Graz University of Technology

Institute for Computer Graphics and Vision

Master Thesis

Registration of Dental CT Images for Jawbone Implant Resorption Analysis

---

Sasa Grbic

Graz, Austria, September 2008

Thesis supervisors

Prof. Dr. Horst Bischof, Graz University of Technology

Dr. Martin Urschler, Graz University of Technology

Abstract

Conducting medical procedures with sophisticated imaging technologies, such as MRI and CT, nowadays is part of the clinical practice. In order to monitor the progress of treatments one may take several many acquisitions of the same patient over time. However, accurate evaluations can not be established based only on the visual inspection of a diagnostician. Registration algorithms have to be provided to assist the diagnostician during the process of comparing data sets.

In this thesis we give a brief review of intra- and inter-modality rigid and intra-modality nonlinear registration methods focusing on CT image registration of the maxilla (upper jawbone). We further present two own registration methods. The first is designed to align rigid structures (e.g bones) and the second is able to align rigid as well as non rigid structures (e.g. soft tissues). The latter is based on a variational framework and allows discontinuities in the flow field and does not have a bias towards piecewise constant flow fields. We did a Graphics Processing Unit (GPU)-based implementation thereby increasing the execution performance dramatically. We further show experimental evaluations of our algorithms on a number of different data sets including jawbone CT images focusing on the posterior maxilla.

Keywords. medical image analysis, rigid image registration, nonlinear image registration, intensity-based registration, optical flow registration, affine flow registration

Acknowledgments

I want to thank my advisor Dr. Martin Urschler for his insight, enthusiasm and continuous support during this thesis and several projects during my undergraduate studies. I am also grateful to Prof. Horst Bischof who sparked my interest towards computer vision during his lectures. I am grateful for the help provided by many ICG members. Therefore I want to especially mention Werner Trobin, Thomas Pock, Manuel Werlberger and Markus Unger.

I would like to acknowledge the help from Robert Kirmaier and Susanne Platzer from the dental clinic (LKH) Graz by initiating this project and providing us with all the data sets and medical insights we needed.

Finally I want to thank my parents for their unwavering support throughout my undergraduate studies.

Contents

1 Introduction **1**
- 1.1 Medical Motivation . . . 2
- 1.2 Image Registration . . . 3
- 1.3 Medical Imaging . . . 4
  - 1.3.1 Modalities . . . 4
  - 1.3.2 Image Representation . . . 5
- 1.4 Aims of the Thesis . . . 6
- 1.5 Synopsis . . . 7

2 State of the Art **9**
- 2.1 A Formal Definition of Image Registration . . . 10
  - 2.1.1 Definition . . . 10
  - 2.1.2 Translation . . . 11
  - 2.1.3 Rotation . . . 12
  - 2.1.4 Rigid Transformations . . . 13
  - 2.1.5 Scaling . . . 14
  - 2.1.6 Affine Transformations . . . 14
  - 2.1.7 Nonlinear Transformations . . . 15
- 2.2 Rigid Registration . . . 17
  - 2.2.1 Feature Based Methods . . . 17
  - 2.2.2 Intensity-Based Methods . . . 21
- 2.3 Nonlinear Registration Strategies . . . 28
  - 2.3.1 Intensity-Based Methods . . . 28
- 2.4 Multi-Resolution Strategies . . . 32
- 2.5 Maxilla Registration . . . 32

3 An Interactive Rigid Registration Algorithm **35**
- 3.1 Preprocessing . . . 36
- 3.2 Manual Masking . . . 36
- 3.3 Registration . . . 39
  - 3.3.1 Initialization . . . 39

3.3.2 Intensity-Based Registration . . . 41
3.3.3 Quaternions and Versors . . . 41
3.3.4 Versor Optimization . . . 44
3.3.5 Rigid Transform in 3D . . . 47
3.3.6 Interpolation . . . 48
3.3.7 Metric . . . 48
3.4 Implementation . . . 49
3.5 Evaluation Experiments & Results . . . 49
3.5.1 Synthetic Data . . . 50
3.5.2 Real Data . . . 51
3.5.3 Conclusion . . . 52

4 Affine Flow Registration **57**
4.1 Unbiased Second-Order Prior . . . 58
4.2 Optimization solutions . . . 59
4.2.1 Solution of **u** . . . 60
4.2.2 Solution of **v** . . . 61
4.3 Implementation . . . 62
4.3.1 Numerical Scheme . . . 62
4.3.2 Acceleration by Graphics Processing Units . . . 63
4.3.3 Implementation on the Graphics Processing Unit . . . 65
4.4 Evaluation Experiments & Results . . . 66
4.4.1 Performance . . . 66
4.4.2 Real Data . . . 67

5 Conclusion and Outlook **75**

A Acronyms and Symbols **77**

Bibliography **82**

Chapter 1

Introduction

Contents

1.1 Medical Motivation **2**

1.2 Image Registration **3**

1.3 Medical Imaging **4**

1.4 Aims of the Thesis **6**

1.5 Synopsis **7**

During the past three decades remarkable developments have taken place in the field of medical imaging. Many imaging technologies, such as `MRI` and `CT`, provide rich sources of data, representing the physical properties and biological function of tissues. Their spatial resolution has evolved enormously. Not just the resolution is becoming higher, the acquiring process is also taking less time and the quality of the acquired images is also improved.

Nowadays it is common for patients to be imaged several times. Often medical images are used to monitor the progress of a treatment or the growth of a disease leading to several scans of one patient. The physician is often overstrained by the huge amount of presented data. To make a successful interpretation or analysis he has to combine them mentally. If the patients position has changed during these scans it will be a challenging task for him. Especially if the underlying transformation of the observed tissue is non-linear. Registration algorithms should disburden the physician from the combining or fusing process.

1.1 Medical Motivation

Dental implants are a standard method in prosthetic dentistry to support restorations in order to resemble a tooth or group of teeth [34]. A key to implant success is the quantity and quality of the bone where the implant has to be placed. The upper back jaw has traditionally been one of the most difficult areas to successfully place dental implants due to insufficient bone quantity and quality and the close proximity to the sinus. Sinus augmentation can help correct this problem by raising the sinus floor and developing bone for the placement of dental implants. After the augmented material is placed in

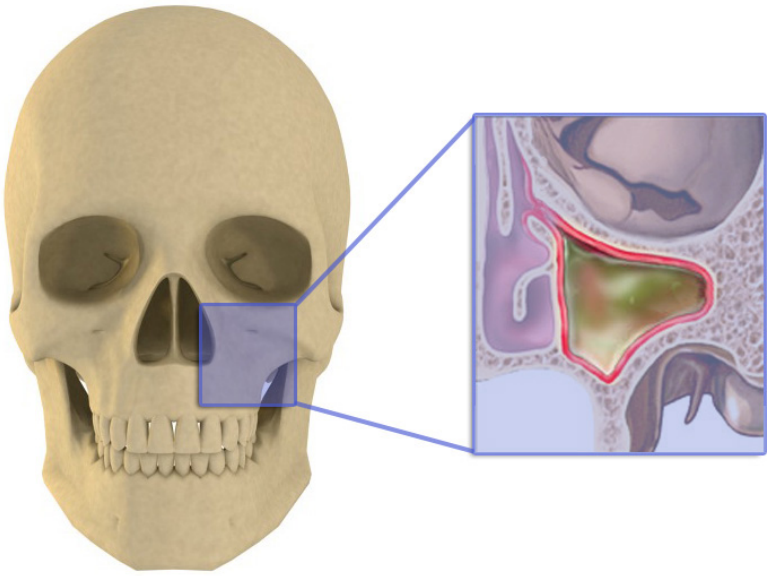


Figure 1.1: The posterior maxilla is the anatomical location where the augmented material is placed.

the **posterior maxilla** (see Figure 1.1) it will be partially absorbed from the surrounding tissue over time. The volume of the augmented material is decreasing in time and its shape is changing. There is a wide range of different materials which are used for augmentation and all of them show different characteristics. A qualitative analysis of these materials is desirable. Even the same material can behave differently on other patients. Therefore monitoring the treatment progress of a single patient will be also of great benefit. During the treatment volume data is acquired several times after the operation procedure of placing the augmented material into the posterior maxilla. However a qualitative analysis

can not be obtained just by visual inspection of the physician due to patient movements and nonlinear deformations of the augmented material between data sets. Computer assistance must be provided to combine the volume data sets and present them in a more convenient way to the physician. From a computer vision point of view our objective of fusing volume data sets translates into solving the registration problem using rigid and nonlinear deformation models.

1.2 Image Registration

Registration is an initial step for a large number of medical image analysis applications. Thereby it is a crucial event, since the performance of any further computation is directly dependent on the results achieved during the registration procedure. Informally image registration is the process of aligning two images of the same or similar object by minimizing an application dependent dissimilarity (or distance) measure under the assumption of a specific transformation (or deformation) model [13]. There are two major registration approaches regarded to the transformation model they use: linear (rigid) and nonlinear transformation models. As we want to match only rigid structures in our first task, the rigid registration methods are suitable. However there are several non-rigidly moving bone structures in our data sets such as the cranial bone, the lower-jaw and the spinal column. It is obvious that a global rigid-transformation model can not be found. A nonlinear method will solve this obstacle implicitly.

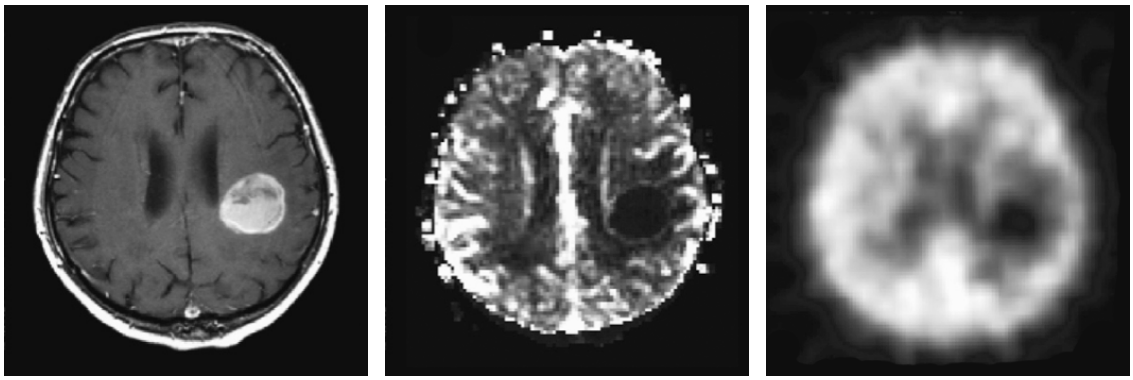
Registrations based on a rigid transformation model are more or less solved but there are many challenges related to non-linear registration [11]. First of all, due to the fact that it is an ill-posed problem, the complexity is much higher compared to any linear method. This yields either to solutions whose runtime is too long for serious clinical applications or lower accuracy solutions caused by information reduction to compensate runtime inefficiency. Another problem is the lack of standardized evaluation methods. Because of the ill posed problem, and the computational complexity there have been many completely different proposals in the current literature to handle this problem, yet up to now there is not an established standard to evaluate these methods [40]. This applies especially to real data evaluation.

1.3 Medical Imaging

In order to develop appropriate registration algorithms it is important to take into account how a medical image/volume is acquired and how the anatomical structures of interest are represented.

1.3.1 Modalities

Medical images are generated by specific acquisition devices. The most important are Computed Tomography (CT), Magnetic Resonance Imaging (MRI), Single Photon Emission Computed Tomography (SPECT), Positron Emission Tomography (PET) and Ultrasound (US). Each of them produces images of different modality. This means that even if the anatomic structure is the same during multiple acquisitions with different modalities, the resulting images can differ. Hereby tissues can be represented with different intensity values or some anatomical structures will be revealed just using a specific modality. It is important to notice that the same device can produce images with different modalities. Figure 1.2 ([21]) shows one slice of the same anatomic structure using three different modalities (two MRI modes and one PET mode). In medical image analysis one distin-



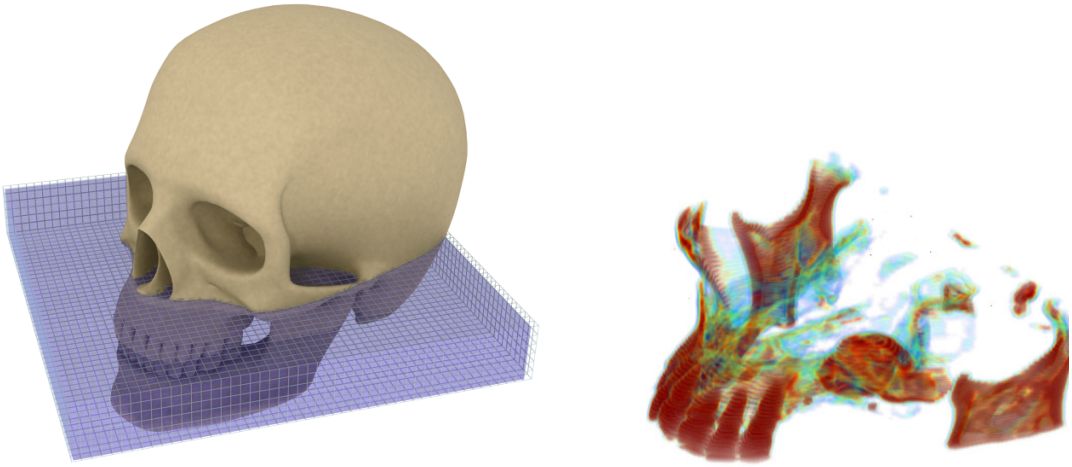
(a) (b) (c)

Figure 1.2: a) MRI image using T_1 mode, b) MRI using T_2 mode and c) PET scan.

guishes intra- or mono-modality and inter- or multi-modality registration. Inter-modality registration represent the aligning process of two patient volume data which are acquired with the same modality, eg. CT-CT or MRI-MRI registration. Inter-modality registration

implies to register anatomic structures acquired with different imaging modalities, eg. CT-MRI, MRI-PET.

The acquisition of the posterior maxilla is done with CT. In Figure 1.3 a) is an illustration of the anatomical structures which are scanned and in Figure 1.3 b) the volume rendering of a real CT scan is shown.



(a) (b)

Figure 1.3: a) Illustration of the anatomical area which is scanned and b) shows a volume rendering of a CT scan of the same area.

1.3.2 Image Representation

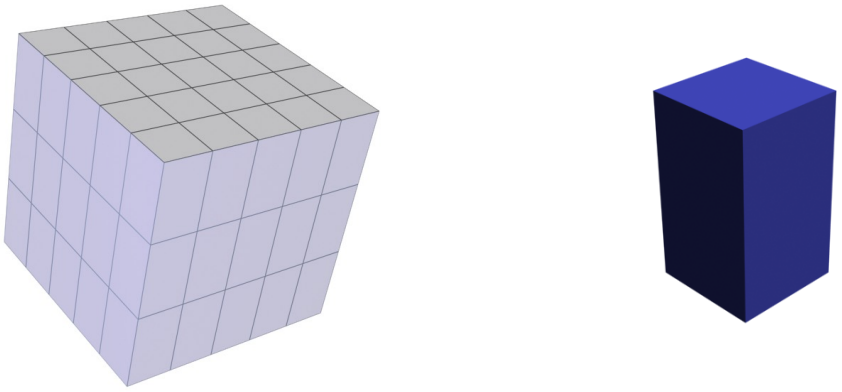
There are many medical image formats representing acquisitions of volumetric data such as DICOM, Analyze and more. However, the contained information associated with the image data is almost the same. The main components of volumetric data sets are:

1. An image grid. It is composed of $x_{size} \cdot y_{size} \cdot z_{size}$ voxels (see Figure 1.4 a)).

2. Spatial spacings. The image has a strong relation to the spatial space. Therefore it is important to bring the data from the image grid to three dimensional space. The spacing thereby indicates the spatial distance between two voxels in the image grid along one of the three main axes. This can be illustrated in 1.4 b) where the

spatial distance between voxels in x direction is stated as $x_{spacing}$, along the y axes as $y_{spacing}$ and along the z axes as $z_{spacing}$. The units are usually given in millimeters.

3. Origin coordinate. The object which is defined by the previous two statements has to be placed somewhere in the three dimensional space. This is defined by the origin coordinates x_{origin}, y_{origin} and z_{origin}. The object is translated from the world coordinate origin $(0, 0, 0)^T$ to the coordinates $(x_{origin}, y_{origin}, z_{origin})^T$. The origin is also given in spatial units, mostly often mm.



(a) (b)

Figure 1.4: a) Demonstration of an three dimensional image grid. The dimensions are scaled with the according spatial spacing. a) is showing the whole grid where b) shows a single voxel.

1.4 Aims of the Thesis

In this thesis we focus on developing fast and accurate registration algorithms for single- or inter-modality volumetric data sets. Therefore we are using two different registration approaches. The first one will allow the physician to compare rigid structures from `CT` data sets using a linear (rigid) transformation model. It is obvious that such a registration method can not be performed completely automatically due to non-rigidly moving bone structures shown in the `CT` data sets. Another drawback is that the structure of interest is performing a nonlinear transformation between two data sets and its intensity values

(HU) do not correspond completely with intensity values of bones. The minimum intensity values of the augmented material are located in the soft tissue intensity range. Therefore our second objective was to develop a powerful registration method to overcome these problems. If a nonlinear method is used local changes in a variety of tissues can be estimated properly. Moreover it can help the physician to estimate the direction of absorption (out of the displacement field). Another advantage is that interesting soft tissue changes can be monitored as well and the method executes without manual interaction. For this purpose we have focused on the extension of a recently developed two dimensional affine flow-based nonlinear registration algorithm [39] to volumetric data sets.

