabla\varphi_i \Rightarrow$$
$$\int_\Omega(\nabla\cdot k\nabla p)\cdot\varphi_i d\Omega = \int_\Omega\nabla\cdot(k\varphi_i\nabla p)d\Omega - \int_\Omega k\nabla\varphi_i\cdot\nabla p d\Omega\ . \tag{A.9}$$

By Gauss's theorem it holds that:

$$\int_\Omega\nabla\cdot(k\varphi_i\nabla p)d\Omega = \oint_{\partial\Omega}k\varphi_i\nabla p d\vec{Omega}\ , \tag{A.10}$$

thus

$$\int_\Omega(\nabla\cdot k\nabla p(\mathbf{x},t))\cdot\varphi_i d\Omega = \oint_{\partial\Omega}k\nabla p\cdot\varphi_i d\vec{\Omega} - \int_\Omega k\nabla p\cdot\nabla\varphi_i d\Omega\ . \tag{A.11}$$

Inserting the identities from Equations A.5, A.8 and A.11 into Equations A.1a and A.1b leads to a modified governing system:

$$(\lambda+\mu)\int_\Omega(\nabla\cdot\mathbf{u}(\mathbf{x},t))\cdot\nabla\varphi_i d\Omega + \mu\int_\Omega\nabla\varphi_i\cdot\nabla\mathbf{u}(\mathbf{x},t)d\Omega + \int_\Omega\nabla p(\mathbf{x},t)\cdot\varphi_i d\Omega =$$
$$(\lambda+\mu)\oint_{\partial\Omega}(\nabla\cdot\mathbf{u}(\mathbf{x},t))\cdot\varphi_i d\vec{\Omega} + \mu\oint_{\partial\Omega}\nabla\mathbf{u}(\mathbf{x},t)\cdot\varphi_i d\vec{\Omega}\ , \tag{A.12a}$$

$$\int_\Omega\frac{\partial}{\partial t}(\nabla\cdot\mathbf{u}(\mathbf{x},t))\cdot\varphi_i d\Omega - \int_\Omega k\nabla\varphi_i\cdot\nabla p(\mathbf{x},t)d\Omega = \oint_{\partial\Omega}k\nabla p\cdot\varphi_i d\vec{\Omega}\ . \tag{A.12b}$$

A.2 Discretization

The matrix form of the system of governing equations in the form presented in the last section can be easily established. For this purpose, the unknown displacement vector $\mathbf{u}$ and the unknown scalar value of interstitial pressure p are approximated by a finite series of the weighted basis functions:

$$\mathbf{u}(\mathbf{x},t) = \sum_{i=1}^{n} u_i(t)\varphi_i(\mathbf{x}) \ , \tag{A.13}$$

$$p(\mathbf{x},t) = \sum_{i=1}^{n} p_i(t)\varphi_i(\mathbf{x}) \ . \tag{A.14}$$

To establish the entries in the stiffness matrix and in the load vector, it is useful to consider the governing equations separately for each coordinate of the displacement vector. For the component j ($j \in \{x,y,z\}$) of the displacement vector $\mathbf{u}$, Equation A.13 turns into:

$$u_j(\mathbf{x},t) = \sum_{k=1}^{n} u_{j_k}(t)\varphi_k(\mathbf{x}) \ . \tag{A.15}$$

For the gradient of the displacement it holds:

$$\nabla u_j(\mathbf{x},t) = \sum_{k=1}^{n} u_{j_k}(t)\nabla\varphi_k(\mathbf{x}) \ . \tag{A.16}$$

The divergence of the displacement vector fulfill the identities:

$$\nabla\cdot\mathbf{u}(\mathbf{x},t) = \nabla\cdot\sum_{k}\mathbf{u_k}\varphi_k(\mathbf{x}) = \sum_{k}\nabla\cdot(\mathbf{u_k}\varphi_k(\mathbf{x})) =$$
$$\sum_{k}(u_{kx}\frac{\partial}{\partial x}\varphi_k(\mathbf{x}) + u_{ky}\frac{\partial}{\partial y}\varphi_k(\mathbf{x}) + u_{kz}\frac{\partial}{\partial z}\varphi_k(\mathbf{x})) \ . \tag{A.17}$$

This can be written in a more compact form as:

$$\nabla\cdot\mathbf{u}(\mathbf{x}) = \sum_{k}\mathbf{u_k}\cdot\nabla\varphi_k(\mathbf{x}) \ . \tag{A.18}$$

Substituting the corresponding terms in Equations A.12a and A.12b (considered for the component j of the displacement vector $\mathbf{u}$) leads to:

$$\sum_{k=1}^{n}((\lambda+\mu)u_{j_k}\cdot\int_\Omega\nabla\varphi_k\cdot\nabla\varphi_i d\Omega + \mu u_{j_k}\int_\Omega\nabla\varphi_k\cdot\nabla\varphi_i d\Omega + p_k\int_\Omega\nabla\varphi_k\cdot\varphi_i d\Omega) =$$
$$(\lambda+\mu)\oint_{\partial\Omega}(\nabla\cdot u_x(\mathbf{x},t))\cdot\varphi_i d\vec{\Omega} + \mu\oint_{\partial\Omega}\nabla u_x(\mathbf{x},t)\cdot\varphi_i d\vec{\Omega} \ , \tag{A.19a}$$

$$\sum_{k=1}^{n}(\frac{\partial u_{j_k}}{\partial t}\int_\Omega\nabla\varphi_k\cdot\varphi_i - p_k\int_\Omega k\nabla\varphi_k\cdot\nabla\varphi_i\Omega) = \oint_{\partial\Omega}k\nabla p\cdot\varphi_i d\vec{\Omega} \ . \tag{A.19b}$$

The derivative of time in the second equation is eliminated by integrating the system of equations in time using a two-point weighting technique:

$$\int_{t_n}^{t_{n+1}} \mathbf{f}(t)dt = \Delta t[\theta \mathbf{f}(t_{n+1}) + (1-\theta)\mathbf{f}(t_n)],\ \Delta t = t_{n+1} - t_n\ 0 \leq \theta \leq 1\ . \tag{A.20}$$

The governing equations can be now expressed in the matrix form as:

$$\mathbf{A}^{\theta}\mathbf{x}_{n+1} - \mathbf{A}^{1-\theta}\mathbf{x}_n = \mathbf{b}_{n+1}^{\theta} - \mathbf{b}_n^{1-\theta}\ . \tag{A.21}$$

For a fixed time point, four unknowns (u_x, u_y, u_z and p) are related to each node of the finite element grid. Thus, the matrix entries a_{ij}^{θ} are submatrices of the size 4×4 and are given by (the coefficients $a_{ij}^{1-\theta}$ have an analogous form):

$$a_{ij}^{\theta} = \begin{bmatrix} \theta(\mu m + (\lambda+\mu)m_{xx}) & \theta(\lambda m_{xy} + \mu m_{yx}) & \theta(\lambda m_{xz} + \mu m_{zx}) & \theta n_x \\ \theta(\lambda m_{yx} + \mu m_{xy}) & \theta(\mu m + (\lambda+\mu)m_{yy}) & \theta(\lambda m_{yz} + \mu m_{zy}) & \theta n_y \\ \theta(\lambda m_{zx} + \mu m_{xz}) & \theta(\lambda m_{zy} + \mu m_{yz}) & \theta(\mu m + (\lambda+\mu)m_{zz}) & \theta m_z \\ m_x & m_y & m_z & \theta \Delta t k m \end{bmatrix} \tag{A.22}$$

$$m_{kl} = \int_{\Omega} \frac{\partial \varphi_i}{\partial k}\frac{\partial \varphi_j}{\partial l}d\Omega, \quad k \in \{x,y,z\},\ l \in \{x,y,z\}\ ,$$

$$m_k = \int_{\Omega} \varphi_i \frac{\partial \varphi_j}{\partial k}d\Omega,\ k \in \{x,y,z\}\ ,$$

$$m = m_{xx} + m_{yy} + m_{zz}\ .$$

Establishing the entries in the matrix does not directly result from Equations A.19a, A.19b and A.20. It requires rearrangement of the boundary integrals, which consists of several steps. First, the following equality has to be considered:

$$\left[\mu \oint_{\partial\Omega} \nabla u_x \cdot \varphi_i d\vec{\Omega} + (\lambda+\mu) \oint_{\partial\Omega} (\nabla \cdot u_x) \cdot \varphi_i d\vec{\Omega}\right]_x = \oint_{\partial\Omega} \left[(\lambda+\mu)n_x(\frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z}\right] \varphi_i ds + \oint_{\partial\Omega} \left[(2\mu+\lambda)n_x\frac{\partial u_x}{\partial x} + \mu n_y \frac{\partial u_x}{\partial y} + \mu n_z \frac{\partial u_x}{\partial z}\right] \varphi_i ds\ . \tag{A.23}$$

Replacing the strains in Equation 6.9 by the respective terms from Equation 6.10 results in:

$$\begin{aligned} \sigma_{xx} &= 2\mu\frac{\partial u_x}{\partial x} + \lambda\left(\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z}\right)\ , \\ \sigma_{xy} &= \mu\left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}\right)\ , \\ \sigma_{xz} &= \mu\left(\frac{\partial u_x}{\partial z} + \frac{\partial u_z}{\partial x}\right)\ . \end{aligned} \tag{A.24}$$

Based on these dependencies, weighting the scalar product $\vec{n} \cdot \sigma_x$ with a basis function φ_i and integrating over the boundary $\partial\Omega$ yields:

$$\left[\oint_{\partial\Omega} \vec{n} \cdot \sigma_x \varphi_i ds\right]_x = \oint_{\partial\Omega} \left[(2\mu+\lambda)n_x \frac{\partial u_x}{\partial x} + \mu n_y \frac{\partial u_x}{\partial y} + \mu n_z \frac{\partial u_x}{\partial z}\right] \varphi_i ds + \oint_{\partial\Omega} \left[\lambda n_x \frac{\partial u_y}{\partial y} + \lambda n_x \frac{\partial u_z}{\partial z} + \mu n_y \frac{\partial u_y}{\partial x} + \mu n_z \frac{\partial u_z}{\partial x}\right] \varphi_i ds \; . \quad \text{(A.25)}$$

Furthermore, note that:

$$\oint_{\partial\Omega} \left[(\lambda+\mu)n_x \frac{\partial u_y}{\partial y} - \mu n_x \frac{\partial u_y}{\partial y} + (\lambda+\mu)n_x \frac{\partial u_z}{\partial z} - \mu n_x \frac{\partial u_z}{\partial z} + \mu n_y \frac{\partial u_y}{\partial x} + \mu n_z \frac{\partial u_z}{\partial x}\right] \varphi_i ds = \oint_{\partial\Omega} \left[\lambda n_x \frac{\partial u_y}{\partial y} + \lambda n_x \frac{\partial u_z}{\partial z} + \mu n_y \frac{\partial u_y}{\partial x} + \mu n_z \frac{\partial u_z}{\partial x}\right] \varphi_i ds \; . \quad \text{(A.26)}$$

Combining Equations A.23, A.25 and A.26 leads to:

$$\left[\mu \oint_{\partial\Omega} \nabla u_x \cdot \varphi_i d\vec{\Omega} + (\lambda+\mu) \oint_{\partial\Omega} (\nabla \cdot u_x) \cdot \varphi_i d\vec{\Omega}\right]_x = \left[\oint_{\partial\Omega} \vec{n} \cdot \sigma_x \varphi_i ds\right]_x + \oint_{\partial\Omega} \left[\mu n_x \frac{\partial u_y}{\partial y} - \mu n_y \frac{\partial u_y}{\partial x} + \mu n_x \frac{\partial u_z}{\partial z} - \mu n_z \frac{\partial u_z}{\partial x}\right] \varphi_i ds \; . \quad \text{(A.27)}$$

Additionally, it holds that:

$$\begin{aligned} \oint_{\partial\Omega} \left[\mu n_x \frac{\partial u_y}{\partial y} - \mu n_z \frac{\partial u_y}{\partial x}\right] \varphi_i ds &= \int_\Omega \mu \frac{\partial}{\partial x}\left(\frac{\partial u_y}{\partial y}\varphi_i\right) d\Omega - \int_\Omega \mu \frac{\partial}{\partial y}\left(\frac{\partial u_y}{\partial x}\varphi_i\right) d\Omega \\ &= \int_\Omega \mu \frac{\partial u_y}{\partial y}\frac{\partial \varphi_i}{\partial x} d\Omega - \int_\Omega \mu \frac{\partial u_y}{\partial x}\frac{\partial \varphi_i}{\partial y} d\Omega \; . \end{aligned} \quad \text{(A.28)}$$

It also holds that:

$$\begin{aligned} \oint_{\partial\Omega} \left[\mu n_x \frac{\partial u_z}{\partial z} - \mu n_z \frac{\partial u_z}{\partial x}\right] \varphi_i ds &= \int_\Omega \mu \frac{\partial}{\partial x}\left(\frac{\partial u_z}{\partial z}\varphi_i\right) d\Omega - \int_\Omega \mu \frac{\partial}{\partial z}\left(\frac{\partial u_z}{\partial x}\varphi_i\right) d\Omega \\ &= \int_\Omega \mu \frac{\partial u_z}{\partial z}\frac{\partial \varphi_i}{\partial x} d\Omega - \int_\Omega \mu \frac{\partial u_z}{\partial x}\frac{\partial \varphi_i}{\partial z} d\Omega \; . \end{aligned} \quad \text{(A.29)}$$

This finally leads to the following form of the boundary integrals:

$$\left[\oint_{\partial\Omega} \varphi_i \mu \nabla u_x d\vec{\Omega} + \oint_{\partial\Omega} \varphi_i (\lambda+\mu) \nabla \cdot u_x d\vec{\Omega}\right]_x = \left[\oint_{\partial\Omega} \vec{n} \cdot \sigma_x \varphi_i ds\right]_x + \int_\Omega \mu \frac{\partial u_y}{\partial y}\frac{\partial \varphi_i}{\partial x} d\Omega - \int_\Omega \mu \frac{\partial u_y}{\partial x}\frac{\partial \varphi_i}{\partial y} d\Omega + \int_\Omega \mu \frac{\partial u_z}{\partial z}\frac{\partial \varphi_i}{\partial x} d\Omega - \int_\Omega \mu \frac{\partial u_z}{\partial x}\frac{\partial \varphi_i}{\partial z} d\Omega \; . \quad \text{(A.30)}$$

Here, the volume integral terms contribute to the stiffness matrix, whereas the boundary integral is placed in the load vector. This results in the final discretization form as presented in Equations 6.22 and 6.23.