1.5 Synopsis

In Chapter 2 we present the state-of-the-art in the current literature regarding the topic of rigid and nonlinear registration. A review of methods used to register the maxillofacial area is also given. Chapter 3 describes our specific interactive rigid-registration method. We used a special rotation model (versors) and the calculus associated with it is also described. An evaluation on synthetic as well as real dental data is provided, too. Chapter 4 describes our nonlinear registration method based on a variational approach (Affine Flow Registration). There, we introduce a new prior to regularize the flow field based on previous work on two dimensional nonlinear registration. Due to the enormous computational complexity of variational methods we put emphasis on efficient hardware implementation. An evaluation is also presented. Chapter 5 provides the conclusion of this thesis and discusses future work.

Chapter 2

State of the Art

Contents

2.1 A Formal Definition of Image Registration 10

2.2 Rigid Registration 17

2.3 Nonlinear Registration Strategies 28

2.4 Multi-Resolution Strategies 32

2.5 Maxilla Registration 32

The discipline of image registration has evolved tremendously in the last two decades. A full review is hard to establish due to the large number of papers that have been published. However some important surveys have been released during the last years. The first one was released by Brown [6]. It categorized the different, previously independently developed registration techniques. Brown did not cover just the topic of medical image registration, but also registration in computer vision and pattern recognition and remotely sensed data processing. An important survey, specialized on the topic of medical images is the work of Maintz and Viergever [23]. A large number of papers has been reviewed there and a categorization schema based on nine criteria is provided.

The book of Hajnal [13] is a standard work focusing exclusively on the topic of medical image registration. The book of Sonka and Fitzpatrick [35] also provides several chapters focusing on medical image registration.

Our main distinction to classify registration methods is the transformation type they use. We separate them into rigid and nonlinear techniques. The next finer classification level distinguishes between feature- and intensity based methods. Feature based methods

include a preprocessing step in order to extract points of interest (`POI`) or segment surface structures. Intensity based methods operate directly on the image volume data.
Prior to explaining state-of-the-art methods we want to start this chapter by giving a formal definition of the image registration problem.

2.1 A Formal Definition of Image Registration

The following assumptions are based on the work of Urschler [40], Modersitzki [25] and Yoo [42].

2.1.1 Definition

Let's assume we have two images $I_M(\mathbf{x})$ and $I_F(\mathbf{x})$. We will call $I_F(\mathbf{x})$ the fixed image and $I_M(\mathbf{x})$ the moving image. The spatial coordinates $\mathbf{x}$ are belonging to a domain Ω. The spatial dimension of $\mathbf{x}$ is given by $d \in \mathbb{N}$. In our case is $d = 3$ and therefore $\mathbf{x}$ consists of three components x, y and z.

$$I_F(\mathbf{x}): \quad \Omega_F \longmapsto \mathbb{R} \qquad \Omega_F \subset \mathbb{R}^d$$
$$I_M(\mathbf{x}): \quad \Omega_M \longmapsto \mathbb{R} \qquad \Omega_M \subset \mathbb{R}^d$$

Registration means to find a spatial transformation ρ by minimizing a dissimilarity measure $\mathbf{S}$ between the fixed image $I_F(\mathbf{x})$ and the deformed moving image $I_M(\rho(\mathbf{x}))$ (see Figure 2.1). Thereby ρ is defined to map the fixed image coordinate frame $\mathbf{x_F}$ defined on the domain Ω_F to the moving image coordinate frame $\mathbf{x_M}$ defined on the domain Ω_M:

$$\rho(\mathbf{x_F}): \quad \Omega_F \longmapsto \Omega_M \qquad \Omega_F \subset \mathbb{R}^d,\ \Omega_M \subset \mathbb{R}^d$$

The dissimilarity or distance measure $\mathbf{S}$ between the fixed image $I_F(\mathbf{x})$ and the moving image $I_M(\rho(\mathbf{x}))$, both defined on the space of d dimensional images $Img(d)$, can be expressed by:

$$\mathbf{S}: \quad Img(d) \times Img(d) \longmapsto \mathbb{R} \qquad I_F(\mathbf{x}), I_M(\mathbf{x}) \subset Img(d)$$

A mapping $\rho : \mathbb{R}^d \longmapsto \mathbb{R}^d$ has to be determined that minimizes the dissimilarity $\mathbf{S}(\ I_M(\rho(\mathbf{x})), I_F(\mathbf{x})\ ) = min$

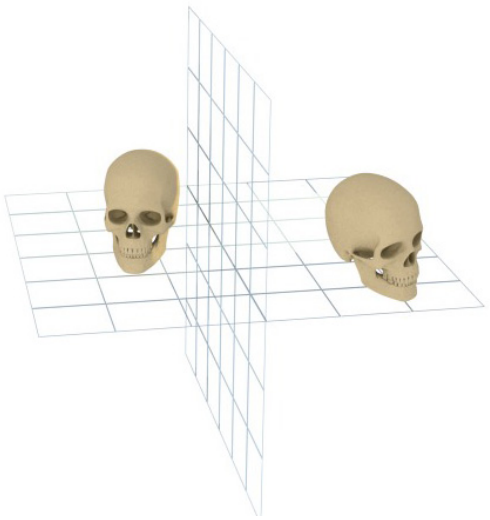


Figure 2.1: Image registration is the task of finding a spatial transformation to map the moving image into the fixed one.

A direct solution is not possible in most cases. An iterative optimization approach is used in the most situations as a numerical implementation.
In the following we will explain the main transformation types, starting with the simplest ones.

2.1.2 Translation

If the fixed point $\mathbf{x}_F$ (x_F, y_F, z_F) is given in three dimensional space, a translation by $\mathbf{t}$ (t_x, t_y, t_z) units can be presented as:

$$\mathbf{x}_M = \mathbf{x}_F + \mathbf{t} \tag{2.1}$$

A more common way to represent a translation is the matrix form ($\mathbf{x}_M = \mathbf{T}\mathbf{x}_F$):

$$\begin{bmatrix} x_M \\ y_M \\ z_M \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_F \\ y_F \\ z_F \\ 1 \end{bmatrix} \tag{2.2}$$

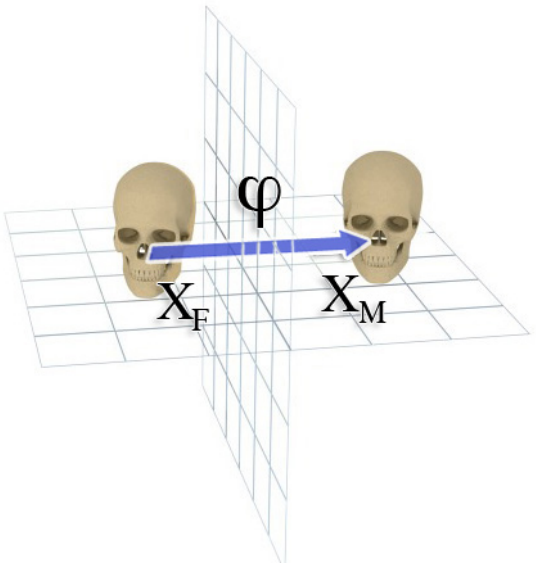


Figure 2.2: Illustration of a translation transformation along one axis.

2.1.3 Rotation

In a two dimensional space a rotation is described by a single angle θ. Considering the fixed point $\mathbf{x}_F$ (x_F, y_F), a transformation to the moving point $\mathbf{x}_M$ (x_M, y_M), by θ radians around the origin can be expressed by:

$$\begin{aligned} x_M &= \cos(\theta)\, x_F + \sin(\theta)\, y_F \\ y_M &= -\sin(\theta)\, x_F + \cos(\theta)\, y_F \end{aligned}$$

In a matrix form, it looks like the following:

$$\begin{bmatrix} x_M \\ y_M \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_F \\ y_F \\ 1 \end{bmatrix} \tag{2.3}$$

In the three dimensional case there are three orthogonal axes that an object can be rotated around. A rotation of θ_X around the x-axis can be performed by ($\mathbf{x}_M = \mathbf{R_x x}_F$):

$$\begin{bmatrix} x_M \\ y_M \\ z_M \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_X) & \sin(\theta_X) & 0 \\ 0 & -\sin(\theta_X) & \cos(\theta_X) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_F \\ y_F \\ z_F \\ 1 \end{bmatrix} \tag{2.4}$$

In the same way rotations around the y and z axis can be carried out:

$$\mathbf{R_y} = \begin{bmatrix} \cos(\theta_Y) & 0 & \sin(\theta_Y) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta_Y) & 0 & \cos(\theta_Y) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{R_z} = \begin{bmatrix} \cos(\theta_Z) & \sin(\theta_Z) & 0 & 0 \\ -\sin(\theta_Z) & \cos(\theta_Z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

To realize rotations along multiple axes the rotation matrices have to be multiplied. E.g. a rotation along all three axes can be written as $\mathbf{R} = \mathbf{R_x R_y R_z}$.

2.1.4 Rigid Transformations

Rigid transformations consist only of rotations and translations. They are a subset of the more general affine transformation. In three dimensional space, rigid transformations are composed of of six parameters, three rotational (θ_X, θ_Y and θ_Z) and three translational components (t_x, t_y and t_z). The rigid transformation can be expressed as:

$$\mathbf{x}_m = \mathbf{R x}_F + \mathbf{T} \tag{2.5}$$

where $\mathbf{R}$ is composed of the rotation matrices $\mathbf{R_x}$, $\mathbf{R_y}$ and $\mathbf{R_z}$:

$$\mathbf{x}_M = (\mathbf{R_x R_y R_z})\, \mathbf{x}_F + \mathbf{T} \tag{2.6}$$

(a) x-axis rotation (b) y-axis rotation (c) z-axis rotation

Figure 2.3: Demonstration of an object rotated around x, y and z axis. The origin is placed in the geometric center of the object.

2.1.5 Scaling

As previously discussed three dimensional medical images contain spacing parameters that allow referencing to physical space. Scaling can be applied to establish an isotropic (equal) spacing. Therefore the points $\mathbf{x}$ have to be multiplied by the scaling matrix ($\mathbf{x}_M = \mathbf{S}\mathbf{x}_F$):

$$\begin{bmatrix} x_M \\ y_M \\ z_M \\ 1 \end{bmatrix} = \begin{bmatrix} s_1 & 0 & 0 & 0 \\ 0 & s_2 & 0 & 0 \\ 0 & 0 & s_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_F \\ y_F \\ z_F \\ 1 \end{bmatrix} \tag{2.7}$$

2.1.6 Affine Transformations

Affine transformations allow more degrees of freedom compared to rigid transformations. In addition to rotation and translation, affine transformations allow also scaling and shearing. Let's assume the fixed point $\mathbf{x}_F$ is given. It is made of three components (x_F, y_F, z_F). An affine transformation of the point $\mathbf{x}_F$ to the moving point $\mathbf{x}_M$ (x_M, y_M, z_M) can be

phrased as:

$$\begin{aligned} x_M &= a_{11}\, x_F + a_{12}\, y_F + a_{13}\, z_F + a_{14} \\ y_M &= a_{21}\, x_F + a_{22}\, y_F + a_{23}\, z_F + a_{24} \\ z_M &= a_{31}\, x_F + a_{32}\, y_F + a_{33}\, z_F + a_{34} \end{aligned} \tag{2.8}$$

which is often represented as a matrix multiplication: ($\mathbf{y} = \mathbf{A}\mathbf{x}$)

$$\begin{bmatrix} x_M \\ y_M \\ z_M \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_F \\ y_F \\ z_F \\ 1 \end{bmatrix} \tag{2.9}$$

2.1.7 Nonlinear Transformations

The naive extension of the affine model to the nonlinear case involves to model the nonlinear transformation with second-, third- or higher order terms. However this approach models just global changes, where usually the changes happen locally. Oscillations of higher order polynomials are also a major drawback.
There are two methods of representing nonlinear transformations in the current literature: 1) using a parametric framework and 2) using a variational function. In a parametric framework n basis functions θ are used to model the transformation.

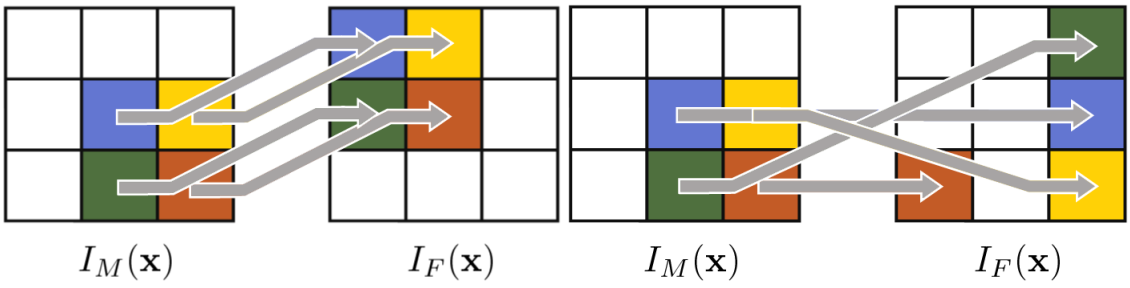
$$\begin{bmatrix} x_M \\ y_M \\ y_M \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ a_{31} & a_{32} & \dots & a_{3n} \\ 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} \theta_1(x_F, y_F, z_F) \\ \vdots \\ \theta_n(x_F, y_F, z_F) \\ 1 \end{bmatrix} \tag{2.10}$$

A basis function can be e.g. a trigonometric function, wavelets or a spline function.
Non-parametric techniques are currently investigated more thoroughly. The basic idea is to formulate an expression, consisting of a similarity measure between the two images and an appropriate non-parametric formulation of the desired transformation model. Lanczos [19] was the first who solved the issue using the calculus of variations. He used them in the field of analytical mechanics. However the same assumptions are valid with nonlinear

registration. Hereby the parametric transformation ρ is replaced by the displacement field $\mathbf{u}(\mathbf{u} : \Omega \subset \mathbb{R}^3)$. It is treated independently from the whole image domain. We use the formulation of Horn and Schunck [15] to demonstrate a nonlinear registration problem. They introduced the following variational formulation:

$$\min_{\mathbf{u}} \left\{ \int_{\Omega} |\nabla \mathbf{u}|^2 \; d\Omega + \lambda \int_{\Omega} (I_M(\mathbf{x} + \mathbf{u}(\mathbf{x})) - I_F(\mathbf{x}))^2 \right\} \tag{2.11}$$

where I_F is the fixed image and I_M the moving image: $(\Omega \subset \mathbb{R}^3) \mapsto \mathbb{R}$. The first term is referring to the regularization (smoothing) of the flow field. It assumes that neighboring points from the moving image have a close proximity to corresponding points in the fixed image, too. This principle is shown in Figure 2.4 a). Locally discontinuous mapping of points (see Figure 2.4 b)) is therefore penalized. The second term measures the similarity between the fixed image $I_F(\mathbf{x})$ and the warped moving image $(I_M(\mathbf{x}+\mathbf{u}(\mathbf{x}))$. λ is a free parameter and weights the importance between the two terms. If λ is small, regularization of the displacement field becomes more important and if it is large the similarity between the two images is the more important factor. Thereby the importance of a smooth displacement field is lowered.



(a) (b)

Figure 2.4: a) Illustration of a regularized displacement field and b) illustration of an unregularized displacement field.

2.2 Rigid Registration

A formal definition of the rigid transformation is given in Section 1.4.4. In the following we will discuss the main concepts of three dimensional rigid registration. The book of Hajnal et al [13] provides additional details about this topic. We will organize the overview by discussing the feature based methods first and then explaining the intensity based methods.

2.2.1 Feature Based Methods

Feature based methods do not make use of the whole information contained in the medical image data. They rather concentrate on specific structures, anatomical or induced by artificial markers, within the data. They can be separated in two main categories by the structure of interest: point-based and surface based methods. Point-based methods are using points in three dimensional space to estimate rigid transformation parameters. These points, which have to be aligned during the registration process, are called fiducial markers or fiducial points.

2.2.1.1 Point-Based Methods

Given two three dimensional images $I_F(\mathbf{x})$ and $I_M(\mathbf{x})$ we want to estimate a rigid transformation $\rho(\mathbf{x})$ that maps points from the moving image to the corresponding points in the fixed image. In the case of rigid anatomical structures, such as bones, three landmarks will be sufficient to estimate the rotation and translation parameters. However, the larger the number of fiducial markers used, the more the precision of the estimated translation rises.

The first step involves identifying fiducial markers. They are pins or markers fixed to the patient. They can be attached to the skin or screwed into the bone. Both versions have their assets and drawbacks. The attached markers do not require an invasive procedure in order to to be placed, but their position can change due to skin movements. Even small position changes can cause serious trouble to perform a valid transformation estimation. On the other side the invasive method is more robust to skin movements but due to its invasive nature it is not very common. It is used just in special cases where high accuracy is improves.

Assuming, a set of points $\{\mathbf{x}_F\}$ is given from the fixed image $I_F(\mathbf{x})$ and the correspond-

ing points $\{\mathbf{x}_M\}$ from the moving image $I_M(\mathbf{x})$. A rigid transformation $\rho(\mathbf{x}_F)$ has to be estimated which will minimize a distance measure d between the points pairs $\{\mathbf{x}_F\}$ and $\{\mathbf{x}_M\}$:

$$d(\mathbf{x}_F, \mathbf{x}_M) = \frac{1}{N} \sum_{i=1}^{N} ||\rho(\mathbf{x}_{F,i}) - \mathbf{x}_{M,i}|| \tag{2.12}$$

The process consists of three main parts:

1. Center fixed points $\mathbf{x}_F$ and moving points $\mathbf{x}_M$
2. Determining rotation matrix $\mathbf{R}$ (Orthogonal Procrustes)
3. Determining translation matrix $\mathbf{T}$

Centering is done by calculating the geometric center points $\overline{\mathbf{x}_F}$ and $\overline{\mathbf{x}_M}$ of all fixed and moving points:

$$\overline{\mathbf{x}_F} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_{F,i} \qquad \overline{\mathbf{x}_M} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_{M,i} \tag{2.13}$$

Then all points are transfered to the origin of the coordinate system:

$$\mathbf{x}_{F_{center}} = \mathbf{x}_F - \overline{\mathbf{x}_F} \qquad \mathbf{x}_{M_{center}} = \mathbf{x}_M - \overline{\mathbf{x}_M} \tag{2.14}$$

The rotation matrix $\mathbf{R}$ will be calculated using Singular Value Decomposition of the fiducial covariance matrix $\mathbf{H}$

$$\begin{aligned} \mathbf{H} &= \sum_{i=1}^{N} (\mathbf{x}_{M_{center},i}\ \mathbf{x}^T_{F_{center},i}) \\ \mathbf{H} &= \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T \\ \mathbf{R} &= \mathbf{V}\mathbf{D}\mathbf{U}^T \quad \text{, where} \\ \mathbf{D} &= diag(1, 1, det(\mathbf{V}\mathbf{U}^\mathbf{T})) \end{aligned}$$

The translation parameters can be derived from the geometric center of the fixed points $\mathbf{x}_F$ and the moving points $\mathbf{x}_M$.

$$\mathbf{T} = \overline{\mathbf{x}_M} - \mathbf{R}\overline{\mathbf{x}_F} \tag{2.15}$$

An illustration of the whole process is shown in Figure 2.5.

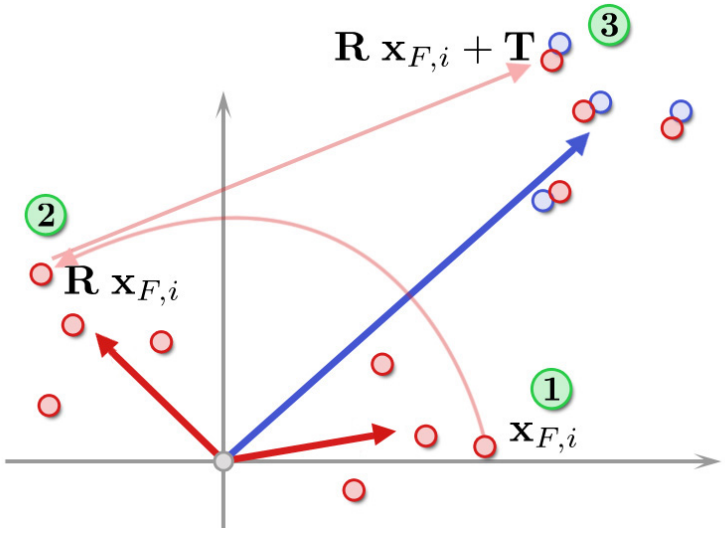
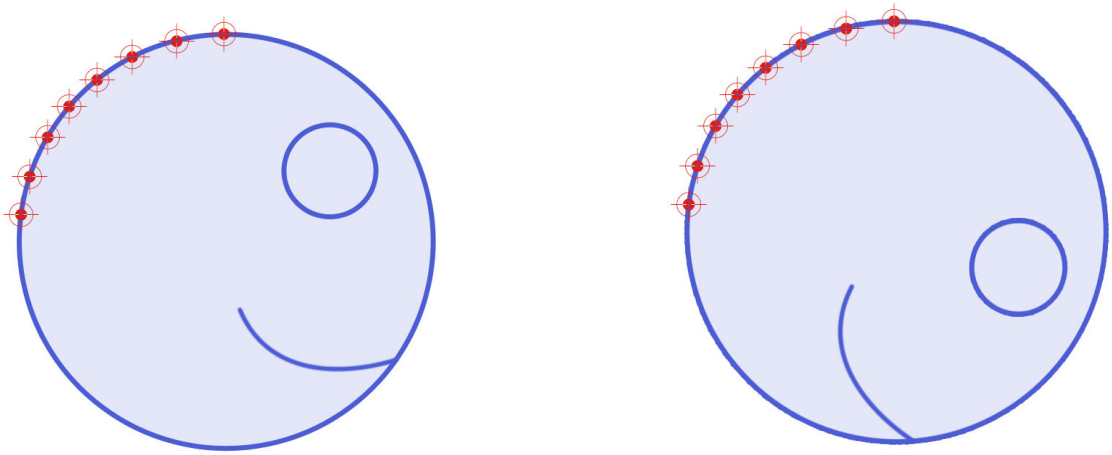


Figure 2.5: An illustration of the Point-Based Alignment Process in 2D.

2.2.1.2 Surface Based Methods

Sometimes surfaces are more distinct than landmarks. Parameters of a rigid transformation ρ can be estimated by fitting two surfaces (of the same anatomic structure) in two images (the fixed image $I_F(\mathbf{x})$ and the moving image $I_M(\mathbf{x})$). A prerequisite is that the surface structures of interest can be segmented. This is valid for surfaces which can be extracted from high contrast areas such as the boundary between the skin tissue and air. The "Head and Hat" algorithm was the first surface based method introduced to match `MRI` and `PET` images. The points extracted from the head surface located in the `MRI` scan are called the "Head" and the points extracted from the same structure on the `PET` scan the "Hat". A progressively refining process will be applied to align the surfaces. To do so at each optimization step the mean square distance between the fixed "Head" points and the "Hat" points will be calculated. During the optimization process the distance between points on the "Head" and the "Hat" will be minimized. However this method will fail if the surface structure shows symmetry to rotation (see Figure 2.6). If the number of surface points is large the calculation of the distance between the "Head" and "Hat" points is the runtime bottle neck of the optimization process. To reduce the computational cost a distance transform can be applied to to the binary "Head" image. The distance transform can be either the Euclidian transform or the Chamfer transform [5].
The most widely used surface matching algorithm is the Iterative Closest Point algorithm.



(a) (b)

Figure 2.6: A 2D illustration of the "Head and Hat" surface alignment procedure. In this case multiple solutions are possible due to the characteristic that our anatomical structure is symmetric to rotations.

The method was introduced by Besl and McKay [4]. The algorithm consists of two stages. Assuming two images are given, $I_F(\mathbf{x})$ is the fixed image and $I_M(\mathbf{x})$ the moving image. Both are containing a similar anatomic structure or object with a surface $\mathbf{S_F}$ in the fixed image and the surface $\mathbf{S_M}$ in the moving image. The surface $\mathbf{S_F}$ is composed of surface points $\{\mathbf{x}_F\}$ and the surface $\mathbf{S_M}$ of surface points $\{\mathbf{x}_M\}$. In the first step we have to identify from every surface point $\mathbf{x}_{F,i}$ the closest surface point $\mathbf{x}_{M,j}$ from the set $\mathbf{x}_M$ on the surface $\mathbf{S_M}$. Thereby a distance d between these points has to be minimal.

$$d(\mathbf{x}_{F,i}, \mathbf{S_M}) = \min_{\mathbf{x}_M \in \mathbf{S_M}} ||\mathbf{x}_M - \mathbf{x}_{F,i}|| \tag{2.16}$$

From this we get a set of corresponding points in relation to the distance between them. Every point $\mathbf{x}_{F,i}$ on the surface $\mathbf{S_F}$ will have its closest point $\mathbf{x}_{M,j}$ on the surface $\mathbf{S_M}$. The second step involves estimating the rigid transformation parameters to map the fixed image points into their corresponding points, estimated during the first step, minimizing a least squares criterion:

$$d(\mathbf{x}_F, \mathbf{x}_M) = \frac{1}{N} \sum_{i=1}^{N} (\rho(\mathbf{x}_{F,i}) - \mathbf{x}_{M,j})^2 \tag{2.17}$$

The Point-based Procrustes method from the previous section can be chosen to compute the transformation $\rho(\mathbf{x})$. The transformation will be applied to the fixed points $\{\mathbf{x}_F\}$ of the surface $\mathbf{S_F}$ and the next iteration will be performed with points $\rho(\{\mathbf{x}_F\})$ and $\{\mathbf{x}_M\}$. The algorithm terminates when the change in mean square error between two iterations is lower than a certain threshold. An improvement to the current method is to start the algorithm multiple times with different initializations.

2.2.2 Intensity-Based Methods

Intensity-Based Methods use voxel intensities directly to optimize a measure in order to estimate a rigid transformation $\rho(\mathbf{x})$ between the fixed image $I_F(\mathbf{x})$ and the moving image $I_M(\mathbf{x})$. They are more flexible than point-based and surface-based methods because they use all the available information. In general they do not require complex data preprocessing, such as segmentation of specific anatomic structures and identifying landmarks in the data which induces errors that have an impact on the later registration steps. However there are a few drawbacks. Operating with the full image content is computationally intensive, especially while working with three dimensional images. The registration procedure in point-based and surface-based methods involves minimizing the distance between physical points. Contrary to that, intensity-based methods minimize a cost function that measure the dissimilarity between the image intensities of corresponding points in the two images. Another obstacle appears when the images are not acquired using the same modality. The relation between the intensity values are not linear anymore.
Due to the computational complexity a closed-form solution is not possible for intensity-based methods. The solution is always obtained during an iterative optimization process. Every intensity-based method has a structure which consists of four main parts: a metric as the similarity measure, an optimization method, a transformation and an interpolation method. The workflow is shown in Figure 2.7.

2.2.2.1 Transformation

The most crucial component of an intensity-based registration framework is the transformation model. It determines the complexity of the whole method. As a transform is typically defined by a set of parameters, the aim of registration is to find optimal values for those parameters with respect to a criterion. The amount of parameters deflects the

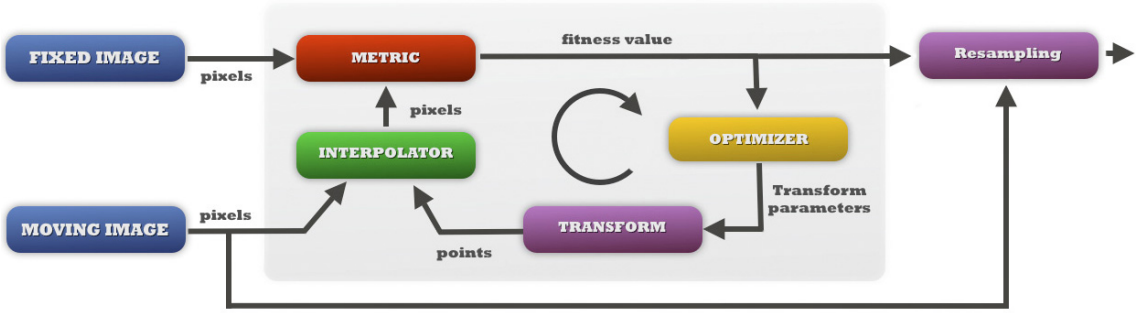


Figure 2.7: The basic intensity-based registration framework composed of an metric as the similarity measure, an optimization method, a transformation and an interpolation method. Taken from the ITK public-domain courseware [1].

degree of freedom a transformation has. If the transform has more degrees of freedom a more complex transformation can be described. On the other hand the computational complexity is rising with the amount of parameters. In the rigid case there are six degrees of freedom as described in the previous chapter.

2.2.2.2 Interpolation

The interpolation is frequently used during the iterative optimization method. Therefore it will have a big impact on the runtime of a registration algorithm. Their degree of accuracy also affects the metric space.

Since the registration is always done in physical space, it is important to consider the interpolation characteristics associated with it. Considering a point $\mathbf{x}$ is given in the fixed image grid. The transformation T_1 will transform the point $\mathbf{x}$ from the fixed image grid to the point $\mathbf{x}^{'}$ from the fixed image coordinate system. The transformation $\rho(\mathbf{x})$ is transforming the point $\mathbf{x}^{'}$ from the fixed image coordinate system to the point $\mathbf{x}^{''}$ of the moving image coordinate system. Finally the point $\mathbf{x}^{''}$ has to be mapped back from the moving image coordinate system to the moving image grid. This is done by the transformation T_2. The whole process can be described by the following equation (see also Figure 2.8):

$$\mathbf{x}^{'''} = T_2(\rho(T_1(x))) \qquad \mathbf{x},\ \mathbf{x}^{'''} \in \mathbb{R}^3 \tag{2.18}$$

In general the point $\mathbf{x}^{''}$ won't point directly to a point on the moving image grid but rather

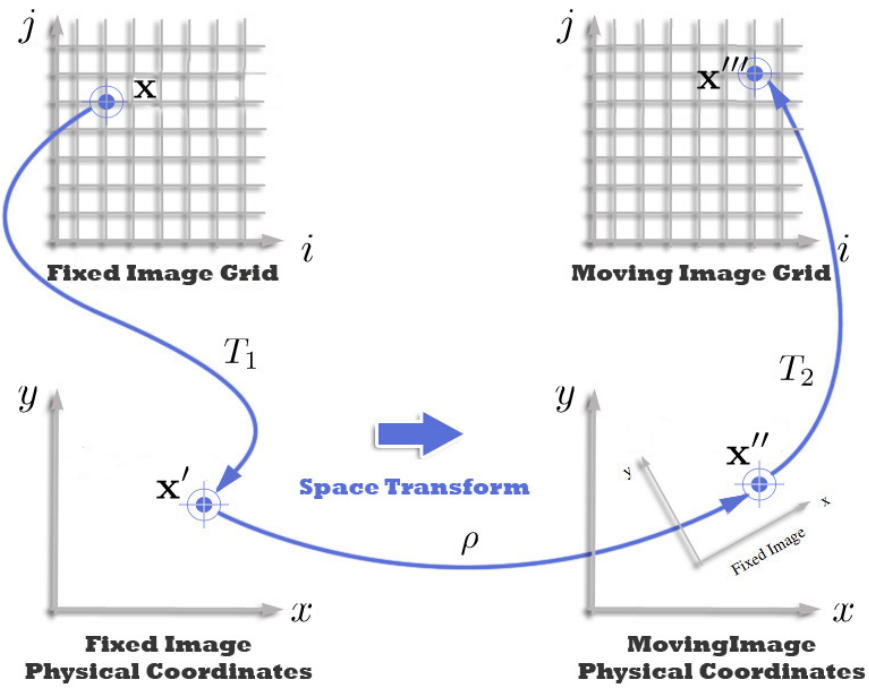


Figure 2.8: Interpolation process during a registration procedure in 2D using physical space coordinates. Taken from the ITK public-domain courseware [1].

somewhere in between. This means that x_1, x_2 and x_3 are no integers. An interpolation method has to be applied to determine the intensity value of the point $\mathbf{x}''$.

There are many different types of interpolation methods. Every method shows a trade-off between computation complexity and interpolation accuracy. If execution speed is the main concern the nearest neighbor interpolation method can be used. Hereby the nearest grid position to the point $\mathbf{x}''$ is used as its intensity value. The computational costs are very cheap. It doesn't even require floating point calculations. On the other hand if the interpolation accuracy is the main concern B-spline interpolation methods can be used.

The most common interpolation method in the current literature is the linear interpolation method. In three dimensional space it is called the tri-linear interpolation. It is a good tradeoff between computational costs and interpolation fidelity.

In Figure 2.9 a detail of a three dimensional image grid is shown. The intensity value of point C, has to be determined. C consist of three coordinates x, y and z. x is a float value between the two integers x_0 and x_1, y lies between the integers y_0 and y_1 and z between the integers z_0 and z_1. Therefore there will be eight neighbor points of point C.

The nearest neighbor method will use the intensity value of the closest neighbor:

$$d(\mathbf{C}_{x,y,z} - \mathbf{V}_{x_0,y_0,z_0}) \leq d(\mathbf{C}_{x,y,z} - \mathbf{V}) \qquad \mathbf{V} \in \mathbf{V}_{x_0,y_0,z_0} \ldots \mathbf{V}_{x_1,y_1,z_1}$$

In our case it will be the intensity value of point $\mathbf{V}_{x_0,y_0,z_0}$. Thereby the intensity value at point x, y and z $\mathbf{I}(\mathbf{C}_{x,y,z})$ is:

$$\mathbf{I}(\mathbf{C}_{x,y,z}) = \mathbf{I}(\mathbf{V}_{x_0,y_0,z_0})$$

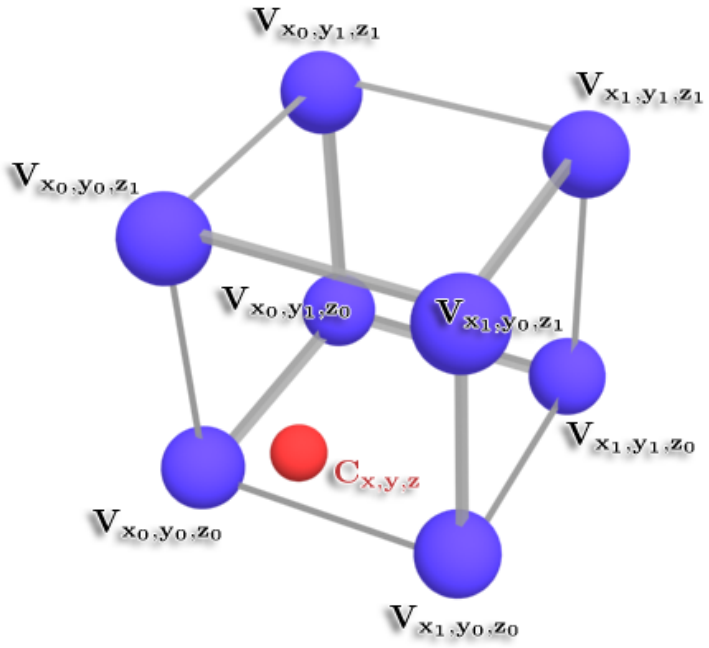


Figure 2.9: A detail of a three dimensional image grid. It demonstrates different interpolation strategies. The intensity value of point $\mathbf{C}_{\mathbf{x},\mathbf{y},\mathbf{z}}$ has to be estimated.