Bibliography

[1] A. L. Alexander, J. S. Tsuruda, and D. L. Parker. Elimination of Eddy Current Artifacts in Diffusion-Weighted Echo-Planar Images: The Use of Bipolar Gradients. *Magnet Reson Med*, 38(6):1016–1021, 1997. 9.1

[2] N. Alpert, J. Bradshaw, D. Kennedy, and J. Correia. The principal axis transforamtion — a method for image registration. *J Nucl Med*, 31:1717–1722, 1990. 2.3.1.3

[3] R. Alterovitz, J. Pouliot, R. Taschereau, I. C. Hsu, and K. Goldberg. Simulating Needle Insertion and Radioactive Seed Implantation for Prostate Brachytherapy. *Stud Health Technol Inform*, 94:19–25, 2003. 5.1

[4] S. L. Aquino, J. C. Asmuth, N. M.. Alpert, E. F. Halpern, and A. J. Fischman. Improved Radiologic Staging of Lung Cancer with 2-[18F]-fluoro-2-deoxy-D-glucose-Positron Emission Tomography and Computed Tomography Registration. *J Comput Assist Tomogr*, 24(7):479–484, 2003. 2.2

[5] F. S. Azar, D. N. Metaxas, and M. D. Schnall. Methods for Modeling and Predicting Mechanical Deformations of the Breast under External Perturbations. *Med Image Anal*, 6(1):1–27, 2002. 5.1

[6] R. Bajcsy and S. Kovacic. Multiresolution elastic matching. *Computer Vision, Graphics and Image Processing*, 46:1–21, 1989. 2.4.1.3

[7] S. Balay, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. PETSc Users Manual. Technical Report ANL-95/11 - Revision 2.1.5, Argonne National Laboratory, 2004. 8.2.2

[8] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition*. SIAM, 1994. 6.7.2

[9] P. J. Basser and D. K. Jones. Diffusion-Tensor MRI: Theory, Experimental Design and Data Analysis - a Technical Review. *NMR Biomed*, 15:456–467, 2002. 9.1

[10] P. J. Basser, J. Mattiello, and D. Le Bihan. MR Diffusion Tensor Spectroscopy and Imaging. *Biophys J*, 66(1):259–267, 1994. 9.2.2

[11] P.J. Basser and C. Pierpaoli. Microstructural and Physiological Features of Tissues Elucidated by Quantitative-Diffusion-Tensor MRI. *J Magn Reson Ser B*, 111(3):209–219, 1996. 9.2.2

[12] J. Berkley, S. Weghorst, H. Gladstone, G. Raugi, D. Berg, and M. Ganter. Fast Finite Element Modeling for Surgical Simulation. *Stud Health Technol Inform*, 62:55–61, 1999. 8.1

[13] P. Besl and N. McKay. A method for registration of 3D shapes. *IEEE Trans. PAMI*, 14:239–256, 1992. 2.3.1.1

[14] P. Bezier. Numerical control: Mathematics and applications, 1972. 4.2.1

[15] D. Le Bihan. Molecular Diffusion, Tissue Microdynamics and Microstructure. *NMR Biomed*, 8(7-8):375–386, 1995. 9.1

[16] D. Le Bihan, J.-F. Mangin, C. Poupon, Chris A. Clark, Sabina Pappata, Nicolas Molko, and Hughes Chabriat. Diffusion Tensor Imaging: Concepts and Applications. *J Magn Reson Imaging*, 13:534–546, 2001. 9

[17] M. A. Biot. General Theory of Three-Dimensional Consolidation. *J Appl Phys*, 12(2):155–164, 1941. 5.2.3

[18] M. Blackwell, C. Nikou, A. M. DiGioia, and T. Kanade. An Image Overlay System for Medical Data Visualization. *Med Image Anal*, 4(1):67–72, 2000. 2.2

[19] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans Pattern Anal Mach Intell*, 11(6):567–585, 1989. 2.4.1.1

[20] G. Borgefors. Hierarchical Chamfer Matching: A Parametric Edge Matching Algorithm. *IEEE T Pattern Anal*, 10(6):849–865, 1988. 2.3.1.2

[21] D. W. A. Brands, G. W. M. Peters, and P. H. M. Bovendeerd. Design and Numerical Implementation of a 3-D Non-Linear Viscoelastic Constitutive Model for Brain Tissue During Impact. *J Biomech*, 37(1):127–134, 2004. 5.2.2

[22] M. Bro-Nielsen. Surgery Simulation Using Fast Finite Elements. In *Proc. Visualization in Biomedical Computing*, number 1131 in Lect Notes Comp Sc, pages 529–534. Springer, 1996. 8.1

[23] M. Bro-Nielsen. Finite Element Modeling in Surgery Simulation. *IEEE Special Issue on Virtual & Augmented Reality in Medicine*, 86(3):490–503, 1998. 8.1

[24] M. Bro-Nielsen and C. Gramkow. Fast Fluid Registration of Medical Images. In *Proc. Visualization in Biomedical Computing*, Lect Notes Comp Sc, pages 267–276. Springer, 1996. 2.4.1.3

[25] M. Bro-Nielsen, D. Helfrick, B. Glass, X. Zeng, and H. Connacher. VR Simulation of Abdominal Trauma Surgery. *Stud Health Technol Inform*, 50:117–123, 1998. 8.1

[26] L. Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, Dez 1992. 2

[27] M. Buist, G. Sands, P. Hunter, A. Pullan, and G. Sands. A Deformable Finite Element Derived Finite Difference Method for Cardiac Activation Problems. *Ann Biomed Eng*, 31(5):577–588, 2003. 5.1

[28] B. Cabral, N. Cam, and J. Foran. Accelerated Volume Rendering and Tomographic Reconstruction Using Texture Mapping Hardware. *ACM Symp. on Vol. Vis.*, pages 91–98, 1994. 3.3.2

[29] J. Cai, J. C. H. Chu, D. Recine, M. Sharma, C. Nguyen, R. Rodebaugh, and V. A. Saxena. CT and PET Lung Image Registration and Fusion in Radiotherapy Treatment Planning using the Chamfer-Matching Method. *Int J Radiat Oncol*, 43(4):883–891, 1999. 2.3.1.2

[30] N. Casap, A. Wexler, N. Persky, A. Schneider, and J. Lustmann. Navigation Surgery for Dental Implants: Assessment of Accuracy of the Image Guided Implantology System. *J Oral Maxil Surg*, 62(suppl. 2):116–119, 2004. 2.2

[31] M. Chabanas, V. Luboz, and Y. Payan. Patient Specific Finite Element Model of the Face Soft Tissues for Computer-Assisted Maxillofacial Surgery. *Med Image Anal*, 7(2):131–151, 2003. 5.1

[32] E. Chen, R. E. Ellis, J. T. Bryant, and J. F. Rudan. A Computational Model of Postoperative Knee Kinematics. *Med Image Anal*, 5(4):317–330, 2001. 5.1

[33] C. H. Chow, H. T. Tsui, and T. Lee. Surface Registration using a Dynamic Genetic Algorithm. *Pattern Recogn*, 37(1):105–117, 2004. 2.3.1.2

[34] G. Christensen, R. Rabbit, and M. Miller. A deformable neuroanatomy textbook based in viscous fluid mechanics. In *Proc. Conf. Inform. Sci. and Sys*, pages 211–216, 1993. 2.4.1.1

[35] G. Christensen, R. Rabbit, and M. Miller. Deformable Templates Using Large Deformation Kinematics. *IEEE T Image Process*, 18:875–884, 1996. 2.4.1.3

[36] C. Chua and U. Neumann. Hardware-Accelerated Free-Form Deformation. In *Proc. ACM SIGGRAPH/Eurographics Workshop on Graphics Hardware*, pages 33–39. ACM Press, 2000. 4.1

[37] H. Chui and A. Rangarajan. A New Point Matching Algorithm for Non-Rigid Registration. *Comput Vis Image Und*, 89(2-3):114–141, 2003. 2.4.1.1

[38] R. W. Clough. The Finite Element Method in Plane Stress Analysis. In *Proc. ASCE Conference on Electronic Computation*, 1960. 6.1

[39] F. S. Cohen and C. Pintavirooj. Invariant Surface Alignment in the Presence of Affine and Some Nonlinear Transformations. *Med Image Anal*, 8(2):151–164, 2004. 2.4.1.2

[40] A. Collignon, D. Vandermeulen, P. Suetens, and G. Marchal. Automated Multi-Modality Image Registration Based on Information Theory. *Information Processing in Medical Imaging*, 3:263–274, 1995. 2.3.1.3, 3.2

[41] S. Coquillart. Extended Free-Form Deformation: a Sculpturing Tool for 3D Geometric Modeling. In *Proc. ACM SIGGRAPH*, pages 187–196. ACM Press, 1990. 4.1

[42] S. Cotin, H. Delingette, and N. Ayache. A Hybrid Elastic Model allowing Real-Time Cutting, Deformations and Force-Feedback for Surgery Training and Simulation. *Visual Comput*, 16(8):437–452, 2000. 5.1, 8.1

[43] S. Cotin, H. Delingette, J.-M. Clement, V. Tassetti, J. Marescaux, and N. Ayache. Volumetric Deformable Models for Simulation of Laparoscopic Surgery. In *Proc. International Symposium on Computer and Communication Systems for Image Guided Diagnosis and Therapy, Computer Assisted Raiology (CAR)*, volume 1124, pages 793–798. Elsevier, 1996. 5.1

[44] M. Bach Cuadra, C. Pollo, A. Bardera, O. Cuisenaire, J. Villemure, and J. Thiran. Atlas-Based Segmentation of Pathological Brain MR Images. In *Proc. International Conference on Image Processing*, volume 1, pages 14–17, 2003. 2.4.1.3

[45] D. Culler, R. Karp, D. Patterson, A. Sahay, K. E. Schauser, E. Santos, R. Subramonian, and T. van Eicken. Logp: Towards a realistic model of parallel computation. In *Proc. ACM SIGPLAN*, pages 1–12. ACM Press, 1993. 8.4

[46] E. D'Agostino, F. Maes, D. Vandermeulen, and P. Suetens. A Viscous Fluid Model for Multimodal Non-Rigid Image Registration using Mutual Information. *Med Image Anal*, 7(4):565–575, 2003. 2.4.1.3

[47] K. K. Darvish and J. R. Crandall. Nonlinear Viscoelastic Effects in Oscillatory Shear Deformation of Brain Tissue. *Med Eng Phys*, 23(9):633–645, 2001. 5.2.2

[48] C. Davatzikos. Spatial Transformation and Registration of Brain Images Using Elastically Deformable Models. *Comput Vis Image Und*, 66(2):207–222, 1997. 2.4.1.3

[49] M. H. Davis, A. Khotanzad, D. P. Flamig, and S. E. Harms. A Physics-based Coordinate Transformation for 3-D Image Matching. *IEEE Trans Med Imaging*, 16(3):317–328, 1997. 2.4.1.1

[50] H. Delingette, G. Subsol, S. Cotin, and J. Pignon. Craniofacial Surgery Simulation Testbed. In *Proc. Visualization in Biomedical Computing*, volume 2359, pages 607–618. SPIE, 1994. 5.1

[51] J. J. Dongarra, R. Hempel, A. J. G. Hey, and D. W. Walker. A proposal for a User-Level, Message Passing Interface in a Distributed Memory Environment. Technical Report Technical Report TM-12231, Oak Ridge National Laboratory, 1993. 8.5

[52] L. Dougherty, J. C. Asmuth, A. S. Blom, L. Axel, and R. Kumar. Validation of an Optical Flow Method for Tag Displacement Estimation. *IEEE Trans Med Imaging*, 18(4):359–363, 1999. 2.4.1.3

[53] G. Eckel. *OpenGL Volumizer Programmer's Guide*. Silicon Graphics, Inc., 1998. 4.1

[54] D. Hill et al. Registration of MR and CT images for skull base surgery using pointlike anatomical features. *Brit. J. Radiol.*, 64:1030–1035, 1991. 2.3.1.1

[55] A. C. Evans, S. Marrett, J. Torrescorzo, S. Ku, and L. Collins. MRI-PET Correlation in Three Dimensions using a Volume-of-Interest (VOI) Atlas. *J Cereb Blood Flow Metab*, 11(2):69–78, 1991. 2.3.1.1

[56] S. Fang, S. Huang, R. Srinivasan, and R. Raghavan. Deformable Volume Rendering by 3D Texture Mapping and Octree Encoding. In *Proc. IEEE Visualization*, pages 73–ff. IEEE Computer Society Press, 1996. 4.1