The trilinear interpolation uses all eight neighbors to estimate the intensity value at

the point $\mathbf{C}_{x,y,z}$). The interpolation schema looks like:

$$\begin{aligned}
\mathbf{I}(\mathbf{C}_{x,y,z}) &= \mathbf{I}(\mathbf{V}_{\mathbf{x_0},\mathbf{y_0},\mathbf{z_0}})\ (x_0 - x)\ (y_0 - y)\ (z_0 - z) \\
&+ \mathbf{I}(\mathbf{V}_{x_1,y_0,z_0})\ x\ (y_0 - y)\ (z_0 - z) \\
&+ \mathbf{I}(\mathbf{V}_{x_0,y_1,z_0})\ (x_0 - x)\ y\ (z_0 - z) \\
&+ \mathbf{I}(\mathbf{V}_{x_0,y_0,z_1})\ (x_0 - x)\ (y_0 - y)\ z \\
&+ \mathbf{I}(\mathbf{V}_{x_1,y_0,z_1})\ x\ (y_0 - y)\ z \\
&+ \mathbf{I}(\mathbf{V}_{x_0,y_1,z_1})\ (x_0 - x)\ y\ z \\
&+ \mathbf{I}(\mathbf{V}_{x_1,y_1,z_0})\ x\ y\ (z_0 - z) \\
&+ \mathbf{I}(\mathbf{V}_{x_1,y_1,z_1})\ x\ y\ z
\end{aligned}$$

2.2.2.3 Metrics

Another critical component inside the intensity-based registration framework is the Metric component. It measures how well a transformation $\rho(\mathbf{x})$ describes the mapping of the moving image $I_M(\mathbf{x})$ to the fixed image $I_F(\mathbf{x})$. The metric can be seen as a function of the two images and the transformation:

$$\mathbf{S}(I_M(\mathbf{x}), I_F(\mathbf{x}), \rho(\mathbf{x})) \qquad \mathbf{S} \in \mathbb{R}$$

There are many metric types. The selection of a specific type is dependent on the registration problem. Some metrics are just suited for single modality registration where others can be used for multi modality registration. Some have the property to be sensitive to outliers in the data where others are more robust.
The most simple similarity measure between images is the sum of the mean squares of all intensity values (`SSD`). It can be used if the images are from the same modality.

$$SSD = \frac{1}{N} \sum_{\mathbf{x}_F} \left| I_F\left(\mathbf{x}_F\right) - I_M\left(\rho\left(\mathbf{x}_F\right)\right) \right|^2 \tag{2.19}$$

The sum of squared difference measure is very sensitive to outliers with large intensity differences between the fixed image $I_F(\mathbf{x})$ and the moving image $I_M(\mathbf{x})$. To overcome this

problem the sum of absolute differences is used as the similarity measure (SAD).

$$SAD = \frac{1}{N} \sum_{\mathbf{x}_F} |I_F(\mathbf{x}_F) - I_M(\rho(\mathbf{x}_F))| \tag{2.20}$$

The SSD method assumes that the fixed image and the moving image differ only by Gaussian noise after a successful registration process. A less strict assumption is that their voxel intesity values have a linear relationship. Therefore the correlation coefficient CC will be the optimum similarity measure:

$$CC = \frac{\sum(I_F(x_F) - \overline{I_F}) \cdot (I_M(\rho(x_F)) - \overline{I_M})}{\{\sum(I_F(x_F) - \overline{I_F})^2 \cdot \sum(I_M(\rho(x_F)) - \overline{I_M})^2\}^{1/2}} \tag{2.21}$$

where $\overline{I_F}$ represents the mean intensity in the fixed image $I_F(x_F)$ and $\overline{I_M}$ the mean intensity of the moving image $I_M(\rho(x_F))$ in the overlap region.

The process of registration can also be described in another way. It can be described as trying to maximize the amount of shared information in two images. When the images are perfectly aligned, corresponding structures will overlap. On the other hand when the images are out of alignment, there will be duplicate structures in both images. An additional benefit is that this kind of measure can be also used for inter-modality registration. These methods were independently proposed by Wells et al. [41] and Maes et al. [22]. An extensive review is done by Pluim et al. [30].

Hence information is used as a metric measure. The most common information measure is the Shannon-Wiener information entropy measure H. In our case of three dimensional images, the images are considered as one dimensional vectors. $H(I)$ describes how much information is contained in the image I. A high value means that the voxel intensity values differ much with high frequency while low values depict a mainly small number of different pixels with low frequency. In other words the voxel data can be compressed in the former case.

$$H(I) = -\sum_i p_i \ \log(p_i) \tag{2.22}$$

If the two images I_F and I_M are completely uncorrelated the joint entropy $H(I_F, I_M)$ is the sum of their single entropies $H(I_F)$ and $H(I_M)$. However, this is usually not the case.

Therefore the following inequation is always correct:

$$H(I_F, I_M) \leq H(I_F) + H(I_M) \tag{2.23}$$

and the joint entropy $H(I_F, I_M)$ is defined as:

$$H(I_F, I_M) = -\sum_{i=1}^{N}\sum_{j=1}^{N} p^{\rho}_{FM}(i,j)\log(p^{\rho}_{FM}(i,j)) \tag{2.24}$$

where $p^{\rho}_{FM}(i,j)$ represents the Probability Distribution Function (`PDF`) of the images I_F and I_M in their overlap region. The joint entropy has to be minimized in order to achieve alignment between the images. However the joint entropy has a drawback. The overlap regions can degrade to match just structures of no interest, such as air resulting in the joint entropy also being minimal. To handle this obstacle, the original image entropies $H(I_F)$ and $H(I_M)$ must be used. This new measure is called Mutal Information (`MI`):

$$\begin{aligned} MI(I_F, I_M) &= H(I_F) + H(I_M) - H(I_F, I_M) \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N} p^{\rho}_{FM}(i,j)\log\left(\frac{p^{\rho}_{FM}(i,j)}{p_F(i)\cdot p^{\rho}_M(j)}\right) \end{aligned}$$

where $p_F(i)$ represents the marginal `PDF` of the fixed image and $p^{\rho}_M(j)$ the marginal `PDF` of the moving image. The `MI` has to be maximized during registration.
The issue of the overlap regions is finally solved by Studholme et al. [36] who used an alteration of the `MI` as a metric measure. It is called the Noramlized Mutal Information (`NMI`) and is defined by:

$$NMI(I_F, I_M) = \frac{H(I_F) + H(I_M)}{H(I_F, I_M)}$$

2.2.2.4 Optimizer

The optimizer has the task to optimize the parameters of the transform component with respect to the dissimilarity value provided by the metric component. Some types of the optimizer require a derivative information and others do not. The derivative information is always provided by the metric component. The outcome of this is that the selection of the metric component is highly dependent on the optimizer component and vice versa.

Optimization is an iterative process. At the beginning the parameters of the transformation are initialized to some values. In every further iteration the metric output is evaluated by the optimizer component and new parameters of the transformation are chosen according to the parameters search space. Due to the fact that transform parameters have often different units, such as angles (radian) and translations (mm), the dynamic ranges of them can vary. This is often a serious obstacle during the search of new parameters in the N dimensional search space. If we assume that a rigid transformation is used as the transformation model, some unit changes in the angles have much more impact on the outcome then changes in the translation. It can be handled to some degree by rescaling the parameters.

2.3 Nonlinear Registration Strategies

There are many applications that require nonlinear transformations to describe the spatial relationship between structures in different images adequately. Any tissue deformations due to interventions or changes over time can not be explained by a rigid transformation. The same assumptions are valid during registration of medical volumes of different patients in order to build medical atlases. The substantial anatomical variability can not be accommodated with rigid transformation models.
Nonlinear registration methods can be separated in feature- and intensity based methods similar to the rigid methods. In this thesis we are focusing on intensity based methods. More details about nonlinear feature-based registration can be found in the books of Hajnal et al. [13] and Yoo [42] or in the PhD thesis of Urschler [40].

2.3.1 Intensity-Based Methods

Intensity-based methods do not rely on a preprocessing step in order to extract specific features. They use the whole volume data during the registration. As mentioned in Section 1.4.7 we can define nonlinear transformations using parametric and non-parametric methods. Parametric methods will allow us to reuse the framework shown in Figure 2.7. However due to the much higher complexity, resulting from a transformation model with much higher degrees of freedom, the optimization algorithm has to be designed with special care. An applied method of this technique, the parametric B-spline deformable registration

algorithm, was introduced by Rueckert et al. [33].
We want to focus on non-parametric nonlinear methods. Moreover we want to present a relation between the topic of nonlinear registration and the optical flow problem, since our registration method (Chapter 4) is a generalization of a method designed to solve the optical flow problem. A more detailed review on non-parametric nonlinear methods is provided by Hajnal et al. [13] and Urschler [40].

2.3.1.1 Optical Flow Estimation

The roots of non-parametric nonlinear registration methods come from the optical flow estimation problem. The recovery of motion has been a major task of many artificial vision systems. The main objective of optical flow methods is to compute a flow field estimating the motion in consecutive images [31].

2.3.1.2 Variational Formulation

Let's assume two three dimensional images I_F and I_M: $(\Omega \subset \mathbb{R}^3) \mapsto \mathbb{R}$ are given. A disparity map $\mathbf{u} : \Omega \subset \mathbb{R}^3$ has to be found that minimizes a cost function, consisting of a disparity map regularization measure and an image-similarity measure:

$$\min_{\mathbf{u}} \left\{ \int_{\Omega} \psi \left(\nabla \mathbf{u} \right) \, d\Omega + \lambda \int_{\Omega} \zeta \left(I_M(\mathbf{x} + \mathbf{u}(\mathbf{x})) - I_F(\mathbf{x}) \right) \right\} \tag{2.25}$$

Many different types of regularization priors $\psi \left(\nabla \mathbf{u} \right)$ and data fidelity priors $\zeta \left(I_M(\mathbf{x} + \mathbf{u}(\mathbf{x})) - I_F(\mathbf{x}) \right)$ can be chosen. However the difficulty to obtain a computational solution can be much higher for priors which allow more freedom to the flow field, such as discontinuities. λ determines the ratio between the regularization and the optical flow constraint.

2.3.1.3 Model of Horn and Schunck

Horn and Schunck were the first to formulate the optical flow problem in the framework of the calculus of variations [15]. The computational solution of Horn and Schuncks model ($\zeta(x) = x^2$ and $\psi(y) = y^2$) can be obtained easily, due to the quadratic exponent both in

the regularization term prior and the data fidelity prior.

$$\min_{\mathbf{u}} \left\{ \mathbf{E} = \int_{\Omega} |\nabla \mathbf{u}|^2 \; d\Omega + \lambda \int_{\Omega} (I_M(\mathbf{x} + \mathbf{u}(\mathbf{x})) - I_F(\mathbf{x}))^2 \right\} \tag{2.26}$$

where the gradient of the scalar field $\mathbf{u}$ is given as:

$$\nabla \mathbf{u} = \left(\frac{\partial \mathbf{u}}{\partial x}, \frac{\partial \mathbf{u}}{\partial y}, \frac{\partial \mathbf{u}}{\partial z} \right)^T \qquad |\nabla \mathbf{u}|^2 = \left(\frac{\partial \mathbf{u}}{\partial x} \right)^2 + \left(\frac{\partial \mathbf{u}}{\partial y} \right)^2 + \left(\frac{\partial \mathbf{u}}{\partial z} \right)^2 \tag{2.27}$$

In our case (three dimensional space) $\mathbf{u}$ is composed of the three components u_1, u_2 and u_3. Therefor the following assumptions are valid:

$$|\nabla u_1|^2 = \left(\tfrac{\partial u_1}{\partial x}\right)^2 + \left(\tfrac{\partial u_1}{\partial y}\right)^2 + \left(\tfrac{\partial u_1}{\partial z}\right)^2; \quad |\nabla u_2|^2 = \left(\tfrac{\partial u_2}{\partial x}\right)^2 + \left(\tfrac{\partial u_2}{\partial y}\right)^2 + \left(\tfrac{\partial u_2}{\partial z}\right)^2; \quad |\nabla u_3|^2 = \left(\tfrac{\partial u_3}{\partial x}\right)^2 + \left(\tfrac{\partial u_3}{\partial y}\right)^2 + \left(\tfrac{\partial u_3}{\partial z}\right)^2$$

The Euler-Lagrange equations of the functional look as follows:

$$\frac{\partial \mathbf{E}}{\partial u_1} = \tfrac{\partial}{\partial x}\left(\tfrac{\partial u_1}{\partial x}\right) + \tfrac{\partial}{\partial y}\left(\tfrac{\partial u_1}{\partial y}\right) + \tfrac{\partial}{\partial z}\left(\tfrac{\partial u_1}{\partial z}\right) + \lambda (I_M(\mathbf{x}+\mathbf{u}(\mathbf{x})) - I_F(\mathbf{x})) \tfrac{\partial}{\partial x}(I_M(\mathbf{x}+\mathbf{u}(\mathbf{x})) - I_F(\mathbf{x})) = 0$$

$$\frac{\partial \mathbf{E}}{\partial u_2} = \tfrac{\partial}{\partial x}\left(\tfrac{\partial u_2}{\partial x}\right) + \tfrac{\partial}{\partial y}\left(\tfrac{\partial u_2}{\partial y}\right) + \tfrac{\partial}{\partial z}\left(\tfrac{\partial u_2}{\partial z}\right) + \lambda (I_M(\mathbf{x}+\mathbf{u}(\mathbf{x})) - I_F(\mathbf{x})) \tfrac{\partial}{\partial y}(I_M(\mathbf{x}+\mathbf{u}(\mathbf{x})) - I_F(\mathbf{x})) = 0$$

$$\frac{\partial \mathbf{E}}{\partial u_3} = \tfrac{\partial}{\partial x}\left(\tfrac{\partial u_3}{\partial x}\right) + \tfrac{\partial}{\partial y}\left(\tfrac{\partial u_3}{\partial y}\right) + \tfrac{\partial}{\partial z}\left(\tfrac{\partial u_3}{\partial z}\right) + \lambda (I_M(\mathbf{x}+\mathbf{u}(\mathbf{x})) - I_F(\mathbf{x})) \tfrac{\partial}{\partial z}(I_M(\mathbf{x}+\mathbf{u}(\mathbf{x})) - I_F(\mathbf{x})) = 0$$

The first order Taylor approximation for $I_M(\mathbf{x} + \mathbf{u}(\mathbf{x}))$ is used to handle the nonlinearity of $\mathbf{u}(\mathbf{x})$. Due to the huge amount of unknown variables a fixed solution to the linearized equation system is not applicable. An iterative method is therefore used to find the solution.

This method suffer from two drawbacks. The first is that it does not allow discontinuities in the displacement field and the second issue is the sensitivity of the method to outliers.

2.3.1.4 Demons Algorithm

A minor alteration (simplification) of the method of Horn and Schunck is the "Demons" algorithm introduced by Thirion [38]. It is one of the most popular nonlinear registration methods at the moment. The algorithm is solely based on the notion of smoothing the displacement field. The norm of the gradient is penalized, forcing a small difference be-

tween neighboring displacement vectors.
The regularization of the SSD data term and the displacement field is decoupled. Therefore the optimization is performed in two steps. First the displacement field is regularized due to Gaussian filtering approximating the Laplacians as the results of the Euler-Lagrange Equations. The second part involves applying the similarity measure with a normalization term to stabilize the solutions in order to preserve discontinuities in the flow.
The algorithm iterates by starting with an initial solution of the flow field $\mathbf{u} = \mathbf{u}_0$ and then applying the following update step (data similarity measure):

$$\mathbf{u}(\mathbf{x})_{update} = \frac{I_M(\mathbf{x} + \mathbf{u}(\mathbf{x})) - I_F(\mathbf{x})}{|\nabla I_F(\mathbf{x})|^2} \nabla I_F(\mathbf{x}) \tag{2.28}$$

Then a Gaussian-filtering step is applied to perform the global regularization of the deformation field. The update step induced by the data similarity measure can cause problems in untextured regions $\nabla I_F(\mathbf{x}) = 0$. Therefore a more stable update step is used:

$$\mathbf{u}(\mathbf{x})_{update} = \frac{I_M(\mathbf{x} + \mathbf{u}(\mathbf{x})) - I_F(\mathbf{x})}{|\nabla I_F(\mathbf{x})|^2 + (I_M(\mathbf{x} + \mathbf{u}(\mathbf{x})) - I_F(\mathbf{x}))^2} \nabla I_F(\mathbf{x}) \tag{2.29}$$

2.3.1.5 TV-L^1 Optical Flow

An improvement of the method of Horn and Schunck is achieved by using $\zeta(x) = |x|$ as the displacement field regularization prior and $\psi(y) = |y|$ as the flow field regularization in the variational framework in section 2.2.1.2. This yields to the Total Variation regularization functional consisting of the L^1 data penalization:

$$\min_{\mathbf{u}} \left\{ \int_\Omega |\nabla \mathbf{u}| \; d\Omega + \lambda \int_\Omega |I_M(\mathbf{x} + \mathbf{u}(\mathbf{x})) - I_F(\mathbf{x})| \right\} \tag{2.30}$$

This method was used by Zach et al [43] and enabled discontinuities in the displacement field. Due to the TV-L^1 formulation it is also robust to outliers. In principle the complexity to obtain an analytical solution is much higher due to the non-quadratic expressions. However, Chambolle showed an efficient strategy to solve this equation based on a dual variable formulation ([8]).

2.4 Multi-Resolution Strategies

Intensity based registration is commonly an iterative optimization process. Therefore a global solution can not be ensured. If the parameter space is large chances to get stuck in a local optimization minimum are very high. To avoid this a multi-resolution approach can be applied.
Consider two input images, the moving image and the fixed image. Both are filtered by a gaussian kernel and then down-sampled forming Gaussian pyramids (see Figure 2.10).