[57] M. Ferrant, A. Nabavi, B. Macq, P. M. Black, F. A. Jolesz, R. Kikinis, and S.K. Warfield. Serial Registration of Intraoperative MR Images of the Brain. *Med Image Anal*, 6(4):337–359, 2002. 2.2, 2.4, 2.4.1.3, 4, 5.2.1, 7.4

[58] B. Fischer and J. Modersitzki. A Unified Approach to Fast Image Registration and a New Curvature Based Registration Technique. *Linear Algebra Appl*, 380:107–124, 2004. 2.4.1.3

[59] J. M. Fitzpatrick, J. B. West, and C. R. Maurer. Predicting Error in Rigid-Body Point-based Registration. *IEEE Trans Med Imaging*, 17(5):694–702, 1998. 8

[60] M. Fujioka, S. Yokoi, T. Yasuda, Y. Hashimoto, J. Toriwaki, and H. Nakajima. Computer-aided Interactive Surgical Simulation for Craniofacial Anomalies Based on 3-D Surface Reconstruction CT Images. *Radiat Med*, 6(5):204–212, 1988. 5.1

[61] N. Gagvani and D. Silver. Animating Volumetric Models. *Graph Models*, 63(6):443–458, 2001. 4.1

[62] K. A. Ganser, H. Dickhaus, and C. R. Wirtz. A Deformable Digital Brain Atlas System according to Talairach and Tournoux. *Med Image Anal*, 8(1):3–22, 2004. 2.2, 2.4

[63] M. Garland and P. Heckbert. Surface simplification using quadric error metrics. In *Proc. SIGGRAPH*, pages 209–216. ACM Computer Graphics, 1997. 6.4.2

[64] Y. Ge, C. R. Maurer, and J. Fitzpatrick. Surface based 3-d image registration using the iterative closest point algorithm with a closest point transform. In *Proc. SPIE Medical Imaging*, volume 2710, pages 358–367. ACM Computer Graphics, 1996. 2.3.1.2

[65] S. Gibson, C. Fyock, E. Grimson, T. Kanade, R. Kikinis, H. Lauer, N. McKenzie, A. Mor, S. Nakajima, and H. Ohkami. Volumetric Object Modeling for Surgical Simulation. *Med Image Anal*, 2(2):121–132, 1998. 5.1

[66] A. C. Godest, M. Beaugonin, E. Haug, M. Taylor, and P. J. Gregson. Simulation of a Knee Joint Replacement During a Gait Cycle Using Explicit Finite Element Analysis. *J Biomech*, 35(2):267–275, 2002. 5.1

[67] D. Haemmerich, L. Chachati, A. S. Wright, D. M. Mahvi, F. T. Lee, and J. G. Webster. Measuring Forces in Liver Cutting: New Equipment and Experimental Results. *IEEE T Bio-Med Eng*, 50(4):493–500, 2003. 5.1

[68] A. Hagemann, K. Rohr, and H.S. Stiehl. Coupling of Fluid and Elastic Models for Biomechanical Simulations of Brain Deformations using FEM. *Med Image Anal*, 6(4):375–388, 2002. 2.4.1.3, 5.2.1

[69] A. Hagemann, K. Rohr, H.S. Stiehl, U. Spetzger, and J.M. Gilsbach. Biomechanical Modeling of the Human Head for Physically Based, Non-Rigid Image Registration. *IEEE Trans Med Imaging*, 18(10):875–884, 1999. 2.4.1.3, 5.2.1, 7.1

[70] J. V. Hajnal, N. Saeed, E. J. Soar, A. Oatridge, I. R. Young, and G. M. Bydder. A registration and Interpolation Procedure for Subvoxel Matching of Serially Acquired MR Images. *J Comput Assist Tomogr*, 19(2):289–296, 1995. 2.3.1.3

[71] T. Hartkens, D. L. G. Hill, D. J. Hawkes, C. R. Maurer, A. J. Martin, W. A. Hall, H. Liu, and C. L. Truwit. Measurement and Analysis of Brain Deformation during Neurosurgery. *IEEE Trans Med Img*, 22(1):82–92, 2003. 2.2, 2.4

[72] P. Hastreiter. *Registrierung und Visualisierung medizinischer Bilddaten unterschiedlicher Modalitäten*. PhD thesis, Computer Graphics Group, Computer Science Department, University of Erlangen-Nuremberg, Germany, 1999. 1.1, VII

[73] P. Hastreiter, K. Engel, G. Soza, M. Bauer, M. Wolf, O. Ganslandt, R. Fahlbusch, G. Greiner, T. Ertl, and C. Nimsky. Remote Analysis for Brain Shift Compensation. In *Proc. MICCAI*, Lect. Notes in Comp. Sc., pages 1248–1249. Springer, 2001. 1.1, VII

[74] P. Hastreiter, K. Engel, G. Soza, M. Wolf, O. Ganslandt, R. Fahlbusch, G. Greiner, and C. Nimsky. Remote Computing Environment Compensating for Brain Shift. *Comput Aided Surg*, 8(4):169–179, 2003. 1.1, VII

[75] P. Hastreiter and T. Ertl. Integrated Registration and Visualization of Medical Image Data. In *Proc. CGI*, pages 78–85, Hannover, Germany, 1998. 2.3.1.3, 3.3, 3.3.2

[76] P. Hastreiter, C. Rezk-Salama, C. Nimsky, C. Lürig, G. Greiner, and T. Ertl. Registration Techniques for the Analysis of the Brain Shift in Neurosurgery. *Comput Graph*, 24(3):385–389, 2000. 2.4.1.2

[77] P. Hastreiter, C. Rezk-Salama, G. Soza, G. Greiner, R. Fahlbusch, O. Ganslandt, and C. Nimsky. Strategies for Brain Shift Evaluation. *Med Image Anal*, 8(4):447–464, 2004. 1.1, VII

[78] G. Hirota, R. Maheshwari, and M. C. Lin. Fast Volume-Preserving Free-Form Deformation using Multi-Level Optimization. *Comput Aided Design*, 32(8-9):499–512, 2000. 4.1

[79] B. K. Horn and B. G. Schunck. Determining Optical Flow. *Artif Intell*, 17(1-3):185–203, 1981. 2.4.1.3

[80] D. Huang, M. Tang, and R. Shekhar. Mathematical Model of Corneal Surface Smoothing after Laser Refractive Surgery. *Am J Ophthalmol*, 135(2):267–278, 2003. 5.1

[81] P. Jezzard, A. S. Barnett, and C. Pierpaoli. Characterization of and Correction for Eddy Current Artifacts in Echo Planar Diffusion Imaging. *Magnet Reson Med*, 39(5):801–812, 1998. 9.1

[82] D. K. Jones, M. A. Horsfield, and A. Simmons. Optimal Strategies for Measuring Diffusion in Anisotropic Systems by Magnetic Resonance Imaging. *Magnet Reson Med*, 42(3):515–525, 1999. 9.1

[83] L. Junck, J. G. Moen, G. D. Hutchins, M. B. Brown, and D. E. Kuhl. Correlation Methods for the Centering, Rotation, and Alignment of Functional Brain Images. *J Nucl Med*, 31(7):1220–1226, 1990. 2.3.1.3

[84] G. Karypis and V. Kumar. Multilevel k-Way Partitioning Scheme for Irregular Graphs. *J Parallel Distr Comp*, 48(1):96–129, 1998. 8.2.1

[85] G. Karypis and V. Kumar. Parallel Multilevel k-Way Partitioning Scheme for Irregular Graphs. *Siam Rev*, 41(2):278–300, 1999. 8.2.1

[86] E. Keeve, S. Girod, R. Kikinis, and B. Girod. Deformable Modeling of Facial Tissue for Craniofacial Surgery Simulation. *Comput Aided Surg*, 3(5):228–238, 1998. 5.1

[87] T.A. Krouskop, T.M. Wheeler, F. Kallel, B. Garra, and T. Hall. The Elastic Moduli of Breast and Prostate Tissues Under Compression. *Ultrason Imaging*, 20:151–159, 1998. 7.1

[88] Y. Kurzion and R. Yagel. Space Deformation using Ray Deflectors. In *Proc. Eurographics Workshop on Rendering*, pages 21–32, 1995. 4.1

[89] Y. Kurzion and R. Yagel. Interactive Space Deformation with Hardware-Assisted Rendering. *IEEE Comput Graph*, 17(5):66–77, 1997. 4.1

[90] U. Labsik. *Multiresolution Mesh Processing for Triangular and Tetrahedral Meshes*. PhD thesis, Computer Graphics Group, Computer Science Department, University of Erlangen-Nuremberg, Germany, 2003. 6.4.2

[91] L. Lemieux, R. Jagoe, D. R. Fish, N. D. Kitchen, and D. G. Thomas. A Patient-to-Computed-Tomography Image Registration Method Based on Digitally Reconstructed Radiographs. *Med Phys*, 21(11):1749–1760, 1994. 2.3.1.3

[92] H. Lester, S. R. Arridge, K. M. Jansons, L. Lemieux, J. V. Hajnal, and A. Oatridge. Non-Linear Registration with the Variable Viscosity Fluid Algorithm. In *Proc. International Conference on Information Processing in Medical Imaging*, Lect Notes Comp Sc, pages 238–251. Springer, 1999. 2.4.1.3

[93] T. Liu, D. Shen, and C. Davatzikos. Deformable Registration of Cortical Structures via Hybrid Volumetric and Surface Warping. *Neuroimage*, 22(4):1790–1801, 2004. 2.4.1.2

[94] W. Lorensen and H. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In *Proc. SIGGRAPH*, pages 163–169. ACM Press, 1987. 6.4.2

[95] M. Lorenzo-Valdes, G. I. Sanchez-Ortiz, R. Mohiaddin, and D. Rueckert. Atlas-based Segmentation and Tracking of 3D Cardiac MR Images using Non-Rigid Registration. In *Proc. MICCAI*, Lect Notes Comp Sc, pages 642–650. Springer, 2002. 2.4.1.3

[96] V. Luboz, A. Pedrono, D. Ambard, F. Boutault, Y. Payan, and P. Swider. Prediction of Tissue Decompression in Orbital Surgery. *Clin Biomech*, 19(2):202–208, 2004. 5.1

[97] K. E. Lunn, K. D. Paulsen, D. W. Roberts, F. E. Kennedy, A. Hartov, and L. A. Platenik. Nonrigid Brain Registration: Synthesizing Full Volume Deformation Fields from Model Basis Solutions Constrained by Partial Volume Intraoperative Data. *Comput Vis Image Und*, 89(2-3):299–317, 2003. 2.2

[98] F. Maes, D. Vandermeulen, and P. Suetens. Comparative Evaluation of Multiresolution Optimization Strategies for Multimodality Image Registration by Maximization of Mutual Information. *Med Image Anal*, 3(4):373–386, 1999. 3.3.3

[99] J. B. Maintz, P. A. van den Elsen, and M. A. Viergever. Comparison of Edge-based and Ridge-based Registration of CT and MR Brain Images. *Med Image Anal*, 1(2):151–161, 1996. 2.3.1.2

[100] A. Manduca, D. S. Lake, S. A. Kruse, and R. L. Ehman. Spatio-temporal Directional Filtering for Improved Inversion of MR Elastography Images. *Med Image Anal*, 7(4):465–473, 2003. 7.1

[101] R. Marmulla, T. Luth, J. Muhling, and S. Hassfeld. Automated Laser Registration in Image-guided Surgery: Evaluation of the Correlation between Laser Scan Resolution and Navigation Accuracy. *J Oral Maxil Surg*, 33(7):642–648, 2004. 2.3.1.2

[102] C. Maurer, D. Hill, A. Martin, H. Liu, M. McCue, D. Rueckert, D. Lloret, W. Hall R. Maxwell D. Hawkes, and C. Truwit. Investigation of Intraoperative Brain Deformation Using a 1,5 T Interventional MR System: Preliminary Results. *IEEE Trans Med Imaging*, 17(5):817–825, 1998. 1, 2.4, VII

[103] C. R. Maurer, J. M. Fitzpatrick, M. Y. Wand, R. L. Galloway, R. J. Maciunas, and G. S. Allen. Registration of Head Volume Images using Implantable Fiducial Markers. *IEEE Trans Med Imaging*, 16(4):447–462, 1997. 2.3.1.1

[104] C. R. Maurer, R. J. Maciunas, and J. M. Fitzpatrick. Registration of Head CT Images to Physical Space using a Weighted Combination of Points and Surfaces. *IEEE Trans Med Imaging*, 17(5):753–761, 1998. 2.3.1.1

[105] D. Merhof, P. Hastreiter, G. Soza, M. Stamminger, and Ch. Nimsky. Nonlinear Integration of DTI-based Fiber Tracks into Standard 3D MR Data. In *Proc. VMV (Vision, Modeling and Visualization)*, pages 371–377, 2004. 1.1, VII

[106] M. Miga. A New Approach to Elastography Using Mutual Information and Finite Elements. *Phys Med Biol*, 48:467–480, 2003. 2.2

[107] M. Miga, K. Paulsen, P. Hoopes, F. Kennedy, and A. Hartov. In Vivo Modeling of Intersitial Pressure in the Brain under Surgical Load Using Finite Elements. *J Biomech Eng*, 122:354–363, 2000. 1.1, 2.2, 7.4, 7.5, VII

[108] M. Miga, K. Paulsen, P. Hoopes, F. Kennedy, A. Hartov, and D. Roberts. In Vivo Analysis of Heterogeneous Brain Deformation Computations for Model-Updated Image Guidance. *Comput Methods Biomech Biomed Eng*, 3(2):129–146, 2000. 5.2.3, 5.2.4

[109] M. Miga, K. Paulsen, P. Hoopes, F. Kennedy, A. Hartov, and D. Roberts. In Vivo Modeling of Interstitial Pressure in the Brain under Surgical Load Using Finite Elements. *J Biomech Eng*, 122(4):354–363, 2000. 5.2.3

[110] M. Miga, K. Paulsen, F. Kennedy, A. Hartov, and D. Roberts. Model-Updated Image Guided Neurosurgery Using the Finite Element Method: Incorporation of the Falx Cerebri. In *Proc. MICCAI*, Lect Notes Comp Sc, pages 900–909. Springer, 1999. 5.2.3

[111] M. Miga, A. Staubert, K. Paulsen, F. Kennedy V. Tronnier, and D. Roberts. Model Updated Image Guidance: Initial Clinical Experiences with Gravity-Induced Brain Deformation. *IEEE Trans. Med. Imaging*, 18:866–874, 1999. 5.2.3

[112] K. Miller. Technical Note: Modelling Soft Tissue Using Biphasic Theory - A Word of Caution. *Comput Methods Biomech Biomed Engin*, 1(3):261–263, 1998. 5.2.4

[113] K. Miller. Constitutive Modelling of Abdominal Organs. *J Biomech*, 33(3):367–373, 2000. 5.1

[114] K. Miller and K. Chinzei. Constitutive Modelling of Brain Tissue: Experiment and Theory. *J Biomech*, 30(11-12):1115–1121, 1997. 5.2.2, 5.2.4, 7.4, 7.5

[115] K. Miller, K. Chinzei, G. Orssengo, and P. Bednarz. Mechanical Properties of Brain Tissue In-vivo: Experiment and Computer Simulation. *J Biomech*, 33(11):1369–1376, 2000. 5.2.2, 5.2.4, 7.1

[116] T. J. A. Mommersteeg, R. Huiskes, L. Blankevoort, J. G. M. Kooloos, J. M. G. Kauer, and P. G. M. Maathuis. A Global Verification Study of a Quasi-Static Knee Model with Multi-Bundle Ligaments. *J Biomech*, 29(12):1659–1664, 1996. 5.1

[117] C. Monserrat, U. Meier, M. Alcaniz, F. Chinesta, and M. C. Juan. A New Approach for the Real-Time Simulation of Tissue Deformations in Surgery Simulation. *Comput Meth Prog Bio*, 64(2):77–85, 2001. 8.1

[118] G. E. Moore. Cramming More Components Onto Integrated Circuits. *Electronics*, 38(8):114–117, 1965. 3

[119] V. C. Mow, S. C. Kuei, W. M. Lai, and C. G. Armstrong. Biphasic Creep and Stress Relaxation of Articular Cartilage in Compression: Theory and Experiments. *J Biomech Eng*, 102(1):73–84, 1980. 5.2.3

[120] T. Nagashima, N. Tamaki, S. Matsumoto, B. Horwitz, and Y. Seguchi. Biomechanics of Hydrocephalus: a New Theoretical Model. *Neurosurgery*, 21(6):898–904, 1987. 5.2.3

[121] P. M. Nielsen, I. J. Le Grice, B. H. Smaill, and P. J. Hunter. Mathematical Model of Geometry and Fibrous Structure of the Heart. *Am J Physiol Heart Circ Physiol*, 260(4):1365–1378, 1991. 5.1

[122] C. Nimsky, O. Ganslandt, S. Cerny, P. Hastreiter, G. Greiner, and R. Fahlbusch. Quantification of, visualization of, and compensation for brain shift using intraoperative magnetic resonance imaging. *Neurosurgery*, 47(5):1070–1080, 2000. 1, VII

[123] C. Nimsky, O. Ganslandt, P. Hastreiter, R. Wang, T. Benner, A. G. Sorensen, and R. Fahlbusch. Intraoperative Diffusion-Tensor MR Imaging: Shifting of White Matter Tracts During Neurosurgical Procedures–Initial Experience. *Radiology*, 234(1):218–225, 2005. 9.1

[124] J. Ophir, F. Kallel, T. Varghese, E. Konofagou, S. K. Alam, T. Krouskop, B. garra, and R. Righetti. Elastography. *Comptes Rendus de l'Acadmie des Sciences - Series IV - Physics*, 2(8):1193–1212, 2002. 7.1

[125] J. Orchard, C. Greif, G. H. Golub, B. Bjornson, and M. S. Atkins. Simultaneous Registration and Activation Detection for fMRI. *IEEE Trans Med Imaging*, 22(11):1427–1435, 2003. 2.2

[126] H. Ozawa, T. Matsumoto, T. Ohashi, M. Sato, and S. Kokubun. Comparison of Spinal Cord Gray Matter and White Matter Softness: Measurement by Pipette Aspiration Method. *J Neurosurg Spine*, 95(2):221–224, 2001. 5.2.4

[127] S. M. Paling, E. D. Williams, R. Barber, E. J. Burton, W. R. Crum, N. C. Fox, and J. T. O'Brien. The Application of Serial MRI Analysis Techniques to the Study of Cerebral Atrophy in Late-Onset Dementia. *Med Image Anal*, 8(1):69–79, 2004. 2.2

[128] C. Pelizzari, G. Chen, D. Spelbring, R. Weichselbaum, and C. Chen. Accurate three–diemensional registration of CT, PET and/or MR Images of the brain. *Comp. Assis. Tomogr.*, pages 20–26, 1989. 2.3.1.2

[129] T. Peters, B. Davey, P. Munger, R. Comeau, A. Evans, and A. Olivier. Three–dimensional multimodal image–guidance for neurosurgery. *IEEE Trans. Med. Img.*, 15, 1996. 2.3.1.1

[130] J. P. Pluim and J. M. Fitzpatrick. Image Registration. *IEEE Trans Med Imaging*, 22(11):1341–1343, 2003. 2.4.1

[131] J. P. W. Pluim, J. B. A Maintz, and M. A. Viergever. Mutual Information Matching in Multiresolution Contexts. *Image Vision Comput*, 19(1-2):45–52, 2001. 3.3.3

[132] M. T. Prange and S. S. Margulies. Regional, Directional, and Age-Dependent Properties of the Brain Undergoing Large Deformation. *J Biomech Eng*, 124(2):244–252, 2002. 5.2.4

[133] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C++*. Cambridge University, New York, 2002. 2.3.1.2, 7.2.3

[134] D. Rey, G. Subsol, H. Delingette, and N. Ayache. Automatic Detection and Segmentation of Evolving Processes in 3D Medical Images: Application to Multiple Sclerosis. *Med Image Anal*, 6(2):163–179, 2002. 2.2

[135] C. Rezk-Salama, M. Scheuering, G. Soza, and G. Greiner. Fast Volumetric Deformation on General Purpose Hardware. In *Proc. SIGGRAPH/Eurographics Workshop on Graphics Hardware*, pages 17–24, 2001. 4.4

[136] A. Roche, G. Malandain, X. Pennec, and N. Ayache. The Correlation Ratio as a New Similarity Measure for Multimodal Image Registration. In *Proc. MICCAI*, Lect Notes Comp Sc, pages 1115–1124. Springer, 1998. 2.3.1.3, 3.2

[137] T. Rohlfing, C. R. Maurer, D. Dean, and R. J. Maciunas. Effect of Changing Patient Position from Supine to Prone on the Accuracy of a Brown-Roberts-Wells Stereotactic Head Frame System. *Neurosurgery*, 52(3):610–618, 2003. 2.3.1.1

[138] T. Rohlfing, J. B. West, J. Beier, T. Liebig, C. A. Taschner, and U. W. Thomale. Registration of Functional and Anatomical MRI: Accuracy Assessment and Application in Navigated Neurosurgery. *Comput Aided Surg*, 5(6):414–425, 2000. 2.2

[139] K. Rohr, H. S. Stiehl, R. Sprengel, T. M. Buzug, J. Weese, and M. H. Kuhn. Landmark-based Elastic Registration using Approximating Thin-Plate Splines. *IEEE Trans Med Imaging*, 20(6):526–534, 2001. 2.4.1.1

[140] W. Röntgen. Über eine neue Art von Strahlen (Vorläufige Mitteilung). In *Sitzungsberichte der Physik.-med. Gesellschaft zu Würzburg*, pages 132–141, 1896. 1, VII

[141] D. Rueckert, A. F. Frangi, and J. A. Schnabel. Automatic Construction of 3-D Statistical Deformation Models of the Brain using Nonrigid Registration. *IEEE Trans Med Img*, 22(8):1014–1025, 2003. 2.2

[142] D. Rueckert, L. I. Sonoda, D.L.G. Hill C. Hayes, M.O. Leach, and D.J. Hawkes. Nonrigid Registration Using Free-Form Deformations: Application to Breast MR Images. *IEEE Trans Med Img*, 18(8):712–721, 1999. 2.2, 2.4, 2.4.1.3, 4, 4.3.2

[143] Y. Saad and M. H. Schultz. GMRES: a Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM J Sci Stat Comp*, 7(3):856–869, 1986. 6.7.2

[144] M. A. Sagar, D. Bulivant, G. Mallinson, P. Hunter, and I. Hunter. A Virtual Environment and Model of the Eye for Surgical Simulation. In *Proc. SIGGRAPH*, pages 205–212. ACM Press, 1994. 5.1

[145] G. Sakas. Trends in Medical Imaging: from 2D to 3D. *Comput Graph*, 26(4):577–587, 2002. 1, VII

[146] L. Sandrin, M. Tanter, S. Catheline, and M. Fink. Shear Modulus Imaging with 2-D Transient Elastography. *IEEE Trans Ultrason Ferroelectr Freq Control*, 49(4):426–435, 2002. 7.1

[147] J.-C. Sarron, C. Blondeau, A. Guillaume, and D. Osmont. Identification of Linear Viscoelastic Constitutive Models. *J Biomech*, 33(6):685–693, 2000. 5.2.2

[148] F. Sauer, U. J. Schoepf, A. Khamene, S. Vogt, M. Das, and S. G. Silverman. Augmented reality system for ct-guided interventions: System description and initial phantom trials. In *Proc. SPIE Medical Imaging*, volume 5029. ACM Computer Graphics, 2003. 2.2

[149] J. Schnabel, D. Rueckert, J. M. Blackall, A. D. Castellano-Smith, T. Hartkens, G. P. Penney, W. A. Hall, H. Liu, C. L. Truwit, F. A. Gerritsen, D. L. G. Hill, and D. J. Hawkes. A Generic Framework for Non-Rigid Registration Based on Non-Uniform Multi-Level Free-Form Deformations. In *Proc. MICCAI*, Lect Notes Comp Sc, pages 573–581. Springer, 2001. 2.4.1.3

[150] Thomas W. Sederberg and Scott R. Parry. Free-form deformation of solid geometric models. *Computer Graphics*, 20(4):151–160, 1986. 2.4.1.3, 4.1, 4.2.1

[151] C. Shannon and W. Weaver. *The mathematical theory of communication*. University of Illinois Press, 1949. 2.3.1.3, 3.2

[152] R. E. Showalter. Diffusion in Poro-Elastic Media. *J Math Anal Appl*, 251:310–340, 2000. 5.2.3, 6, 6.2

[153] O. Skrinjar and J. Duncan. Real Time 3D Brain Shift Compensation. In A. Kuba, M. Samal, and A. Todd-Pokropek, editors, *Proc. Information Processing in Medical Imaging (IPMI)*, pages 42–55. Springer, 1999. 5.2.1

[154] O. Skrinjar, C. Studholme, A. Nabavi, and J. Duncan. Steps Toward a Stereo-Camera-Guided Biomechanical Model for Brain Shift Compensation. In *Proc. IPMI 2001*, Lect. Notes in Comp. Sc., pages 183–189. Springer, 2001. 5.2.1

[155] S. M. Smith. Fast Robust Automated Brain Extraction. *Hum Brain Mapp*, 17(3):143–155, 2002. 4.2.8, 6.4.1

[156] J. Sobotta. *Atlas der Anatomie des Menschen, Band 1: Kopf, Hals, Obere Extremität*. Urban & Fischer, 2000. 6.8

[157] G. Soza, M. Bauer, P. Hastreiter, Ch. Nimsky, and G. Greiner. Non-rigid Registration with Use of Hardware-Based 3D Bézier Functions. In *Proc. MICCAI*, pages 549–556. Lect Notes Comput Sc, Springer, 2002. 1.1, VII

[158] G. Soza, R. Grosso, P. Hastreiter, U. Labsik, Ch. Nimsky, R. Fahlbusch, and G. Greiner. Fast and Adaptive Finite Element Approach for Modeling Brain Shift. In *Proc. CURAC (Dt. Gesell. für Computer- und Roboterassistierte Chirurgie)*, 2003. 1.1, VII

[159] G. Soza, R. Grosso, P. Hastreiter, U. Labsik, Ch. Nimsky, R. Fahlbusch, and G. Greiner. Fast and Adaptive Finite Element Approach for Modeling Brain Shift. *Comput Aided Surg*, 8(5):241–246, 2004. 1.1, VII

[160] G. Soza, R. Grosso, Ch. Nimsky, G. Greiner, R. Fahlbusch, and P. Hastreiter. Combined Simulation and Registration for the Determination of the Mechanical Properties of Brain Tissue. In *Proc. CURAC (Dt. Gesell. für Computer- und Roboterassistierte Chirurgie)*, 2003. 1.1, VII