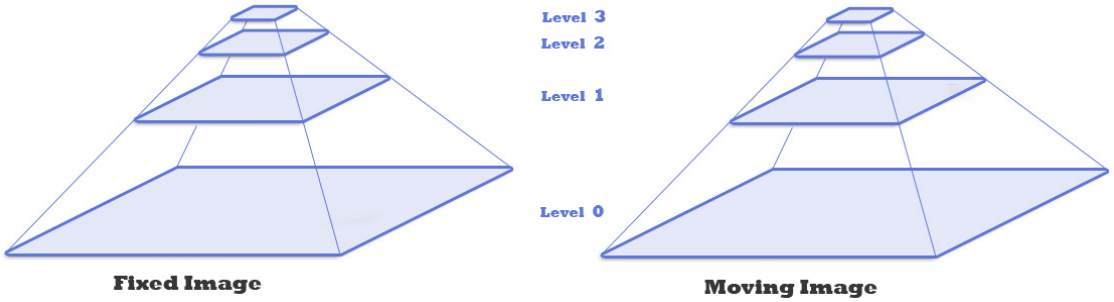


Figure 2.10: Image pyramid showing the hierarchical multi-resolution approach. Taken from the ITK public-domain courseware [1].

The registration is started on the coarsest level of the image pyramid and initialized with the identity transform. The results of the registration on the coarsest level is used as an initialization for the next finer level. An additional advantage is the speed-up of the whole registration process, because the coarsest level provides a good estimation of the transformation parameters for the next finer level and the runtime on a coarse level is performed much faster due to the smaller amount of voxel data. Saved time can be used to iterate longer on the finest level in order to provide registration solutions with higher accuracy.

2.5 Maxilla Registration

There have been published several papers aiming the registration of the maxillofacial area. However the majority of the sophisticated methods are designed for two dimensional registration tasks.
Several techniques have been published to register two dimensional radiographic images.

Nikaido [26] proposed a phase-based image registration algorithm for two dimensional x-ray images. Zacharaki [44] used an intensity based method to align also two dimensional x-ray images. He used the affine transformation model and information theoretic (cross correlation and mutal information) similarity measures. Chen and Jain[10] proposed a feature based method to align dental radiographs. In the first step they extracted the features (surfaces) by enhancing the image data using anisotropic diffusion and afterwards applying a Mixture of Gaussians models to segment the dental work. A matching method based on the distance between the corresponding teeth is then used to measure the similarity between the two radiographs. Leung [20] produced also a nonlinear registration method based on a variational formulation.

Baumert et al [3] solved the problem of misalignment by introducing a special fixation mask. The most methods for dental registration in three dimensional spatial space are using fiducial markers. Such feature based methods are used by Mischkowski et al [24] and Labadie et al [18]. None of these presented methods showed significant relevance for solving our specific problem, therefore we investigated novel techniques in this work.

Chapter 3

An Interactive Rigid Registration Algorithm

Contents

3.1 Preprocessing 36
3.2 Manual Masking 36
3.3 Registration 39
3.4 Implementation 49
3.5 Evaluation Experiments & Results 49

The proposed method consists of three main parts: 1) Preprocessing, 2) Manual Masking and 3) Registration. The Preprocessing and Registration steps are performed automatically whereas the Manual Masking step requires user interaction. The Preprocessing

Figure 3.1: A survey of the Rigid Registration Model.

step involves acquiring volume data from clinical DICOM CT data sets, and extracting bone

structures out of them. Implicated by some hassles shown in most volume data, such as noise artifacts and non-rigidly moving structures, the Manual Masking step involves selecting the maxilla. The Registration process, an intensity-based method, computes a rigid transformation to map one volume into another. The result is a difference image between these two registered volumes.

3.1 Preprocessing

Considering that our main interest lies in the deformation of bone structures, we can exclude soft tissue from our registration process. In order to do so we had to apply a voxel-intensity mapping function. Although we want to remove all structures except bones we do not want a binary image as a result. The variation inside the bone intensities must be retained. Therefore we set a first threshold at 400 `HU`. All lower values will be mapped to 0. Values between 400 `HU` and 2000 `HU` will be mapped linearly to values between zero and one thousand `HU`.
There are often slices where certain areas (mostly teeth) are overexposed (Figure 3.3). To reduce their impact on the metric we map all values above 2000 `HU` to 1000 `HU`. This process is illustrated in figure 3.2.

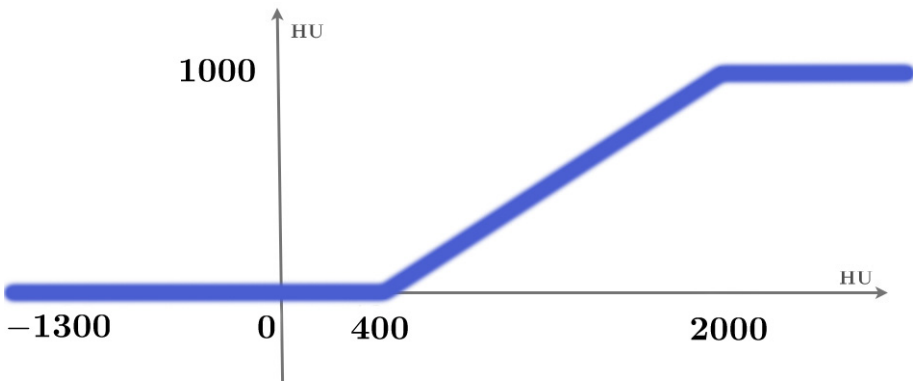
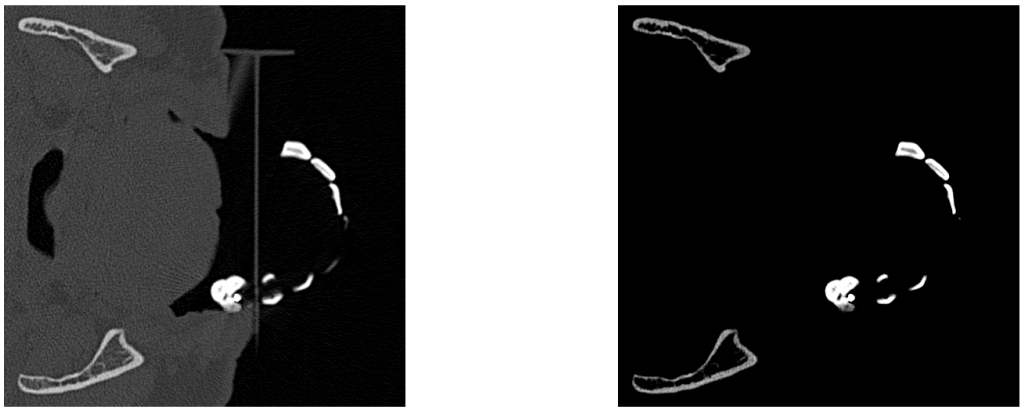


Figure 3.2: Mapping process to extract bone structures.

3.2 Manual Masking

There are two main obstacles that prevent a complete automatic registration:

- Artifacts, expressed by ray-like structures shown in Figure 3.4.

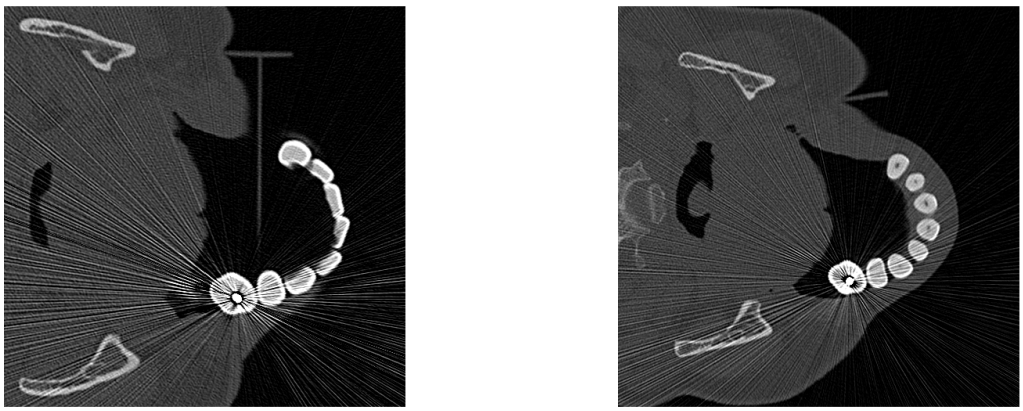


(a) before preprocessing (b) after preprocessing

Figure 3.3: Axial view on two volume slices showing the preprocessing method.

- Non-rigidly moving structures, such as the lower-jaw and the cranial bone, shown in Figure 3.5.

The artifacts are produced during the acquisition of a CT-scan. These effects were described by Svendsen [37] and Kamel [17].



(a) (b)

Figure 3.4: Axial view on two volume slices showing noise artifacts.

Not all bone structures, shown in our CT data, are moving rigidly to each other. The lower jaw and cranial bone including the maxilla are not moving rigidly. Therefore it is not possible to obtain a global rigid transformation model to map all bone structures from one data set into the other.
The spinal column is also presented in some data sets, whose movement is also independent of the cranial bone. We are solving these problems by defining a volume mask to cover

(a) lower-jaw possition 1 (b) lower-jaw possition 2

Figure 3.5: Frontal and sagittal view of non-rigid structures in the data.

just the cranial bone and the maxilla. This process consists of five steps:

1. Select the first slice S_1 where the maxilla or rather the teeth on them are presented.

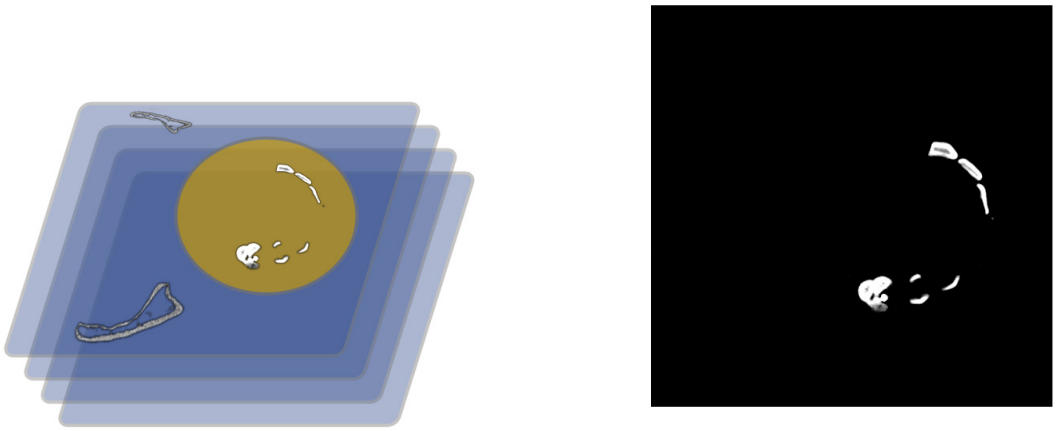
2. Use N points on that slice to define a polygon surrounding the desired structure.

3. Select the last slice S_k where the maxilla is presented.

4. Use again N points on that slice to define a polygon surrounding the desired structure.

5. Make a linear interpolation between the two point sets on the slices S_2 to slice S_{k-1} and build a binary image containing the result.

Figure 3.6 is showing the result of this process.



(a) masking process (b) result

Figure 3.6: Result of Manualy Masking the maxilla.

3.3 Registration

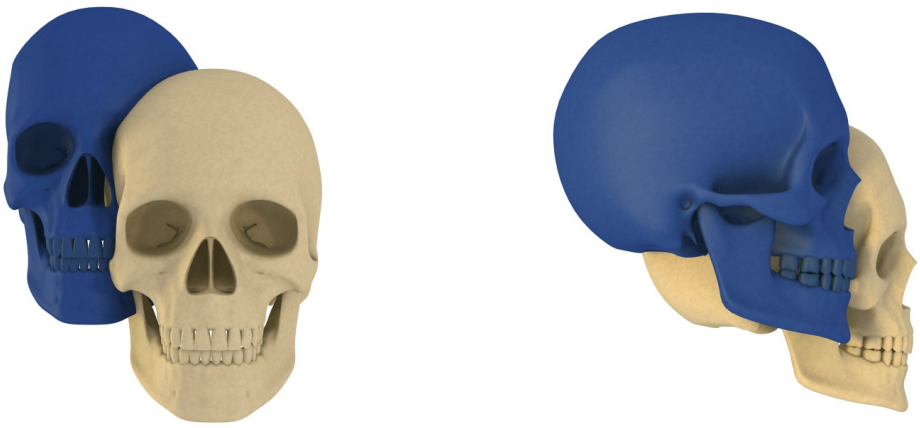
3.3.1 Initialization

The Registration is performed in physical space. Because the position of a patient varies on every CT scan it is important to pre-align them before performing registration. Even though data changes are not significant between muliple CT scans, the translation in physical space can be. An example of a not pre-aligned data set is shown in Figure 3.7. The data sets acquired at two different points of time are the same, but the acquisition position of the patient was not. In this case our optimizer gets stuck in a local minimum.

There are two possibilities to solve this issue:

- Pre-aligning data sets in relation to their **Center of Gravity**.
- Pre-aligning data sets in relation to their **Geometric Center**.

The difference between those two methods is that the center of gravity uses normalized image intensity values and their geometric information, where a geometric center uses just the geometric points of an object and treats the image as a binary image. The center of



(a) frontal view (b) transversal view

Figure 3.7: Transversal view.

gravity in a three dimensional space is defined by:

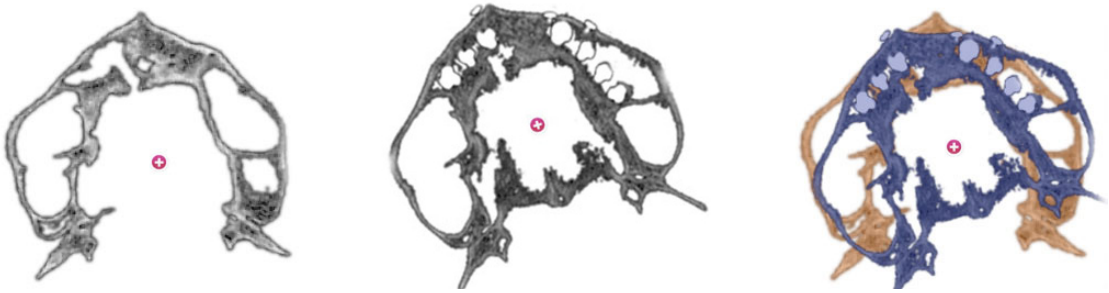
$$\mathbf{C}_{gravity}(x,y,z) = \frac{1}{N}\sum_{i=1}^{N} I(x,y,z)\cdot(x,y,z) \tag{3.1}$$

where N represents all voxels which belong to an object. (x,y,z) is defined by the x, y and z coordinate of these voxels.
The geometric center is defined by:

$$\mathbf{C}_{geometry}(x,y,z) = \frac{1}{N}\sum_{i=1}^{N}(x,y,z) \tag{3.2}$$

where N represents all voxels which belong to an object. In our case they are all voxels with radiometric values above a threshold and thereby represent bone structures.
To prevent that our pre-aligning process can be affected by inhomogeneities in the bone intensity values we used the geometrical center for our application. We translate all object points with $-\mathbf{C}_{geometry}(x,y,z)$ to the origin. A demonstration is shown in Figure 3.17.

Another important issue is to set the correct origin of an image. Since we will iteratively update the rotation parameters the origin must be set properly to be able to estimate a rotation. Our origin is also set during the pre-aligning process and is placed in the geometric center of an image.



(a) volume slice at time t_1 (b) volume slice at time t_2 (c) pre-aligned volume slices

Figure 3.8: Volume slices of two data sets an their geometry center.

3.3.2 Intensity-Based Registration

The work-flow of our intensity-based registration is shown in Figure 3.9. It consists of four main components, a similarity metric, an optimization method, a transformation model and an interpolation method.

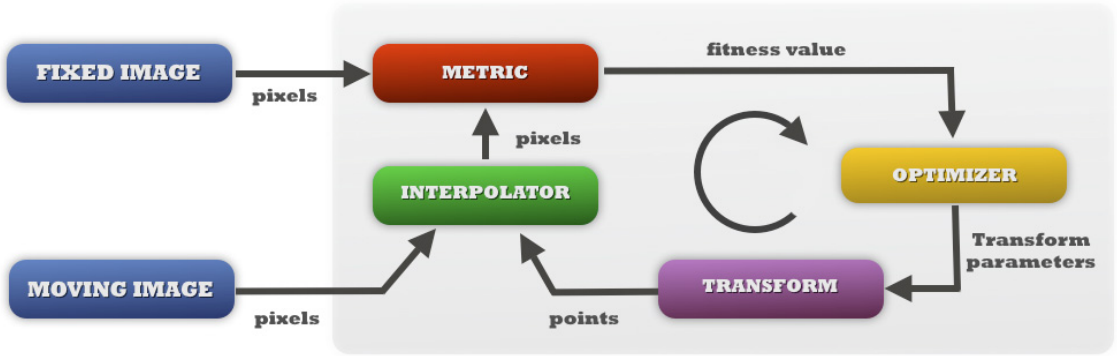


Figure 3.9: Our basic image registration framework.

3.3.3 Quaternions and Versors

There are three main models to describe a rigid object rotation: euler angles, rotation matrices and quaternions. The most common way is certainly using euler angles. The rotation can be described by three parameters: a rotation angle along the x-axis, a rotation angle along the y-axis and a rotation angle along the z-axis. However euler angles are not

differentiable at values of $\pi/2$ and $-\pi/2$. This leads to ambiguities at these points during the optimization step, performed by the gradient descent optimizer. The rotation matrices suffer from the same drawbacks. Another discomfort is that they need nine parameters in 3D. To avoid these disadvantages we use quaternions as our rotational model. In the following we will discuss some details associated with quaternions. The two main sources were the book of T. Yoo [42] and the `ITK` documentation, provided on their website [1] where also the following illustrations are taken from.