[161] G. Soza, R. Grosso, Ch. Nimsky, G. Greiner, and P. Hastreiter. Estimating Mechanical Brain Tissue Properties with Simulation and Registration. In *Proc. MICCAI*, volume 2, pages 276–283. Lect Notes Comput Sc, Springer, 2004. 1.1, VII

[162] G. Soza, R. Grosso, Ch. Nimsky, G. Greiner, and P. Hastreiter. High Performance Implementation for the Simulation of Brain Deformation. In *Proc. BVM (Bildverarbeitung für die Medizin)*, pages 455–459. Springer, 2005. 1.1, VII

[163] G. Soza, R. Grosso, Ch. Nimsky, P. Hastreiter, R. Fahlbusch, and G. Greiner. Determination of the Elasticity Parameters of Brain Tissue with Combined Simulation and Registration. *Int J Med Robot*, 2005. in press. 1.1, VII

[164] G. Soza, P. Hastreiter, M. Bauer, C. Rezk-Salama, C. Nimsky, and G. Greiner. Intraoperative Registration on Standard PC Graphics Hardware. In *Proc. BVM (Bildverarbeitung für die Medizin)*, pages 334–337, 2002. 1.1, VII

[165] G. Soza, P. Hastreiter, F. Vega, C. Rezk-Salama, M. Bauer, Ch. Nimsky, and G. Greiner. Non-linear Intraoperative Correction of Brain Shift with 1.5 T Data. In *Proc. BVM (Bildverarbeitung für die Medizin)*, pages 21–25. Springer, 2003. 1.1, VII

[166] J. C. Spall. Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation. *IEEE T Automat Contr*, 37(3):332–341, 1992. 4.2.6, 4.2.6, 4.3.3

[167] J. C. Spall. Implementation of the Simultaneous Perturbation Algorithm for Stochastic Optimization. *IEEE T Aero Elec Sys*, 34(3):817–823, 1998. 4.2.8

[168] R. Stefanescu, X. Pennec, and N. Ayache. Grid Powered Nonlinear Image Registration with Locally Adaptive Regularization. *Med Image Anal*, 8(3):325–342, 2004. 2.4.1.3

[169] E. O. Stejskal and J. E. Tanner. Spin Diffusion Measurements: Spin Echoes in the Presence of a Time-Dependent Field Gradient. *J Chem Phys*, 42(1):288–292, 1965. 9.2.2

[170] C. Studholme, R. T. Constable, and J. S. Duncan. Accurate Alignment of Functional EPI Data to Anatomical MRI using a Physics-based Distortion Model. *IEEE Trans Med Imaging*, 19(11):1115–1127, 2000. 2.4

[171] C. Studholme, D. L. G. Hill, and D. J. Hawkes. Automated 3-D Registration of MR and CT Images of the Head. *Med Image Anal*, 1(2):163–175, 1996. 3.3.3

[172] C. Studholme, D. L. G. Hill, and D. J. Hawkes. An Overlap Invariant Entropy Measure of 3D Medical Image Alignment. *Pattern Recogn*, 32(1):71–86, 1999. 2.3.1.3, 3.2, 3.2, 7.2.1

[173] G. Subsol, J. P. Thirion, and N. Ayache. A scheme for Automatically Building Three-dimensional Morphometric Anatomical Atlases: Application to a Skull Atlas. *Med Image Anal*, 2(1):37–60, 1998. 2.4.1.2

[174] G. Szekely, Ch. Brechbuhler, A. Rhomberg, N. Ironmonger, and P. Schmid. Modelling of Soft Tissue Seformation for Laparoscopic surgery Simulation. *Med Image Anal*, 4(1):57–66, 2000. 5.1, 8.1

[175] Y. F. Tai and P. Piccini. Applications of Positron Emission Tomography (PET) in Neurology. *J Neurol Neurosurg Psychiatry*, 75(5):669–676, 2004. 2.2

[176] C. Tanner, A. Degenhard, J. A. Schnabel, C. Hayes, L. I. Sonoda, M. O. Leach, D. R. Hose, D. L. G. Hill, and D. J. Hawkes. A Comparison of Biomechanical Breast Models: a Case Study. In *Proc. SPIE Medical Imaging 2002*, volume 4683, pages 1807–1818, 2002. 2.2, 2.4.1.3, 6.5, 7.1

[177] D. G. Taylor and M. C. Bushell. The Spatial Mapping of Translational Diffusion Coefficients by the NMR Imaging Technique. *Phys Med Biol*, 30(4):345–349, 1985. 9.1

[178] Z. Taylor and K. Miller. Reassessment of Brain Elasticity for Analysis of Biomechanisms of Hydrocephalus. *J Biomech*, 37(8):1263–1269, 2004. 5.2.3

[179] J. Thirion. Non–rigid matching using demons. In *Proc. Conf. Comp. Vis. and Pat. Recog. (CVPR)*, 1996. 2.3.1.1

[180] J. P. Thirion. Image Matching as a Diffusion Process: an Analogy with Maxwell's Demons. *Med Image Anal*, 2(3):243–260, 1998. 2.4.1.3

[181] P. M. Thompson and A. W. Toga. A Surface-Based Technique for Warping Three-Dimensional Images of the Brain. *IEEE Trans Med Imaging*, 15(4):402–417, 1996. 2.4.1.2

[182] T. M. Tucker and T. R. Kurfess. Newton Methods for Parametric Surface Registration. Part I. Theory. *Comput Aided Design*, 35(1):107–114, 2004. 2.3.1.2

[183] S. Tungjitkusolmun, E. J. Woo, H. Cao, J. Z. Tsai, V. R. Vorperian, and J. G. Webster. Thermal-Electrical Finite Element Modelling for Radio Frequency Cardiac Ablation: Effects of Changes in Myocardial Properties. *Med Biol Eng Comput*, 38(5):562–568, 2000. 5.1

[184] M. van Herk, J. C. de Munck, J. V. Lebesque, S. Muller, C. Rasch, and A. Touw. Automatic Registration of Pelvic Computed Tomography Data and Magnetic Resonance Scans Including a Full Circle Method for Quantitative Accuracy Evaluation. *Med Phys*, 25(10):2054–2067, 1998. 2.3.1.2

[185] B. C. Vemuri, S. Huang, S. Sartaj, C. M. Leonard, C. Mohr, R. Gilmore, and J. Fitzsimmons. An Efficient Motion Estimator with Application to Medical Image Registration. *Med Image Anal*, 2(1):79–98, 1998. 2.4.1.3

[186] P. Viola and W. Wells. Alignment by Maximization of Mutual Information. In *Proc. International Conference on Computer Vision 95*, pages 16–23, Cambridge, MA, 1995. 2.3.1.3

[187] Y. Wang, B. S. Peterson, and L. H. Staib. 3D Brain Surface Matching based on Geodesics and Local Geometry. *Comput Vis Image Und*, 89(2-3):252–271, 2003. 2.4.1.2

[188] Y. Wang and L. Staib. Elastic Model Based Non-Rigid Registration Incorporating Statistical Shape Information. In *Proc. MICCAI*, Lect Notes Comp Sc, pages 1162–1173. Springer, 1998. 5.2.1

[189] S. Warfield, M. Ferrant, X. Gallez, A. Nabavi, F. Jolesz, and R. Kikinis. Real-Time Biomechanical Simulation of Volumetric Brain Deformation for Image Guided Neurosurgery. In *Proc. ACM/IEEE Supercomputing*, volume 23 of *CD-ROM*, pages 1–16. IEEE Computer Society, 2000. 8.1

[190] S. Warfield, F. Talos, A. Tei, A. Bharatha, A. Nabavi, M. Ferrant, P. Black, F. Jolesz, and R. Kikinis. Real-Time Registration of Volumetric Brain MRI by Biomechanical Simulation of Deformation during Image Guided Neurosurgery. *Comput Visual Sci*, 5:3–11, 2002. 2.2, 2.4, 4

[191] M. Wawro and M. Fathi-Torbaghan. A Parallel Framework for the FE-based Simulation of Knee Joint Motio. *IEEE Trans Biomed Eng*, 51(8):1490–1494, 2004. 8.1

[192] W. Wells, P. Viola, H. Atsumi, S. Nakajima, and R. Kikinis. Multi-modal Volume Registration by Maximization of Mutual Information. *Med Image Anal*, 1(1):35–51, 1997. 3.2

[193] J. West, J.M. Fitzpatrick, M.Y. Wang, B.M. Dawant, C. R. Maurer, R.M. Kessler, R.J. Maciunas, C. Barillot, D. Lemoine, A. Collignon, F. Maes, P. Suetens, D. Vandermeulen, P. A. van den Elsen, S. Napel, T. S. Sumanaweera, B. Harkness, P. F. Hemler, D. L. Hill, D. J. Hawkes, C. Studholme, J. B. Maintz, M. A. Viergever, G. Malandain, X. Pennec, M. E. Noz, G. Q. Maguire, M. Pollack, C. A. Pelizzari, R. A. Robb, D. Hanson, and R. P. Woods. Comparison and Evaluation of Retrospective Intermodality

Brain Image Registration Techniques. *J Comput Assist Tomogr*, 21(4):554–566, 1997. 2.3.1.3

[194] R. Westermann and C. Rezk-Salama. Real-Time Volume Deformations. *Comput Graph Forum*, 20(3):443–451, 2001. 4.1

[195] C.-F. Westin, S.E. Maier, H. Mamata, A. Nabavi, F.A. Jolesz, and R. Kikinis. Processing and Visualization for Diffusion Tensor MRI. *Med Image Anal*, 6(2):93–108, 2002. 9.2.2

[196] R. Woods, J. Mazziotta, and S. Cherry. MRI-PET Registration with Automated Algorithm. *J Comput Assist Tomogr*, 17(4):536–546, 1993. 2.3.1.3

[197] R. P. Woods, S. R. Cherry, and J. C. Mazziotta. Rapid Automated Algorithm for Aligning and Reslicing PET Images. *J Comput Assist Tomogr*, 16(4):620–633, 1992. 2.3.1.3

[198] R. P. Woods, S. T. Grafton, C. J. Holmes, S. R. Cherry, and J. C. Mazziotta. Automated Image Registration: I. General Methods and Intrasubject, Intramodality Validation. *J Comput Assist Tomogr*, 22(1):139–152, 1998. 2.3.1.3

[199] T. H. Wu, J. K. Wang, J. J. Lee, R. S. Liu, and W. Y. Guo. An Imaging Co-Registration System using Novel Non-Invasive and Non-Radioactive External Markers. *Eur J Nucl Med Mol Imaging*, 30(6):812–818, 2003. 2.3.1.1

[200] X. Yuan, J. Zhang, and B. P. Buckles. Evolution Strategies based Image Registration via Feature Matching. *Med Image Anal*, 2004. in press. 2.3.1.3

Part VII

German Part

Kurzfassung

In den letzten Jahren hat sich die Qualität der chirurgischen Verfahren hauptsächlich infolge der Anwendung von Navigationssystemen deutlich verbessert. Insbesondere neurochirurgische Interventionen haben enorm von der Möglichkeit profitiert, chirurgische Werkzeuge gleichzeitig im physikalischen Raum, der mit dem Operationssaal verbunden ist und im virtuellen Raum der volumetrischen Bilder, die Einblick in den Körper des Patienten geben, zu verfolgen. Navigationssysteme, kombiniert mit intraoperativer Bildgebung, erlauben eine präzisere Resektion des kranken Gewebes und verringern damit postoperative traumatologische Defizite. Gleichzeitig gibt es einen erheblichen Bedarf an neue Methoden, die eine Korrespondenz zwischen zeitlich versetzt aufgenommenen Daten, wie prä- und intraoperativen Bildern, herstellen. Das wird chirurgische Eingriffe durch eine umfangreichere Integration präoperativer Planungsdaten in die intraoperative Umgebung weiter verbessern.

In dieser Doktorarbeit werden verschiedene Algorithmen zur Registrierung von medizinischen tomographischen Bilddaten präsentiert. Die eingeführten Methoden beziehen sich auf die Anwendung in einer intraoperativen Umgebung. In der Neurochirurgie sind strenge Zeitanforderungen ein extrem wichtiger Faktor. Aus diesem Grund konzentrieren sich die dargestellten Methoden auf Beschleunigungstechniken und gleichzeitig auf hohe Genauigkeit und Zuverlässigkeit. In dieser Hinsicht ist eine starre Registrierungsstrategie, die Grafikhardware zur Beschleunigung der Interpolationsoperationen ausnutzt, optimiert und durch eine hierarchische Technik erweitert worden. Diese Implementierung kann für eine Ausgangsregistrierung der prä- und intraoperativen Daten verwendet werden.

Jedoch erfordert die Deformation des Gehirngewebes, die nach der Craniotomie auftritt (im Englischen als *Brain Shift* bekannt), eine nichtlineare Registrierung der Daten. Dies wird mit einer volumetrischen Bilddeformation erzielt, die auf einem flexiblen Hexaedermodell basiert. Dieses Modell wurde entworfen, um eine Beschleunigung der Registrierung mit Grafikkarten zu erlauben. Um die Qualität des Algorithmus zu verbessern, ist eine Erweiterung vorgeschlagen worden, bei der das Konzept der Freiformdeformationen (FFD) zur Realisierung einer nichtlinearen Registrierung angewendet wurde. Das entwickelte System hat sich in Experimenten mit realen klinischen Daten bewährt, die vor und während der Operation am offenen Schädel gewonnen wurden. Das System ist auch erfolgreich angewendet

worden, um die Verzerrungen in Diffusions-Tensor (DT) Daten zu kompensieren. Damit wurde die Flexibilität und Stärke der Techniken demonstriert.

Ferner präsentiert diese Arbeit eine grundlegende Simulationsumgebung zur physikalisch basierten Modellierung intraoperativer Gehirndeformationen. In diesem Konzept wird das Gehirn als ein linear elastisches Medium beschrieben, das mit einer viskosen Flüssigkeit gesättigt ist. Die Implementierung des Systems ermöglicht eine Vorhersage des Umfanges und der Richtung des Brain Shifts. Damit ist eine bessere Chirurgieplanung möglich, wobei die Bestimmung der korrekten elastischen Parameter für das Hirngewebe ein wichtiges Problem ist. Dies ist mit einem neuen Ansatz gelöst worden, der Simulation und Registrierung kombiniert. In einem automatischen Optimierungsverfahren wurden die intraoperative Daten mit dem Bild verglichen, das nach der Simulation rekonstruiert wurde. Auf diese Weise wurden die besten elastischen Konstanten gefunden. Eine signifikante Reduktion der Rechenzeit wurde durch eine parallele Implementierung des Simulationsmodells erzielt. Mit diesem Ansatz konnten auch sehr große Datenmengen, die nicht in den Speicher einer Einprozessor-Maschine passten, erfolgreich bearbeitet werden.

Inhaltsverzeichnis

Kurzfassung iii

Inhaltsverzeichnis ix

Abbildungsverzeichnis x

Tabellenverzeichnis xi

Quellcodes xiii

Danksagung xv

I Einleitung 2

1 Einleitung 3
1.1 Wissenschaftliche Beiträge 5
1.2 Gliederung der Arbeit 8

II Registrierung 12

2 Überblick über Bildregistrierung 13
2.1 Mathematischer Hintergrund 15
2.2 Anwendungen 16
2.3 Rigide Registrierung 18
2.3.1 Verwandte Arbeiten 19
2.4 Nichtlineare Registrierung 23
2.4.1 Verwandte Arbeiten 24

3 Hardwarebasierte rigide Registrierung 29
3.1 Grundlagen der Grafikhardware 30
3.2 Ähnlichkeitsmaße 33

3.3 Registrierung 35
3.3.1 3D Texturen 36
3.3.2 Reformatierungsalgorithmus 36
3.3.3 Auflösungspyramide 39
3.4 Ergebnisse 42
3.5 Diskussion 46
3.6 Zusammenfassung 49

4 Hardwarebasierte nichtlineare Registrierung 51
4.1 Verwandte Arbeiten 52
4.2 3D stückweise lineares Deformationsmodell 53
4.2.1 Deformation im Texturraum 53
4.2.2 Deformation im Geometrieraum 55
4.2.3 Hierarchische Octree-Struktur 55
4.2.4 Implementierungsdetails 57
4.2.5 Registrierungsalgorithmus 58
4.2.6 Optimierung 60
4.2.7 Ansatz mit Auflösungspyramide 61
4.2.8 Ergebnisse 61
4.2.9 Diskussion 67
4.3 3D Free-Form Deformation 68
4.3.1 Deformationsmodell 68
4.3.2 Hardwarebeschleunigte Freiformdeformation 69
4.3.3 Registrierung 70
4.3.4 Ergebnisse 72
4.3.5 Diskussion 74
4.4 Zusammenfassung 76

III Simulation 80

5 Überblick über Weichgewebesimulation 81
5.1 Klinische Anwendungen 82
5.2 Überblick über Simulation von Brain Shift 84
5.2.1 Linear elastische Modelle 85
5.2.2 Viskoelastische Modelle 86
5.2.3 Biphasische Theorie und poroelastische Modelle 87
5.2.4 Zusammenfassung 87

6 Berechnungsmodell von Brain Shift 89
6.1 Theorie der finiten Elemente 90
6.2 Basisgleichungen 93
6.2.1 Spannungsverteilung 93

6.2.2 Darcy-Gesetz . . . 96
6.2.3 Randbedingungen . . . 97
6.3 Finite Elemente Diskretisierung . . . 98
6.4 Geometriegenerierung . . . 99
6.4.1 Gehirnsegmentierung aus MR-Daten . . . 99
6.4.2 Extraktion von Dreiecksoberflächen . . . 99
6.4.3 Tetraedrisierung . . . 102
6.5 Randbedingungen . . . 104
6.6 Volumenrekonstruktion . . . 104
6.7 Ergebnisse . . . 106
6.7.1 Synthetischer Datensatz . . . 106
6.7.2 Klinische Experimente . . . 107
6.8 Diskussion . . . 110
6.9 Zusammenfassung . . . 112

7 Bestimmung der mechanischen Eigenschaften des Gehirns 113
7.1 Verwandte Arbeiten . . . 114
7.2 Methode . . . 115
7.2.1 Ähnlichkeitsfunktion . . . 115
7.2.2 Biomechanisches Model und Volumenrekonstruktion . . . 116
7.2.3 Minimierungsumgebung . . . 116
7.3 Ergebnisse . . . 116
7.4 Diskussion . . . 118
7.5 Zusammenfassung . . . 120

8 Hochleistungssimulation 121
8.1 Verwandte Arbeiten . . . 122
8.2 Parallele Softwareumgebung . . . 123
8.2 Gitterpartitionierung . . . 123
8.2.2 Elementeverteilung und -verwaltung . . . 125
8.2.3 Matrixspeicherung . . . 125
8.2.4 Lösung des linearen Gleichungssystems . . . 127
8.3 Ergebnisse . . . 127
8.4 Diskussion . . . 129
8.5 Zusammenfassung . . . 132

IV Anwendung der nichtlinearen Registrierung 134

9 Nichtlineare Korrektur von Diffusions-Tensor-Daten 135
9.1 Einleitung . . . 135
9.2 Algorithmus . . . 137
9.2.1 Bildaufnahme . . . 137

9.2.2 Bahnverfolgung 138
9.2.3 Registrierung 139
9.3 Ergebnisse 140
9.4 Diskussion 141
9.5 Zusammenfassung 142

V Zusammenfassung **144**

10 Zusammenfassung **145**
10.1 Zukünftige Herausforderungen 147

VI Anhang **150**

A Finite Elemente Diskretisierung **151**
A.1 Form der gewichteten Residuen 151
A.2 Diskretisierung 153

Literaturverzeichnis **157**

VII Deutscher Teil **176**

Kurzfassung **177**

Inhaltsverzeichnis **182**

Einleitung **183**
Wissenschaftliche Beiträge 185
Gliederung der Arbeit 189

Zusammenfassung **191**
Zukünftige Herausforderungen 193

Einleitung

Die Ära der medizinischen Bildgebung begann mit der Entdeckung der Röntgenstrahlen Ende des 19. Jahrhunderts. Die erste bildgeführte Chirurgie wurde vom Prof. Cox von der McGill Universität in Montreal durchgeführt, nur sechs Wochen nachdem Wilhelm Röntgen einen einleitenden Bericht über seine Entdeckung veröffentlicht hatte [140]. Prof. Cox erstellte ein radiologisches Bild vom Bein eines jungen Patienten, der angeschossen worden war und entdeckte auf diese Weise eine Gewehrkugel, die in den vorangegangenen chirurgischen Untersuchungen nicht gefunden worden war. Die Röntgenbilder wurden dann manuell an der Position des Patienten ausgerichtet, was dem Chirurgen die erfolgreiche Entfernung der Gewehrkugel ermöglichte. Dieser Eingriff war die erste Anwendung der medizinischen Bildregistrierung.

Seit diesem Meilenstein in der Operationstechnik hat sich die Technologie, die bei bildgeführten Operationen verwendet wird, ständig verbessert. Die Einführung der (Röntgen-)Computertomographie (CT), des Ultraschalls (US), der Positronen-Emissions-Tomographie (PET) und der Kernspintomographie (MRI: *magnetic resonance imaging*) revolutionierte die traditionellen chirurgischen Techniken und stellte hochwertige dreidimensionale (3D) Bilder bereit, die Einblick in den Körper des Patienten ermöglichen. Diese Bilder können durch ein Computerprogramm manipuliert und angepasst werden, um zusätzliche diagnostische Informationen in den Operationssaal zu bringen. Ferner brachte die Erfindung der Navigationshardware die Möglichkeit, chirurgische Werkzeuge während der Intervention zu verfolgen. Das ermöglichte simultane Navigation im physikalischen Raum, der mit dem Operationssaal verbunden ist, und im virtuellen Bildraum. Besonders in den letzten Jahren ließ sich eine sehr schnelle Entwicklung der Navigationstechniken beobachten [145]. Diese Fortschritte hängen mit einer enormen Entwicklung der Prozessor- und Grafikhardwaretechnologie zusammen, die Manipulationen und Echtzeitvisualisierung der 3D Bilder ermöglicht.

Heutzutage wird Bildführung für Diagnose, Therapieplanung und Chirurgie bei vielen Krankheiten verwendet, z.B. bei Brustbiopsie und -tumorresektion, Kältechirurgie[1] der Prostata und Brachytherapie[2], Gesichts-

[1] Kältechirurgie ist eine Methode bei der das Gewebe (z.B. Krebstumor) eingefroren wird, um es zu zerstören.

[2] Strahlentherapie auf kleine Distanz.

und Schädelkorrekturen, spinalen Verfahren einschließlich Strahlentherapie, Nachbehandlungsüberwachung von Karotisatherosklerose[3], renaler[4] Tumorentfernung und bei der Leberchirurgieplanung.

Der Bereich der Medizin, der vermutlich am meisten von den Bildführungstechniken profitiert hat, ist die Neurochirurgie. Einerseits erfordert Neurochirurgie höchste Präzision bei der Planung des Zugangs zum Zielbereich und der optimalen Positionierung medizinischer Instrumente. Andererseits ist Genauigkeit essentiell, um eine Schädigung kritischer Strukturen zu verhindern. Es ist auch bei der Resektion von krankem Gewebe entscheidend, da dies häufig von den umgebenden gesunden Strukturen visuell nicht zu unterscheiden ist.

Zur Zeit erhältliche kommerzielle Neuronavigationssysteme beruhen stark auf der Annahme, dass die präoperativ aufgenommene Bilder während chirurgischen Interventionen gültig bleiben. Jedoch treten im Weichgewebe starke Veränderungen auf, so dass sich prä- und intraoperativ aufgenommenen Bilder voneinander unterscheiden. Während eines Eingriffs am geöffneten Schädel kann sich das Hirngewebe stark verformen (siehe Abbildung G.1). Dieses Phänomen ist als *Brain Shift* bekannt und ergibt sich auf Grund von neurochirurgischen Vorgängen und mehreren physikalischen Einflüssen, wie der Schwerkraft, der Änderung des auf das Gehirn wirkenden Drucks oder dem Abfluss von Liquor (CSF: *cerebrospinal fluid*) [102, 122].

Es gibt nicht-kommerzielle Ansätze, die sich dieser Aufgabe mittels komplizierten mathematischen und numerischen Techniken widmen. Dennoch sind die intraoperativen Anforderungen der Neurochirurgie sehr einschränkend, sowohl im Bezug auf die Präzision von Algorithmen als auch auf die Zeit, die für die Berechnungen zur Verfügung steht. Folglich gibt es einen hohen Bedarf an bessere und schnellere Algorithmen, die auch in der klinischen Praxis angewendet werden können.

In Abbildung G.2 wird der gesamte Informationsfluss einer computergestützten Methode, die den Chirurgieprozess unterstützt, dargestellt. Dabei werden prä- und intraoperative Bilddaten für die Erzeugung der Modelle und für die Therapieplanung verwendet. Basierend auf den präoperativen Bilddaten wird ein patientenspezifisches Modell und ein chirurgischer Plan entwickelt. Während eines Eingriffs werden diese Informationen mit Hilfe eines Navigationssystems in die operative Umgebung transferiert.

Jedoch müssen das erzeugte patientenspezifische Modell und der chirurgische Plan intraoperativ aktualisiert werden. Eine Möglichkeit dafür ist die intraoperative Bildgebung, mit der die auftretenden Gehirnverschiebungen aufgenommen werden können. Zusätzlich dienen die intraoperativen Bilddaten als Grundlage für eine *Registrierung*, die die Integration der ausschließlich präoperativ vorhandenen Informationen in das Neuronavigationssystem ermöglicht. Alternativ kann eine biomechanische *Simulation* für die Vorhersage der auftretenden Deformation eingesetzt

[3]Verkalkung der internen Karotisarterie, die zu einem Schlaganfall führen kann.
[4]Die Bezeichnung „renal“ bezieht sich auf Nieren.