A quaternion is the quotient of two vectors. Thereby it is also a relationship between two vectors and it makes it possible to retrieve one vector from the other using the quaternion as an operator. We can also interpret it as an extension of a vector model. In the same way that a vector represents the relation between two points a versor represents the relation between two vectors.

The quaternion $\mathbf{Q}$ is given by the quotient of two vectors $\overrightarrow{A}$ and $\overrightarrow{B}$:

$$\mathbf{Q} = \frac{\overrightarrow{A}}{\overrightarrow{B}} \tag{3.3}$$

If the quaternion operator $\mathbf{Q}$ is applied to the vector $\overrightarrow{B}$ it will produce the vector $\overrightarrow{A}$:

$$\overrightarrow{A} = \mathbf{Q} \star \overrightarrow{B} \tag{3.4}$$

where the symbol $\star$ stands for the application of the quaternion to a vector.

Every vector can be characterized by its magnitude and direction. Since a quaternion represents the quotient between them it also represents the relation between the magnitudes of two vectors and the orientation of one vector to another. According to Hamilton [14] a quaternion can be split into two sub-components: a tensor and a versor. A tensor represents the ratio between the magnitudes of two vectors and the versor the relative orientation between them.

In our case, the vector $\overrightarrow{B}$ is rotated by the versor operator and scaled by the tensor

operator:

$$\begin{aligned} \overrightarrow{A} &= \mathbf{T} \star \left(\mathbf{V} \star \overrightarrow{B} \right) \\ &= (\mathbf{T} \diamond \mathbf{V}) \star \overrightarrow{B} \\ &= \mathbf{Q} \star \overrightarrow{B} \end{aligned} \tag{3.5}$$

where $\mathbf{T}$ is the tensor operator, $\mathbf{V}$ the versor operator and the symbol $\diamond$ represents the

(a) (b)

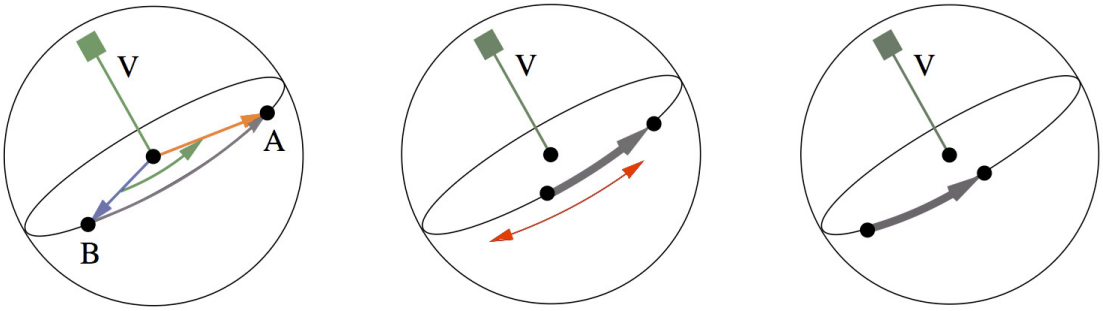
Figure 3.10: a) A scalar, quotient of parallel vectors. The quotient of their magnitudes is a tensor. b) Versor, quotient of two nonparallel vectors.

composition of two quaternions. From this follows that a quaternion $\overrightarrow{Q}$ can always be built of a tensor $\mathbf{T}$ and a versor $\mathbf{V}$:

$$\overrightarrow{Q} = \mathbf{T} \diamond \mathbf{V} \tag{3.6}$$

The quaternion needs four parameters for its numerical representation. The versor claims three of them and the tensor one. If we assume that our vectors are normalized, which means that their magnitudes are the same and equal to one, we need only three parameters to describe a quaternion. This means that this quaternion consists only of a versor.

A geometrical illustration in Figure 3.11 helps to intuitively understand the meaning of a versor. We assume that we have two normalized vectors $\overrightarrow{A}$ and $\overrightarrow{B}$. A versor has always a direction and a norm. The direction is orthogonal to the plane defined by the



(a) (b) (c)

Figure 3.11: A versor, represented as a directed arc on the surface of an unit sphere.

vectors $\overrightarrow{A}$ and $\overrightarrow{B}$. The angle to map vector $\overrightarrow{B}$ into vector $\overrightarrow{A}$ along the axis, formed by the direction of the versor, is a function of the norm. However the norm can never exceed the value of one. Literally, the norm stays always inside the unit circle. If the norm is equal one, it means that the vector $\overrightarrow{A}$ and $\overrightarrow{B}$ are pointing into opposite directions.
The versor can also be seen as a directed arc traced on the sphere surface. It is important to mention that the arc can be moved freely along the circle, formed by the intersection of the unit sphere and the plane defined by the vector $\overrightarrow{A}$ and $\overrightarrow{B}$, without losing its identity. The vectors can be freely translated as well, without losing their identity.

3.3.4 Versor Optimization

Due to the fact that we are using a gradient descent optimizer, an iterative approach, it is very important to understand specific optimization characteristics related with versors. Let's assume we have a set of parameters $\mathbf{u}$ to be optimized. A usual update step, performed by the gradient descent optimizer looks like this:

$$\mathbf{u}' = \mathbf{u} + \lambda \Delta \mathbf{u} \tag{3.7}$$

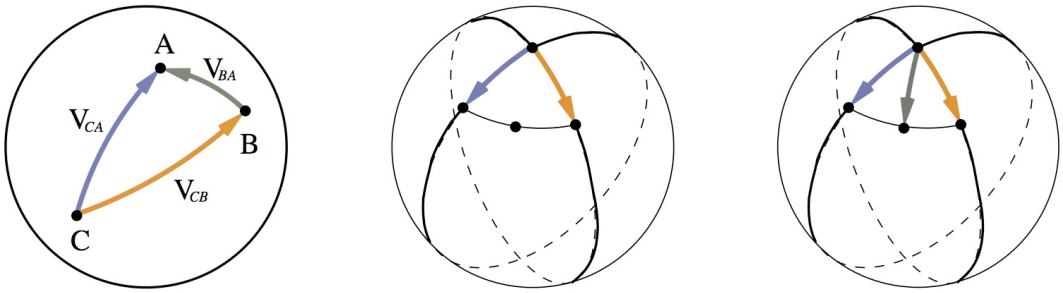
where $\mathbf{u}$ are the current parameters, $\mathbf{u}'$ the new, updated, parameters, $\Delta\mathbf{u}$ the derivation of a cost function with respect to the parameters $\mathbf{u}$ and λ the step size.
The space of versor parameters is not a vector space. Adding the components of two versors does not result in another versor. In order to combine two versors we must use

the operation of versor composition, instead of the simple addition of their components. Compositions of versors can be represented very intuitively using directed arcs on the surface of a unit sphere which is shown in Figure 3.3.4. Consider we have three given vectors ($\overrightarrow{A}$, $\overrightarrow{B}$ and $\overrightarrow{C}$). The points in Figure 3.3.4 a) are showing the intersection points between the unit sphere and these vectors. There are also three versors ($\mathbf{V}_{CB}$, $\mathbf{V}_{BA}$ and $\mathbf{V}_{CA}$) which transfers one vector into another:

$$\begin{aligned} \mathbf{V}_{CB} &= \overrightarrow{B}/\overrightarrow{C} \\ \mathbf{V}_{BA} &= \overrightarrow{A}/\overrightarrow{B} \\ \mathbf{V}_{CA} &= \overrightarrow{A}/\overrightarrow{C} \end{aligned} \tag{3.8}$$

where $\mathbf{V}_{CA}$ is the resulting composition.

$$\mathbf{V}_{CA} = \mathbf{V}_{BA} \diamond \mathbf{V}_{CB} \tag{3.9}$$



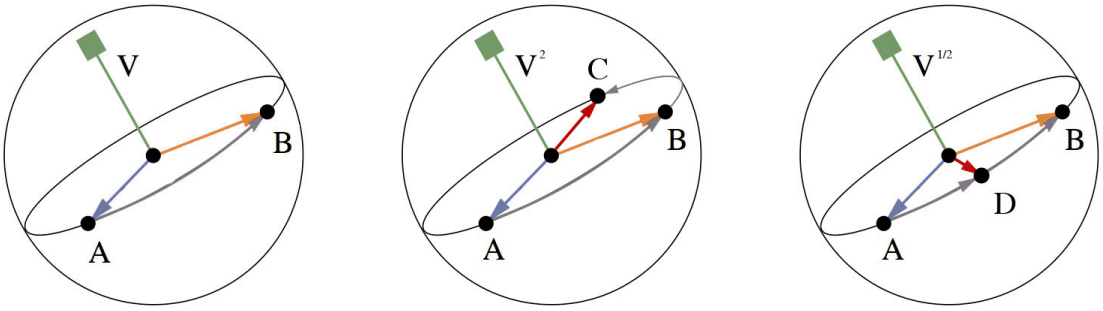
(a) (b) (c)

Figure 3.12: a) demonstration of a versor composition using directed graphs on the surface of a unit square. b) and c) are showing the versor addition operation.

Another important issue is to understand how versors and the angle they represent correlate. Thereby we have to look at the outcomes of versor exponentiation starting by the definition of the square of a versor:

$$\mathbf{V}^2 = \mathbf{V} \diamond \mathbf{V} \tag{3.10}$$

The operator $\mathbf{V}^2$ can be considered as a repeated operation of the versor $\mathbf{V}$. As we mentioned before, a versor can be considered as a directed arc. In Figure 3.3.4 a) is shown an illustration of a versor $\mathbf{V}$, which transforms vector $\overrightarrow{A}$ into vector $\overrightarrow{B}$. If we apply the same versor on the result we get vector $\overrightarrow{C}$. We can achieve this also by applying the squared operator $\mathbf{V}$ on the vector $\overrightarrow{A}$:



(a) (b) (c)

Figure 3.13: Versor exponentiation. The square of a versor is a versor with a doubled angle. A square root of a versor is a versor with halved angle.

$$\begin{aligned} \overrightarrow{B} &= \mathbf{V} \star \overrightarrow{A} && (3.11) \\ \overrightarrow{C} &= \mathbf{V} \star \overrightarrow{B} \\ &= \mathbf{V} \diamond \mathbf{V} \star \overrightarrow{A} \\ &= \mathbf{V}^2 \star \overrightarrow{A} && (3.12) \end{aligned}$$

The same assumption applies to the square root operator, shown in Figure 3.3.4 c):

$$\begin{aligned} \mathbf{V}^{1/2} &= \mathbf{R} \\ \mathbf{V} &= \mathbf{R} \diamond \mathbf{R} && (3.13) \end{aligned}$$

Now, the relation between the versor operator $\mathbf{V}$ and the resulting rotation angle θ becomes clear.

$$\begin{aligned} \Theta(\mathbf{V}) &= \theta \\ \Theta(\mathbf{V}^n) &= n\,\theta && (3.14) \end{aligned}$$

where $\Theta(\mathbf{V})$ is a function which evaluates the angle of the versor.
The gradient descent update step will not include a versor addition, because this is the wrong operation here (Figure 3.3.4). Instead, the operation which corresponds to a vector addition is the versor composition $\diamond$. An update step is achieved by composing the current versor with another versor, which represents the small variation.

$$\mathbf{V}' = d\mathbf{V} \diamond \mathbf{V} \tag{3.15}$$

where $\mathbf{V}$ is the current versor, $\mathbf{V}'$ the resulting versor after the optimization step and $d\mathbf{V}$ the versor increment computed by the gradient descent optimizer. The versor increment is defined as:

$$d\mathbf{V} = \left[\frac{\partial \mathbf{S}(\mathbf{V})}{\partial \mathbf{V}}\right]^{\lambda} \tag{3.16}$$

where $\mathbf{S}(\mathbf{V})$ represents the cost function. The meaning of this expression is how much will the cost function $\mathbf{S}(\mathbf{V})$ change with a variation of the angle and direction of $\mathbf{V}$.
It is important to notice that that the increment is not multiplied by the step length λ but raised to the power of λ. This follows from the previous discussed matter that a multiplication of an angle by a scalar equals to a exponentiation in the versor space.

3.3.5 Rigid Transform in 3D

A robust rotation model, which sucessfully represents rotation around the origin, has been presented in the previous chapter. In order to represent rigid transformations it is necessary to extend this model to capture also translations. To do so we introduce a new parameter $\mathbf{C}$ which represents a fixed point. It consists of three elements: x-axis coordinate, y-axis coordinate and z-axis coordinate.

$$\mathbf{P}' = \mathbf{V} \star (\mathbf{P} - \mathbf{C}) + \mathbf{C} \tag{3.17}$$

where $\mathbf{P}$ is the current point set, $\mathbf{P}'$ the updated point set and $\mathbf{V}$ the versor. We can convert the previous equation into another one, where the rotation components are separated from

the translation ones:

$$\begin{aligned} \mathbf{P}' &= \mathbf{V} \star \mathbf{P} + [\mathbf{C} - \mathbf{V} \star \mathbf{C}] \\ \mathbf{P}' &= \mathbf{V} \star \mathbf{P} + \mathbf{T} \\ \mathbf{T} &= \mathbf{C} - \mathbf{V} \star \mathbf{C} \end{aligned} \tag{3.18}$$

This means that we first transform our object to the origin and then perform the versor rotation. Accounting to Equation 3.16 translation and rotation are de-correlated. Hence updates of the fixed point set **C** can be treated independently using linear expressions. However this applies just in theory, because it presumes that the correct fixed point **C** has already been found. In practice we are searching for the optimal fixed point and the optimal versor in a six dimensional space at the same time.

3.3.6 Interpolation

During the registration process we have to apply an interpolation to the moving image many times and compare it with the fixed image using a similarity measure. Therefore it is necessary to consider possible obstacles which can occur during an interpolation step. Interpolation errors can introduce modulations in the similarity measure as shown by Pluim et al [29]. This periodic modulation can produce local optima and lead to an incorrect registration solution. This is especially valid for transformations such as translations with unequal sample spacing.
However the computational cost of "correct" interpolation, Hajnal [13], is too high for an iterative approach, hence a a lower cost interpolation techniques must be used. There are many possible interpolation types that can be used. We use the trilinear interpolation method because it gives the best trade-of between accuracy and runtime for our iterative registration approach. This is also the most common interpolation method in current medical image analysis applications.

3.3.7 Metric

Registration using voxel similarity measure (or metric) involves calculating the registration transformation by optimizing some measure directly from the voxel intensity values.

We used the sum of squared differences (SSD) as our voxel similarity measure.

$$SSD = \frac{1}{N} \sum_{\mathbf{x}_F} |I_F(\mathbf{x}_F) - I_M(\rho(\mathbf{x}_F))|^2 \tag{3.19}$$

where I_F represents the fixed image, I_M the moving image and $\rho(\mathbf{x}_F)$ the transformation of the fixed image points. The SSD has to be normalized by the number of voxels N that lie in the overlap domain of I_F and I_M with respect to the transformation ρ.
Due to the fact that we are dealing only with data sets acquired with the same modality, CT scans, we do not have to use information theoretic similarity measures as joint entropy or mutal information.

3.4 Implementation

The implementation was done using the Insight Segmentation and Registration Toolkit [1]. ITK is a cross-platform development framework for many medical image processing applications. The software is implemented in C++.
The registration toolbox, a part of the ITK framework, is designed to be very modular. Particular components can be easily assigned and exchanged. However there a some limitations in order that particular transformations require specific optimizers. Constructing multi-resolution registration approaches is also simplified through the framework.
As the transformation model we used the Versor 3D Transformation and the associated optimizer, the Versor 3D Optimizer.

3.5 Evaluation Experiments & Results

To benchmark our intensity-based registration approach, evaluations on synthetic and clinical dental data were performed. The calculations were performed on an Apple MacBook (notebook) with a 1.8 GHz Core2-Duo CPU and 2 GB of main memory. The computer runs the 32-bit Mac OSX Leopard operating system. We used the 3.4 version of the ITK framework and performed registration using a multi-resolution approach. Our image pyramid consists of four layers. The registration is started on the coarsest layer. The solution on the coarsest layer is propagated to the next finer level and used as its initialization. A common registration process takes about two minutes.

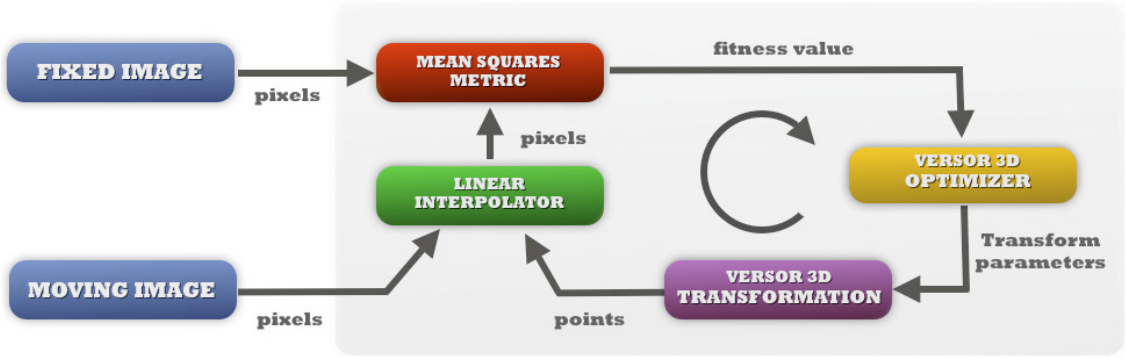


Figure 3.14: Details of our registration framework implemented using the ITK [1] framework.