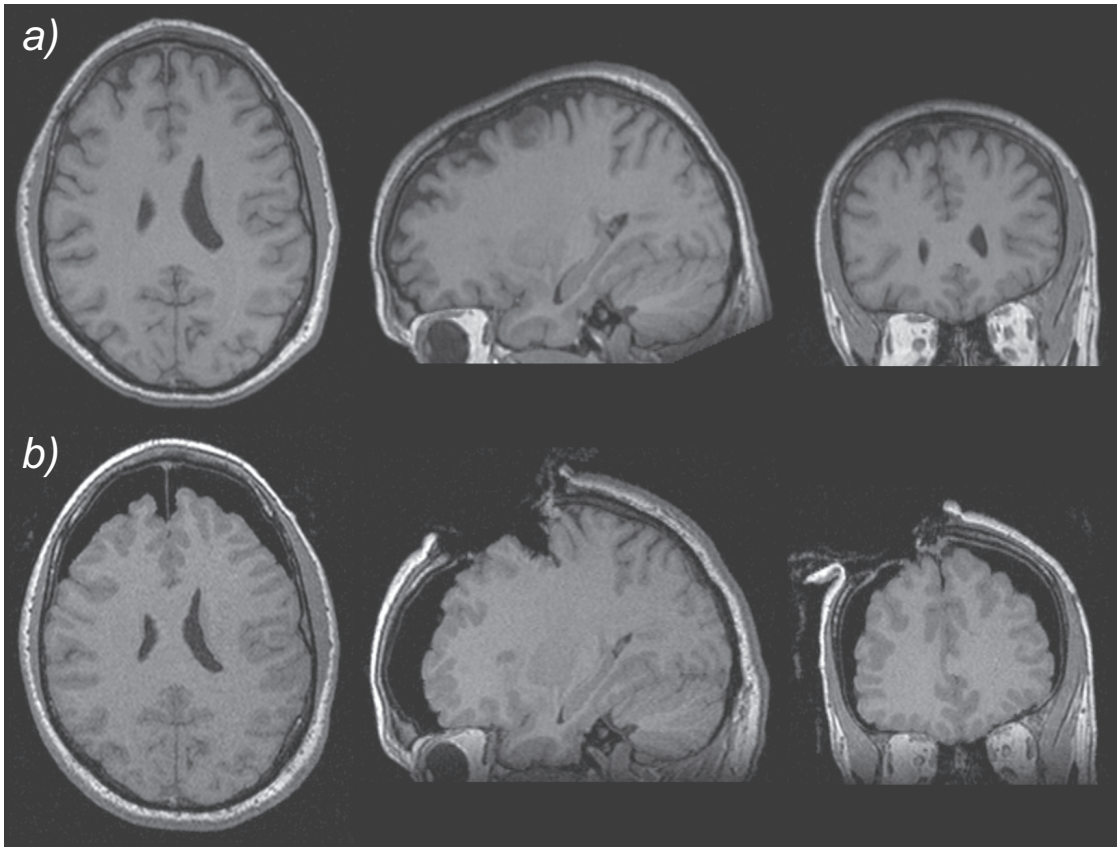

Abbildung G.1: Präoperative *a)* und intraoperative *b)* MR-Bilder vom Kopf zeigen das Brain Shift Phänomen. Eine erhebliche Verschiebung des Weichgewebes ist sichtbar

werden, wenn keine intraoperative Bildgebung vorhanden ist. Um die verfügbare Information vollständig zur Verfügung zu stellen und ein Maximum an Genauigkeit und Sicherheit zu erzielen, werden optimalerweise intraoperative Bildgebung, Registrierung und Simulation kombiniert.

Wissenschaftliche Beiträge

Der Hauptfokus dieser Doktorarbeit ist die Entwicklung neuer Methoden, die die Aktualisierung eines patientenspezifischen Modells und eines chirurgischen Planes während eines computergestützten neurochirurgischen Eingriffes unterstützen. In diesem Zusammenhang sind Algorithmen entwickelt worden, die das Brain Shift Phänomen kompensieren. Zum Einen sind lineare und nichtlineare bildbasierte Registrierungsansätze implementiert worden, die die Berechnung der Transformation präoperativer MR-Bilder auf nach dem Brain Shift aufgenommene intraoperative Daten ermöglichen. Zum Anderen ist eine physikalisch basierte Simulation imple-

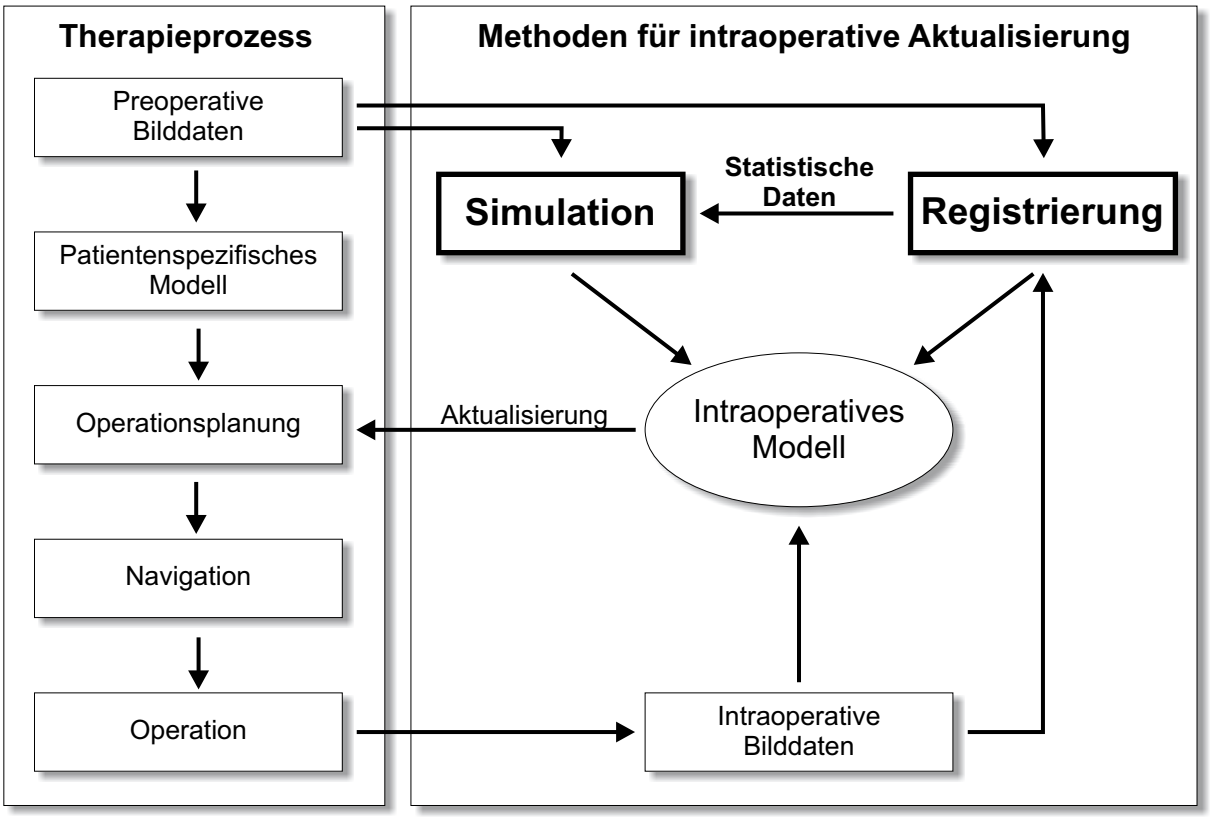

Abbildung G.2: Der Informationsfluss in computergestützter Chirurgie zeigt die Relation zwischen einem konventionellen Therapieprozess und fortgeschrittenen Methoden für intraoperative Aktualisierung des Planungskonzepts

mentiert worden, die die Vorhersage der auftretenden Gehirndeformation erlaubt. Mit diesem System können Umfang und Richtung der intraoperativen Gehirnbewegungen demonstriert und damit die Interventionsplanung unterstützt werden.

Schließlich ist eine Sammlung von Daten angelegt worden, die aus intraoperativen MR-Daten und den korrespondierenden, gemäß Simulationsberechnungen deformierten, präoperativen MR-Bildern besteht. Statistische Informationen, die von dieser Datensammlung abgeleitet wurden, sind in die entwickelte Simulationsumgebung integriert worden, um die Simulation zu optimieren und die Qualität einer intraoperativen Aktualisierung zu verbessern. Die entwickelten Registrierungs- und Simulationsstrategien zeichnen sich durch beträchtlich reduzierte Rechenzeiten aus. Dies wurde mit der Entwicklung neuer Techniken erzielt, die moderne PC Grafikkarten und Parallelisierungsstrategien ausnutzen, und ist eine wesentliche Voraussetzung für den intraoperativen Einsatz eines bildgeführten Systems.

Die vorliegende Doktorarbeit beinhaltet folgende Hauptbeiträge:

Hardwarebasierte starre Registrierung

Starre Registrierung ist der grundlegende Schritt, der für eine Aktualisierung präoperativer Bilder benötigt wird. Die daraus hervorgehende globale, lineare Transformation ist eine wesentliche Voraussetzung für eine nachfolgende lokale, nichtlineare Registrierung. Zusätzlich ist die Qualität der starren Registrierung entscheidend für das Setzen passender Randbedingungen für die statistische Methode, die Registrierung und Simulation kombiniert. Zu diesem Zweck wurde ein Ansatz für die starre 3D/3D Registrierung von medizinischen Bildern, der für spezialisierte Hardwaresysteme eingeführt worden ist [72], wesentlich für Standard PC Architekturen optimiert und erweitert. Dies führte zu einer hierarchischen Umgebung, die eine automatische Registrierung ermöglicht. Um die Berechnungen zu beschleunigen, wurden auftretende trilineare Interpolationen durch Grafikhardware übernommen. Das implementierte System ist in [73, 74] vorgestellt worden.

Hardwarebasierte nichtlineare Registrierung

Nichtlineare Registrierungsalgorithmen sind durch hohe Rechenkosten gekennzeichnet. Für klinische Anwendungen sind Beschleunigungstechniken von großem Wert. Um diese Anforderung zu erfüllen, sind schnelle und gleichzeitig genaue Methoden der nichtlinearen Registrierung entwickelt worden.

In diesem Zusammenhang wurden nichtlineare Volumendeformationen mit stückweise linearen Funktionen approximiert. Bei dieser Technik wird ein Objekt in eine Anzahl von volumetrischen Patches unterteilt, deren Form sich leicht ändern lässt. Diese Idee ist auf Grafikkarten transferiert worden, mit denen sich Deformationen modellieren lassen, indem Knoten im 3D Texturraum verschoben werden. Hardwarebasierte trilineare Interpolation beschleunigt den iterativen Registrierungsprozess erheblich. Der Nutzen dieses Ansatzes wurde in [164, 77] nachgewiesen.

Im nächsten Schritt sind 3D Bézier Funktionen in die entwickelte Registrierungsumgebung integriert worden, die weitere Möglichkeiten moderner Graphikkarten ausnutzen. Dies macht den Algorithmus zu einer völlig neuen Methode auf dem Gebiet der Registrierung. Der Ansatz ist in [157, 165] eingeführt worden.

Biomechanisches Modell zur Simulation von Brain Shift

Die entwickelte Software-Plattform beinhaltet alle Schritte, die für die Durchführung einer Finiten Elemente Simulation erforderlich sind. Beginnend mit der Segmentierung des Gehirns aus den MR-Daten wird

eine Dreiecksoberfläche errechnet, so dass die Regularität des Gitters sichergestellt wird. Anschließend wird ein Tetraedergitter erzeugt. Diese Technik der Volumendiskretisierung ist entscheidend für die numerische Stabilität des Simulationsprozesses. Ferner werden physikalische Gleichungen definiert, die die Gehirndeformation beschreiben. Diese basieren auf dem von Miga *et al.* [107] vorgeschlagenen Modell, das hier wesentlich erweitert wurde. Die Regionen der unterschiedlichen Randbedingungen werden mit Hilfe der Ergebnisse der starren Registrierung von prä- und intraoperativer MR-Aufnahmen festgelegt. Diese Arbeiten wurden in [158, 159] beschrieben.

Gekoppelte Simulations- und Registrierungsumgebung

Um die Deformation des Gehirnweichgewebes zuverlässig vorauszusagen, ist eine elastische Beschreibung erforderlich. Für die Simulation ist es entscheidend, sowohl den Young'schen Elastizitätsmodul als auch die Poisson-Zahl zu rekonstruieren, da ihre Kombination für das Modell essentiell ist. Zu diesem Zweck ist eine neue Methode eingeführt worden, mit der beide elastischen Modulen iterativ errechnet werden. Der Algorithmus basiert auf Registrierung der intraoperativen MR-Daten mit den korrespondierenden Ergebnissen einer biomechanischen Simulation. Die Ideen dieser Arbeit sind in [161, 160, 163] erschienen.

Effiziente Implementierung der Simulation

Eine wichtige Anforderung an klinische Anwendbarkeit ist zeitliche Effizienz. Aus diesem Grund ist eine parallele Verarbeitung implementiert worden, die diesen Anforderungen gerecht wird. Infolge der dezentralen Verarbeitung der Daten können Berechnungen auf einer sehr hoch aufgelösten Geometrie effizient durchgeführt werden. Damit können Gehirnstrukturen detailliert modelliert werden, während die Antwort des Systems mit der Rate zur Verfügung gestellt wird, die für den Gebrauch während neurochirurgischer Eingriffe akzeptabel ist. Eine Darstellung dieses Ansatzes findet man in [162].

Nichtlineare Korrektur von Diffusions-Tensor-Bildern

Die Diffusions-Tensor (DT) Bildgebung ist vor kurzer Zeit eingeführt worden und liefert wichtige Informationen über neurologische Verbindungen im Gehirn. Jedoch stehen diese Daten nur mit großen Verzerrungen zur Verfügung. Dies entsteht durch das zugrunde liegende schnelle MR-Bildprotokoll. Um die Artefakte in den DT-Daten zu korrigieren, ist der entwickelte Registrierungsansatz, der auf 3D Bézier Funktionen basiert, angewendet worden [105]. Auf diese Weise wird eine leistungsstarke Unterstützung angeboten, die anstrebt, postoperative neurologische Defizite zu minimieren.

Gliederung der Arbeit

Diese Doktorarbeit besteht aus fünf Hauptteilen. Im ersten Teil werden die Hauptkonzepte der präsentierten Forschung und die Motivation dafür gegeben. Der eigene Beitrag dieser Dissertation wird hier hervorgehoben und die Gliederung der Arbeit wird dargestellt.

Im zweiten Teil werden Registrierungsalgorithmen angesprochen. Kapitel 2 enthält die grundlegenden Definitionen und Begriffe, die in Zusammenhang mit Registrierung gebräuchlich sind. Verschiedene bestehende lineare und nichtlineare Ansätze zur Registrierung medizinischer Bilder werden präsentiert und diskutiert. Kapitel 3 fängt mit einer Einleitung zu Grafikhardware an. Dann werden die implementierte hardwarebasierte starre 3D/3D Registrierungsmethode und die Ergebnisse ihrer Anwendung auf medizinischen Bilder besprochen. In Kapitel 4 wird das Konzept der stückweise linearen Registrierung und seine Implementierung mit Hilfe von Grafikhardware beschrieben. Darüber hinaus wird eine Erweiterung dieses Ansatzes, die auf 3D Bézier Funktionen basiert, vorgestellt und ausgewertet.

Der dritte Teil dieser Dissertation beschäftigt sich mit der biomechanischen Simulation von Weichgewebedeformation. Kapitel 5 gibt einen Überblick über die Arbeiten, die in Verbindung mit der Modellierung von Weichgewebedeformationen stehen. Kapitel 6 fängt mit einer Einleitung zu ausgewählten Techniken der Finite Elemente Simulation an. Dabei werden grundlegende Begriffe definiert und die relevanten mathematischen Eigenschaften erklärt. Dann wird die konstruierte Umgebung für die numerische Simulation der Gehirnverformung ausführlich beschrieben. Es werden verschiedene Elemente des implementierten Systems behandelt. Beginnend mit der Geometrieerzeugung werden die numerische Lösung der zugrunde liegenden partiellen Differentialgleichungen, korrekte Randbedingungen und schließlich die Visualisierung und Interpretation der Ergebnisse angesprochen. In Kapitel 7 wird ein System für die Bestimmung der mechanischen Eigenschaften des Hirngewebes präsentiert. In diesem ZUsammenhang wird die Idee der Kopplung von Simulation und Registrierung für die Berechnung der elastischen Modulen des Gehirns eingeführt, und die Ergebnisse der Experimente werden besprochen. Schließlich stellt Kapitel 8 eine parallele Umgebung zur Durchführung einer Hochleistungssimulation vor. Die Implementierungsdetails werden beschrieben und die Rechenzeiten präsentiert.

Teil vier zeigt eine weitere mögliche Anwendung der hardwarebasierten nichtlinearen Deformation zur Korrektur von DT-Daten (siehe Kapitel 9). Die Position und die Form der Faserbahnen vor und nach der Korrektur werden im Bezug auf die physikalischen Parameter der Messung angesprochen.

Der fünfte und abschließende Teil dieser Doktorarbeit beinhaltet Kapitel 10, das die dargestellte Arbeit zusammenfasst und einen Ausblick auf zukünftig mögliche Erweiterungen der eingeführten Techniken vorstellt. Zusätzlich ist Anhang A enthalten, der einige ausführliche Herleitungen umfasst, die sich auf die präsentierte Simulationsumgebung beziehen.

Zusammenfassung

In den letzten Jahren sind chirurgische Verfahren durch die Erfindung der Navigationshardware revolutioniert worden. Die Anwendung von dieser Hardware hat zur bildgeführten Chirurgie geführt. Diese neue chirurgische Strategie bringt die Möglichkeit, chirurgische Werkzeuge während der Intervention zu verfolgen. Das ermöglicht simultane Navigation im physikalischen Raum, der mit dem Operationssaal verbunden ist und im virtuellen Bildraum. Besonders in der Neurochirurgie hat dies zu einer enormen Verbesserung der Qualität und eine beträchtliche Vereinfachung der chirurgischen Verfahren geführt. Zum Beispiel ist es mit Hilfe der Registrierungstransformation zwischen dem physikalischen und dem Bildraum möglich, Tumorgrenzen in den halbtransparenten Spiegel eines Operationsmikroskops zu projizieren. Das erlaubt es, die Resektion von krankem Gewebe genauer und effektiver durchzuführen.

Da bildgeführte Neurochirurgie stark mit Computertechnologie und der Verfügbarkeit mathematischer Modelle zusammenhängt, gibt es noch viel Bedarf an Lösungen und offene Herausforderungen. Eine von ihnen ist die Entwicklung von schnellen und gleichzeitig zuverlässigen Methoden für die Registrierung medizinischer Bilder. In diesem Zusammenhang können Algorithmen der starren Registrierung eingesetzt werden, um die Transformation zwischen prä- und intraoperativen MR-Daten zu bestimmen. Diese Transformation kann dann während einer bildgeführten Operation verwendet werden, um eine präoperative Planung in die intraoperative Umgebung zu transferieren oder eine Aktualisierung der Bilddaten im Navigationssystem durchzuführen. Ferner gibt es eine Notwendigkeit für die Entwicklung rechnerisch effizienter und gleichzeitig genauer nichtlinearen Registrierungstechniken, um die in hohem Grade komplexen Deformationen des Hirngewebes zu berechnen, die während einer Operation am offenen Schädel auftreten. Eine andere Herausforderung ist eine zuverlässige Simulation des Brain Shifts zur Planung eines Eingriffs und auch zur Durchführung einer Aktualisierung in einer operativen Umgebung, in der keine intraoperative Bildgebung vorhanden ist.

In dieser Doktorarbeit sind neue Methoden der linearen und nichtlinearen Registrierung medizinischer Bilddaten und einer physikalisch basierten Simulation intraoperativer Gehirndeformationen entwickelt worden. Erstens ist ein hardwarebasierter Algorithmus für dreidimensionale starre Registrierung implementiert worden. Weil die intraoperative Umgebung durch extrem hohe Anforderungen an

die Rechenzeiten gekennzeichnet ist, wurde moderne Grafikhardware für zeitintensive Berechnungen eingesetzt worden. Dabei werden keine externen Merkmale wie künstliche Marker gebraucht, da der Algorithmus nur auf den Voxelintensitäten medizinischer Bilder arbeitet. Verschiedene Experimente auf der Basis von MR-Daten, die vor und während mehrerer Operationen aufgenommen wurden, haben den Nutzen des Algorithmus für neurochirurgische Verfahren bewiesen und zeigten, mit Bezug auf seine Leistung, seine Eignung für eine intraoperative Umgebung.

Ferner sind neue Methoden der nichtlinearen Registrierung vorgeschlagen und implementiert worden. In diesem Zusammenhang sind ein Ansatz, der auf einer volumetrischen Hexaederstruktur basiert, und ein Algorithmus, der 3D Bézier Funktionen verwendet, eingeführt und umfassend für neurochirurgische Anwendungen ausgewertet worden. Bei beiden Methoden sind die Interpolationsmöglichkeiten von 3D Texturhardware zur Beschleunigung der Berechnungen ausgenutzt worden. In Experimenten, die mit realen Daten durchgeführt worden sind, zeichnete sich der erste Ansatz durch eine leicht bessere Recheneffizienz aus. Allerdings wurden bei den zweiten Technik qualitativ bessere Resultate nachgewiesen. Generell waren beide Methoden in den meisten Fällen in der Lage, große nichtlineare Gehirndeformationen zu kompensieren und gleichzeitig zufrieden stellende Lösungen innerhalb angemessener Zeit zu liefern.

Um intraoperative Gehirndeformationen vorauszusagen, ist eine Simulationsumgebung eingeführt worden, in der das auftretende Phänomen mittels physikalisch basierter Gleichungen modelliert wurde. Ausgehend von präoperativen MR-Daten wurde eine passende 3D Geometrie mit einer gesonderten Methode erzeugt, die die Regularität des konstruierten Gitter sicherstellte. Dann wurden die Basisgleichungen mit einer Finite Elemente Technik gelöst. Anschließend ist ein dreidimensionales Bild gemäß dem resultierenden Verschiebungsvektorfeld rekonstruiert worden. Dieses Bild kann in ein Navigationssystem integriert werden und dadurch Informationen über die Richtung und die Größe der Gewebeverschiebung zur Verfügung stellen. Ferner ist eine Hochleistungsimplementierung zur Beschleunigung der Berechnungen eingeführt worden. In dieser Erweiterung zum Simulationssystem wurde die zugrunde liegende Geometrie in eine Reihe topologisch optimaler Komponenten partitioniert und die Berechnungen wurden auf mehrere Prozessoren verteilt, die zu einem PC Cluster gruppiert waren. Diese Technik führte zu einer drastischen Reduktion der Rechenzeiten, während die Genauigkeit der Lösung erhalten blieb. Außerdem ist das vorgeschlagene System weiter skalierbar und bietet Potential für viele andere Anwendungen.

Schließlich sind die Methoden, die für die Registrierung entwickelt wurden, mit der Simulationsumgebung gekoppelt worden, um die optimalen elastischen Eigenschaften von Hirngewebe für die Anwendung in einem physikalisch basierten Modell zu bestimmen. Eine statistische Analyse der Ergebnisse, die in einer Reihe von Experimenten mit prä- und intraoperativen MR-Bildern vom Kopf erzielt wurden, erlaubte es, die physikalischen Konstanten zu berechnen, die das Hirngewebe beschreiben. Dies ermöglicht eine realistische Vorhersage des Brain Shifts vor einer

Operation (z.B. zu Planungszwecken) unter der Voraussetzung, dass die Position der Schädelöffnung bekannt ist.

Alle Algorithmen, die hier präsentiert worden sind, wurden entworfen, um bildgeführte neurochirurgische Interventionen zu unterstützen. Dies stellt einen beträchtlichen Beitrag zum schwierigen Problem der Kompensation und zur Analyse intraoperativer Gehirndeformationen dar. Eine weitere Anwendung der entwickelten Techniken war der Ausgleich von Verzerrungen in Diffusions-Tensor (DT) Bilddaten. Dafür wurden die Bilder, die die funktionellen Informationen repräsentieren, mit den anatomischen MR-Daten registriert. Die resultierende Transformation ist dann verwendet worden, um die Position der Faserbahnen zu korrigieren, die aus den Gradientenbildern errechnet worden waren. Obwohl der nichtlineare Algorithmus ursprünglich für Registrierung von prä- und intraoperativen MR-Daten entwickelt worden war ist er auch in diesem Zusammenhang erfolgreich angewendet worden, was die Stärke und die Flexibilität der präsentierten Systeme unterstreicht.

Zukünftige Herausforderungen

Der schnelle Fortschritt in der Computertechnologie und das enorme Wachstum des medizinischen Wissens in den letzten Jahren hat den Bedarf an der Einführung neuer und verbesserter computergestützter Interventionen und medizinischer Bildverarbeitung erhöht. Die Aktivitäten in entsprechenden Forschungsfeldern deuten darauf hin, dass bildgeführte Neurochirurgie immer wichtiger wird. Daraus folgt eine zunehmende Notwendigkeit für bessere und schnellere Techniken, die die Diagnose und Therapie unterstützen. Dabei ist die Echtzeitanforderung, die durch intraoperative Beschränkungen auferlegt wird, extrem wichtig.

Die Methoden und ihre Anwendungen, die innerhalb dieser Arbeit eingeführt worden sind, stellen eine flexible und leistungsfähige Grundlage für die Entwicklung weiterer Algorithmen und für ihre Implementierung in intraoperativen Unterstützungssystemen zur Verfügung. Zukünftige Herausforderungen liegen in einer weiteren Reduktion der Rechenzeiten, insbesondere bei der nichtlinearen Registrierung. Obwohl die in der Arbeit präsentierten Methoden eine komplette dreistufige nichtlineare Registrierung in weniger als einer Stunde ermöglichen, sind Bearbeitungszeiten unter 10 Minuten notwendig, um diese Technik intraoperativ anwendbar zu machen.

Grafikhardware zeigt ein signifikantes Potential, das zu diesem Zweck ausgenutzt werden kann. Die neue Generation von programmierbaren Grafikkarten erlaubt eine weitere Beschleunigung der Berechnungen bei der nichtlinearen Registrierung. Die in der Arbeit eingeführten Methoden sind eine umfassende Grundlage für die Anwendung von Grafikhardware für medizinische Bildverarbeitung und versprechen weitere Forschungsfortschritte in dieser Richtung.

Abgesehen von den Berechnungen gewinnen Visualisierung und Strategien der

erweiterten Realität in zunehmendem Maße an Bedeutung. In den momentan vorhandenen kommerziellen Systemen wird der Chirurg durch visuelle Integration präoperativer Planungsdaten in optische Systeme unterstützt. Diese Informationen sind jedoch nicht dreidimensional. Die Integration registrierter und simulierter Bilddaten in ein 3D Bildüberlagungssystem, das den realen und virtuellen Raum fusioniert, würde die Qualität chirurgischer Interventionen erheblich verbessern.

LEBENSLAUF GRZEGORZ SOZA

29.04.1976	geboren in Kraków, Polen
09/1983 – 06/1991	Grundschule Zabierzów, Polen
09/1991 – 06/1995	Jan III. Sobieski Gymnasium, Kraków, Polen
21.05.1995	Abitur
10/1995 – 04/2000	Studium der Informatik an der Jagiellonen Universität Kraków, Polen
10/1998 – 07/1999	Studium der Informatik als Stipendiat der Friedrich-Alexander Universität Erlangen-Nürnberg
20.04.2000	Diplom in Informatik
07/2000 – 06/2005	Wissenschaftlicher Mitarbeiter am Lehrstuhl für Graphische Datenverarbeitung Institut für Informatik an der Friedrich-Alexander Universität Erlangen-Nürnberg
20.06.2005	Antrag auf Zulassung zur Promotion
27.07.2005	Tag der Promotion