3.5.1 Synthetic Data

Synthetic experiments were performed on one maxilla data set, taken one year after the dental operation. Therefore augmented bone material is shown in the scan. The CT data set has a volume size of $(512 \times 512 \times 37)$ voxels with $(0.21_{mm} \times 0.21_{mm} \times 1_{mm})$ spacings. We simulated just rotation movements. The manual masking step of the maxilla and the initialization step afterwards where two data sets are pre-aligned with regard to their geometric centers makes it impossible to establish noticeable translation displacement of the two data sets. To evaluate the robustness of our registration method, we simulated noise by adding gaussian noise to the CT data.

3.5.1.1 Rotation movements

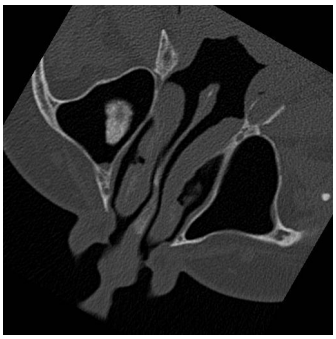
We rotated the data set along the x, y and z axis. All rotations were done in 5^o steps from 5^o till 35^o. We measured the SSD at the initialization of the registration process and after the registration process was finished. We also compared the estimated angles with the correct ones.

3.5.1.2 Gaussian noise

We used one other clinical data set with $(512 \times 512 \times 40)$ voxels and $(0.21_{mm} \times 0.21_{mm} \times 1_{mm})$ spacings. The rotation was held constant at 15^o. We added gaussian noise to both, the fixed image and the moving image. The intensity of the noise was 5%, 10%, 20% and 30%.

(a) demonstration of z-axis rotation

(b) 10^o rotation along z-axis



(c) 30^o rotation along z-axis

Figure 3.15: Synthetic data showing rotation along the z-axis.

α_{real}	5^o	10^o	15^o	20^o	25^o	30^o	35^o
α_{est}	5.011^o	9.99^o	15.015^o	20.03^o	24.982^o	$29,98^o$	11^o
$\lvert\alpha_{est} - \alpha_{real}\rvert$	0.011^o	0.01^o	0.015^o	0.03^o	0.018^o	0.02^o	24^o
SSD_{start}	653.12	728.68	768.2	831.8	911.74	951.43	1024.8
SSD_{end}	5.07	5.08	5.17	5.3	5.43	5.86	850

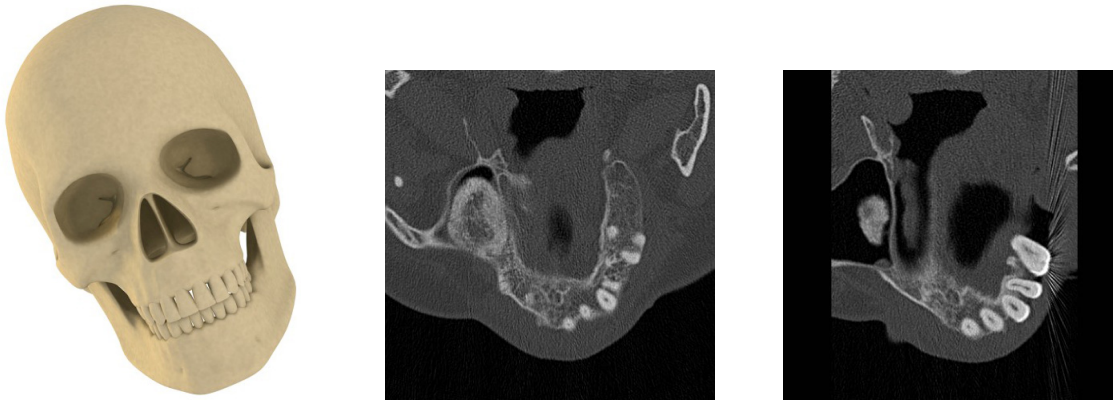
Table 3.1: Rotation along z-axis.

α_{real}	5^o	10^o	15^o	20^o	25^o	30^o	35^o
α_{est}	4.9^o	10.25^o	14.69^o	20.34^o	24.63^o	29.59^o	35.92^o
$\lvert\alpha_{est} - \alpha_{real}\rvert$	0.1^o	0.25^o	0.31^o	0.34^o	0.37^o	0.61^o	0.92^o
SSD_{start}	596.3	626.1	650.8	697.3	740.6	833.7	912.2
SSD_{end}	8.5	9.2	10.1	10.5	10.9	12.3	22.3

Table 3.2: Rotation along y-axis.

3.5.2 Real Data

We used `CT` data sets of four different patients. Every patient was scanned two times after the maxilla operation. Hence ground truth data was not available, evaluation is based just by visual inspection of the experts. In Figure 3.19 and 3.20 are shown six slices with difference images before the registration process (just initialisation) and after the registration procedure.



(a) demonstration of y-axis rotation

(b) 10^o rotation along y-axis

(c) 30^o rotation along y-axis

Figure 3.16: Synthetic data showing rotation along the y-axis.

(a) 10% noise

(b) 20% noise

(c) 30% noise

Figure 3.17: Synthetic data showing different percentages of added gaussian noise.

3.5.3 Conclusion

We showed that our interactive registration algorithm leads to appealing results. It is robust to gaussian noise. Our algorithm can handle Gaussian noise up to 30%. Due to the preprocessing step it is also robust in terms of outliers (overexposed structures). It can also handle large rotation displacements. If we consider that the patients position is fixed by a frame during the CT scan, the rotations between acquisitions will never exceed rotations above 30^o degrees in any direction. The experts were satisfied by the outcomes

σ_{noise}	0%	5%	10%	20%	30%
α_{est}	14.9^o	14.91^o	14.85^o	15.3^o	15.4^o
α_{real}	15^o	15^o	15^o	15^o	15^o
$\lvert\alpha_{est} - \alpha_{real}\rvert$	0.1^o	0.09^o	0.15^o	0.3^o	0.4^o
SSD_{start}	977.1	925.4	856.7	804.62	911.7
SSD_{end}	3.31	21.3	61.8	191.22	362.9

Table 3.3: Registration evaluation by adding Gaussian noise to CT data.

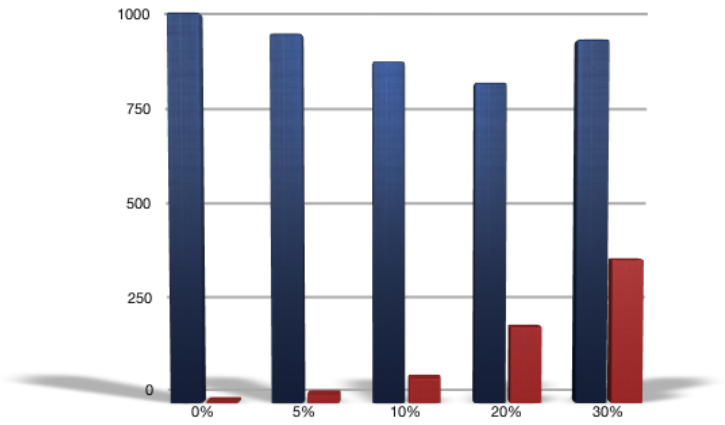
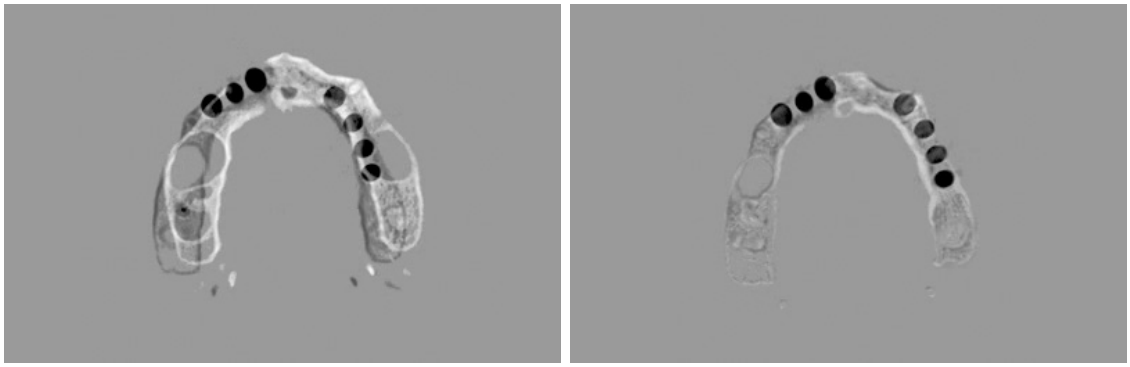


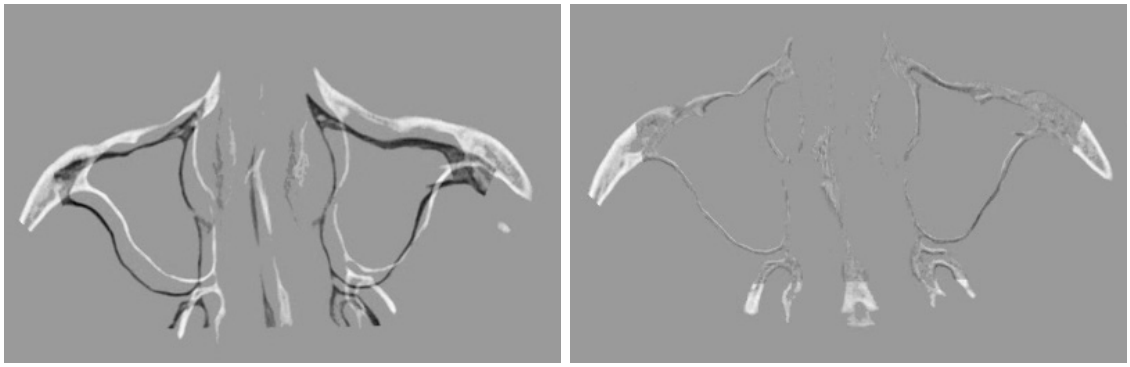
Figure 3.18: A graphical visualization of the results in table 3.3. The blue bar indicates the metric value between the fixed and moving image at the initialization and the red bar after the registration process.

of real data registration. The runtime is also acceptable if we consider that iterations are also performed on the finest level ($512 \times 512 \times 40$ voxels). The multi resolution approach is reducing the overall runtime immensely.

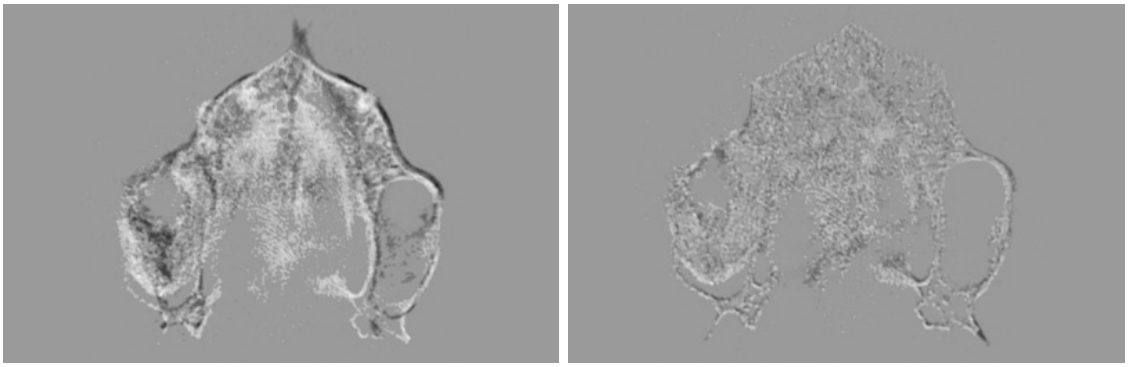
Due to the thresholding in the preprocessing step we excluded all tissues besides bones. However our augmented material has not all intensity values above 400 HU. But if we reduce the threshold our rigid registration does not lead to an appropriate solution anymore. The user interaction is another drawback of this method. This leads us to the investigation of nonlinear registration techniques. Hereby global changes as well as local changes can be modeled. Therefore a thresholding step is not required anymore. Also the independent movement of several rigid structures can be described through local transformations and does not require user interaction.



(a) before registration (b) after registration

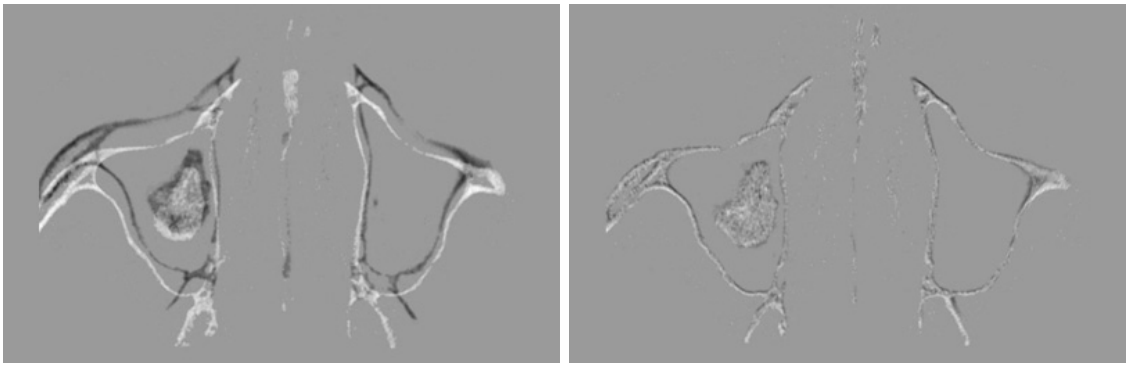


(c) before registration (d) after registration

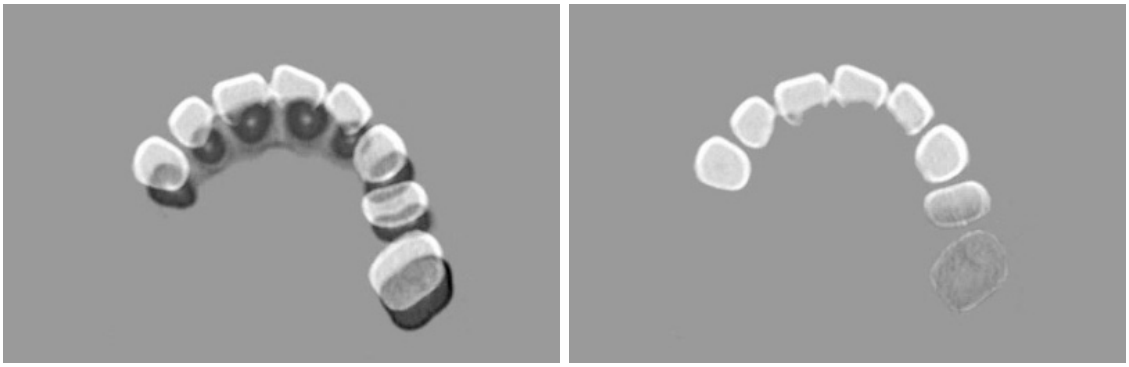


(e) before registration (f) after registration

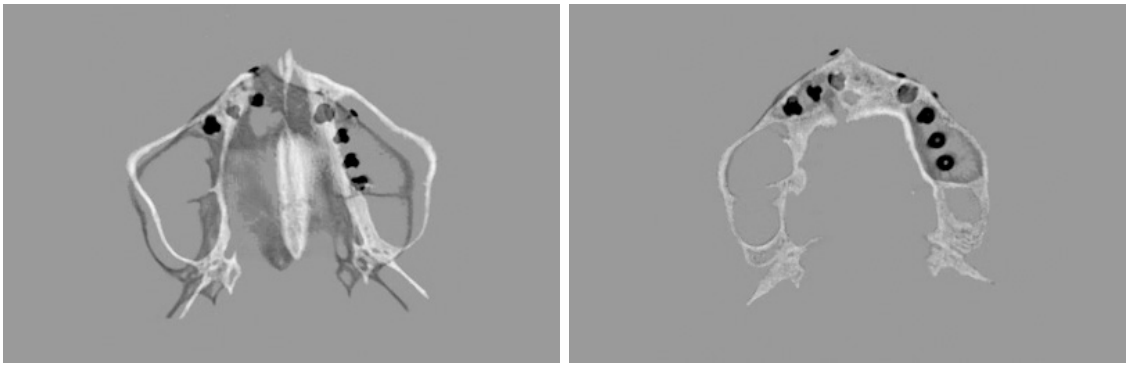
Figure 3.19: Sample slices showing the registration results on clinical CT data sets.



(a) before registration (b) after registration



(c) before registration (d) after registration



(e) before registration (f) after registration

Figure 3.20: Sample slices showing the registration results on clinical CT data sets.

Chapter 4

Affine Flow Registration

Contents

4.1 Unbiased Second-Order Prior 58

4.2 Optimization solutions 59

4.3 Implementation 62

4.4 Evaluation Experiments & Results 66

As seen in the previous chapter structures such as the cranial bone, lower-jaw and the spine, shown in the medical image data sets, form an obstacle for a rigid intensity-based registration approach. It is not possible to determine a valid global rigid transformation to map points from one data set to the other. Therefore user interaction must be brought in to overcome this problem. Another way to solve this obstacle is to change the transformation model to a nonlinear transformation model. Thereby each of these three structures (cranial bone, lower-jaw and the spine) is mapped by a displacement field from one image to the other and not by a global parametric transformation technique. An additional benefit is that we do not have to select just specific tissues for registration by a thresholding preprocessing step, we are able to align all tissues shown in the data. Therefore a nonlinear registration approach aligns the whole augmented material without excluding parts due to thresholding.

Our method is based on the work of Trobin et al [39]. They proposed a variational framework for optical flow estimation using a specific prior for flow field regularization. It does not suffer from drawbacks such as penalizing discontinuities in the displacement field, it is not sensitive to outliers and it does not tend towards biasing piecewise constant

motions in weakly textured areas. We extended the method to work in three dimensional spatial space. Therefore we had to use the work of Danielsson [12] to use a specific formulation of the second-order prior for three dimensional space.

4.1 Unbiased Second-Order Prior

The TV-L^1 method (Pock et al [32]) shown in section 2.3.1.5 shows affinity towards piecewise constant solutions in untexturized regions. Ishikava [16] showed that this behavior is not consistent with human behavior. There are several approaches in the current literature to handle this obstacle.

Danielsson [12] defined a new operator to solve this problem. They used circular harmonic functions to map the second order derivative operators into an orthogonal space. In three dimensional space, the new operator is given by:

$$\diamond = \begin{pmatrix} \sqrt{\frac{1}{6}}\left(\frac{\partial^2}{\partial_x^2}+\frac{\partial^2}{\partial_y^2}+\frac{\partial^2}{\partial_z^2}\right) \\ -\sqrt{\frac{5}{24}}\left(\frac{\partial^2}{\partial_x^2}+\frac{\partial^2}{\partial_y^2}\right)+\sqrt{\frac{5}{6}}\frac{\partial^2}{\partial_z^2} \\ \sqrt{\frac{5}{8}}\left(\frac{\partial^2}{\partial_x^2}-\frac{\partial^2}{\partial_y^2}\right) \\ \sqrt{\frac{5}{2}}\frac{\partial^2}{\partial_x \partial_y} \\ \sqrt{\frac{5}{2}}\frac{\partial^2}{\partial_x \partial_z} \\ \sqrt{\frac{5}{2}}\frac{\partial^2}{\partial_y \partial_z} \end{pmatrix} \tag{4.1}$$

The magnitude can be calculated as the Euclidian vector norm:

$$||\diamond|| = \sqrt{\frac{1}{6}\left(\frac{\partial^2}{\partial_x^2}+\frac{\partial^2}{\partial_y^2}+\frac{\partial^2}{\partial_z^2}\right)^2+\left(\sqrt{\frac{5}{24}}\left(\frac{\partial^2}{\partial_x^2}+\frac{\partial^2}{\partial_y^2}\right)+\sqrt{\frac{5}{6}}\frac{\partial^2}{\partial_z^2}\right)^2+\frac{5}{8}\left(\frac{\partial^2}{\partial_x^2}-\frac{\partial^2}{\partial_y^2}\right)^2+\frac{5}{2}\left(\left(\frac{\partial^2}{\partial_x \partial_y}\right)^2+\left(\frac{\partial^2}{\partial_x \partial_z}\right)^2+\left(\frac{\partial^2}{\partial_y \partial_z}\right)^2\right)}$$

The variational model, which will be used to estimate the flow $\mathbf{u} = (u_1, u_2, u_3)$ of the fixed image I_F and the moving image I_M is given as:

$$\min_{\mathbf{u}} \left\{ \int_\Omega ||\diamond \mathbf{u}||\ d\Omega + \lambda \int_\Omega |I_M(\mathbf{x}+\mathbf{u}(\mathbf{x})) - I_F(\mathbf{x})| \right\} \tag{4.2}$$

In order to find a solution of the variational functional, derivatives of both terms have to be determined. The second term, implying the optical flow constraint, is highly nonlinear

due to the nonlinear displacement field $\mathbf{u}$. Therefore derivatives can not be computed. To overcome this problem the nonlinear intensity profile $I_M(\mathbf{x}+\mathbf{u}(\mathbf{x}))$ is replaced by the first order Taylor approximation to linearize the problem locally.

$$I_M(\mathbf{x}+\mathbf{u}(\mathbf{x})) = I_M(\mathbf{x}+\mathbf{u}_0) + \langle \mathbf{u}-\mathbf{u}_0, \nabla I_M \rangle \tag{4.3}$$

Therefore the second term of the variational formulation will be replaced by $\varphi(\mathbf{u})$:

$$I_M(\mathbf{x}+\mathbf{u}(\mathbf{x})) - I_F = I_M(\mathbf{x}+\mathbf{u}_0) + \langle \mathbf{u}-\mathbf{u}_0, \nabla I_M \rangle - I_F = \varphi(\mathbf{u}) \tag{4.4}$$

The variational functional can thereby be formulated as:

$$\min_{\mathbf{u}} \left\{ \int_\Omega ||\diamond \mathbf{u}||\ d\Omega + \lambda \int_\Omega |\varphi(\mathbf{u})| \right\} \tag{4.5}$$

The previous functional is convex but non-strictly convex. Therefore we introduce an auxiliary variable $\mathbf{v} = (v_1, v_2, v_3)^T$ to frame the following convex functional:

$$\min_{\mathbf{u},\mathbf{v}} \left\{ \int_\Omega ||\diamond \mathbf{u}||\ d\Omega + \frac{1}{2\theta} \int_\Omega (\mathbf{u}-\mathbf{v})^2 d\Omega + \lambda \int_\Omega |\varphi(\mathbf{v})| \right\} \tag{4.6}$$

where the constant θ describes the similarity between the variables $\mathbf{u}$ and $\mathbf{v}$. Commonly θ is set to a small value and thereby forcing that $\mathbf{v}$ is a close approximation to $\mathbf{u}$. The method was introduced by Aujol et al. [2].

4.2 Optimization solutions

In consequence of the circumstances having a mimization problem with two variables $\mathbf{u}$ and $\mathbf{v}$, we have to perform an alternating optimization procedure:

1. Treat $\mathbf{v}$ as a constant and solve the following functional for $\mathbf{u}$:

$$\min_{\mathbf{u}} \left\{ \int_\Omega ||\diamond \mathbf{u}||\ d\Omega + \frac{1}{2\theta} \int_\Omega (u-v)^2 d\Omega \right\} \tag{4.7}$$

2. For fixed $\mathbf{u}$ solve the functional for $\mathbf{v}$:

$$\min_{\mathbf{v}} \left\{ \frac{1}{2\theta} \int_\Omega (\mathbf{u}-\mathbf{v})^2 d\Omega + \lambda \int_\Omega |\varphi(\mathbf{v})| \right\} \tag{4.8}$$

4.2.1 Solution of u

A major drawback to solve the equation 4.7 is that our prior norm $||\diamond\mathbf{u}||$ (equation 4.3) is not differentiable. To overcome this obstacle we use the dual formulation of the $\diamond$ norm. The dual variable was first suggested by T. Chan, G. Golub, and P. Mulet [9]. Following it was used by J. Carter [7] and A. Chambolle [8] to solve the functional in equation 2.19. A dual formulation is given by:

$$||\diamond\mathbf{u}|| = \max\{\mathbf{p} \diamond \mathbf{u} \ : \ ||\mathbf{p}|| \leq 1\} \tag{4.9}$$

where $\mathbf{p} = (p_1, p_2, p_3)^T$ is the dual variable. The approach of J. Carter [7] is used for further assumptions. Replacing $||\diamond\mathbf{u}||$ with the dual formation we get the following functional:

$$\min_{\mathbf{u}} \max_{||\mathbf{p}||\leq 1} \underbrace{\left\{ \int_\Omega \mathbf{p} \diamond \mathbf{u} + \frac{1}{2\theta} \int_\Omega (\mathbf{u} - \mathbf{v})^2 \ d\Omega \right\}}_{\Psi(\mathbf{u},\mathbf{p})} \tag{4.10}$$

We can interchange the *min* and *max* due to the facts:

1. The dual variable $\{\mathbf{p} : ||\mathbf{p}|| \leq 1\}$ is bounded.

2. The functional $\Psi(\mathbf{u}, \mathbf{p})$ in equation 4.11 is convex.

The obtained functional is:

$$\max_{||\mathbf{p}||\leq 1} \min_{\mathbf{u}} \underbrace{\left\{ \int_\Omega \mathbf{p} \diamond \mathbf{u} + \frac{1}{2\theta} \int_\Omega (\mathbf{u} - \mathbf{v})^2 \ d\Omega \right\}}_{\Psi(\mathbf{u},\mathbf{p})} \tag{4.11}$$

Chambolle showed in [8] that for a linear operator in the discrete domain the following assumption is valid:

$$\int_\Omega \mathbf{p} \diamond \mathbf{u} = \int_\Omega \diamond\mathbf{p} \ \mathbf{u} \tag{4.12}$$

The functional in equation 4.11 is changing to:

$$\max_{||\mathbf{p}||\leq 1} \min_{\mathbf{u}} \underbrace{\left\{ \int_\Omega \diamond\mathbf{p} \ \mathbf{u} + \frac{1}{2\theta} \int_\Omega (\mathbf{u} - \mathbf{v})^2 \ d\Omega \right\}}_{\Psi(\mathbf{u},\mathbf{p})} \tag{4.13}$$

$\Psi(\mathbf{u}, \mathbf{p})$ is strictly convex in the primal variable $\mathbf{u}$. Therefore $\mathbf{u}$ can be written as a function $\mathbf{v}$ by solving the first-order condition:

$$\nabla\Psi(\mathbf{u}, \mathbf{p}) = 0 \quad \Longleftrightarrow \quad \mathbf{u} = \mathbf{v} - \theta \diamond \mathbf{p} \tag{4.14}$$

Now, the functional can be formed without the primal variable $\mathbf{u}$:

$$\max_{||\mathbf{p}||\leq 1} \underbrace{\left\{ \int_\Omega \diamond\mathbf{p} \ (\mathbf{v} - \theta \diamond \mathbf{p}) + \frac{1}{2\theta} \int_\Omega (-\theta \diamond \mathbf{p})^2 \ d\Omega \right\}}_{\Gamma(\mathbf{u})} \tag{4.15}$$

We can form a minimization functional out of the previous by changing the algebraic sign:

$$\min_{||\mathbf{p}||\leq 1} \left\{ - \int_\Omega \diamond\mathbf{p} \ \mathbf{v} + \frac{\theta}{2} (\diamond \mathbf{p})^2 \, d\Omega \right\} \tag{4.16}$$

Using the formulation of Equation 4.15 we get the functional shown in [39]:

$$\min_{||\mathbf{p}||\leq 1} \left\{ - \int_\Omega \mathbf{p} \ \diamond \mathbf{v} + \frac{\theta}{2} (\diamond \mathbf{p})^2 \, d\Omega \right\} \tag{4.17}$$

The Euler Lagrange equation of the variational functional is:

$$- \diamond (\mathbf{v} - \theta \diamond \mathbf{p}) = 0 \qquad ||\mathbf{p}|| \leq 1 \tag{4.18}$$

The solution can be obtained by a gradient descent approach. A gradient descent step is defined as:

$$\widetilde{\mathbf{p}}^{k+1} = \widetilde{\mathbf{p}}^k + \frac{\tau}{\theta} [\diamond (\mathbf{v} - \theta \diamond \mathbf{p})] \tag{4.19}$$

The reprojection of the dual variable $\mathbf{p}$ is given by:

$$\mathbf{p}^{k+1} = \frac{\widetilde{\mathbf{p}}^{k+1}}{\max\{1, ||\widetilde{\mathbf{p}}^{k+1}||\}} \tag{4.20}$$

4.2.2 Solution of v

The Euler Lagrange equation of Equation 4.8 is:

$$\mathbf{v} - \mathbf{u} + \lambda\theta \frac{\varphi(\mathbf{v})}{|\varphi(\mathbf{v})|} \nabla I_M = 0 \tag{4.21}$$

where the first order image residual $\varphi(\mathbf{v})$ is defined as:

$$\varphi(\mathbf{v}) = I_M(\mathbf{x} + \mathbf{u}_0) + \langle \mathbf{v} - \mathbf{u}_0, \nabla I_M \rangle - I_F \tag{4.22}$$

The image residual is just valid for small variations of $\mathbf{u}$ and $\mathbf{v}$. Another problem arises when $|\varphi(\mathbf{v})|$ vanishes. To overcome these problems the following thresholding schema is applied:

$$\mathbf{v} = \mathbf{u} + \begin{cases} \lambda\,\theta\,\nabla I_M & \text{if} \quad \varphi(\mathbf{u}) < -\lambda\,\theta\,\left|\nabla I_M\right|^2 \\ -\lambda\,\theta\,\nabla I_M & \text{if} \quad \varphi(\mathbf{u}) > \lambda\,\theta\,\left|\nabla I_M\right|^2 \\ \varphi(\mathbf{u})\nabla I_M / \left|\nabla I_M\right|^2 & \text{if} \quad |\varphi(\mathbf{u})| \leq \lambda\,\theta\,\left|\nabla I_M\right|^2 \end{cases} \tag{4.23}$$

Hereby the image residual $\varphi(\mathbf{v})$ is allowed to vanish if the step from $\mathbf{u}$ to $\mathbf{v}$ is sufficiently small. Otherwise $\mathbf{v}$ makes a bounded step from $\mathbf{u}$ to decrease the magnitude of the residual $\varphi(\mathbf{v})$.

4.3 Implementation

This section will show the numerical schema of the previously described method and the specific issues associated with the implementation on the GPU.

4.3.1 Numerical Scheme

The functional in equation 4.2 is not convex. After image intensity linearization (equation 4.3) done by the Taylor approximation it becomes convex. The linearization is just valid for small displacements. In order to allow also valid estimations of larger displacements a multi resolution approach is applied. Therefore we build an image pyramid of N levels. The level 1 is estimated by down-sampling the image at level 0 by the factor two.
We solve the solution of Equation 4.2 at the level N (coarsest level). Then the solution (displacement field $\mathbf{u}$) is propagated to the next finer level. It is important to do the down-sampling of the image and the up-sampling of the displacement field complementarily. The propagated displacement field $\mathbf{u}$ is used to to compute the coefficients of the residual function φ by sampling I_M and I_F. ∇I_M is approximated by central differences. In each iteration on a pyramid level, $\mathbf{v}$ is initialized with $\mathbf{u}$ and all $\mathbf{p}_d$, $d \in 1 \ldots 3$, are set to $\mathbf{0}$.

$\mathbf{u}$ is initialized with $\mathbf{0}$ at the level N (coarsest level). The multi resolution approach is accelerating the method, too. In untextured regions a good approximation can be found on the next coarser level. Therefore we need less iterations on the finer level to reach convergence.
In order to apply the minimization procedure we perform the thresholding step from equation 4.23 to update $\mathbf{v}$ and then apply the fix-point scheme to update all $\mathbf{p}_d$ and thereby also $\mathbf{u}$ [43]. In order to perform the update in a stable way we have to use forward differences to approximate the gradient operator and backward differences to approximate the divergence operator.

4.3.2 Acceleration by Graphics Processing Units

In the past GPU's (Graphics Processing Units) were solely used in computer graphics for visualization purposes. Commonly, they were considered as "black boxes" with graphic primitives (vertices) as input and a rendered image as an output. They were used to perform specialized tasks. However, powered by the strong games industry with their rapidly rising demands the development was forced intensely. To fulfill the expectations the GPU architecture had to be altered several times. In every iteration the architecture became more general, able to perform calutations beyond the graphic domain, too. With the 8-series [27] NVIDIA introduced the CUDA (Compute Unified Device Architecture) framework [28]. Now it was possible to use the GPU for any kind of parallel computing purposes. It also unburdened the algorithm designer of dealing with hardware-oriented programming by providing a standard C language interface. Moreover, CUDA provides an interface to design algorithms with a large amount of parallel threads and handles thereby scheduling and execution on the GPU on its own.
Figure 4.1 gives a brief view of the development of GPU computing power over the last six years. With current GPU's (such as the NVIDIA GeForce GTX 280) an impressive performance of 933 GFlops can be achieved. A disparity between the growth of the computational efficiency (GFlops) and memory bandwidth (GB/s) is also obvious. While the computational efficiency looks like an exponential function the memory bandwidth is just a linear. It is evident that the memory bandwidth is becoming a bottle-neck. In this case just arithmetic intensity algorithms will exploit the full potential of current GPU's. Still, there have been developed several methods to overcome or reduce the effect of the

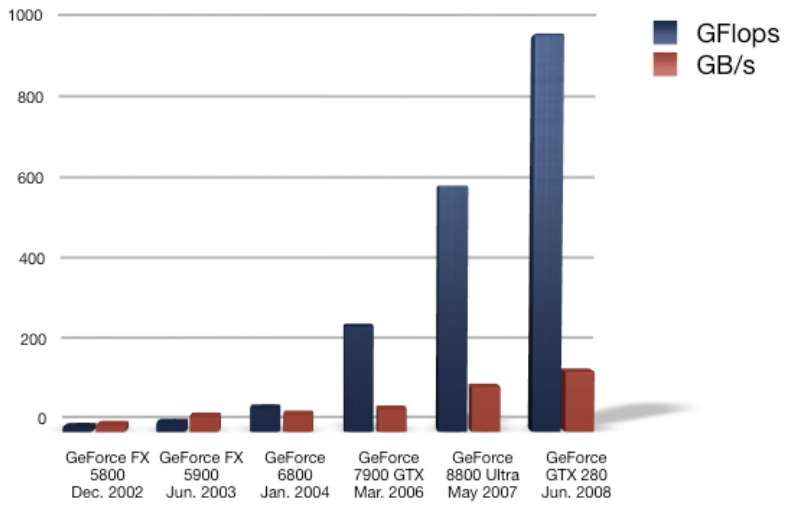


Figure 4.1: Computational efficiency (GFlops) and memory bandwidth (GB/s) for some nvidia GPU's, released during recent years.

misbalance between the growth of GFlops and GB/s.

1. Using shared memory (see Figure 4.2). A GPU consists of small arithmetic units called stream-processors (e.g. the GeForce GTX 280 has 240). They are organized into larger blocks called multiprocessors. Usually a multiprocessor consists of eight stream processors. To overcome the limitations of the device (global) memory bandwidth stream processors within a multiprocessor can use the shared memory (see Figure 4.2).
   It is important to notice that there is no direct connection between the shared memory and the device memory. Therefore the shared memory must be filled by the stream processors before starting computations.

2. Caching global device memory. Since the GeForce GTX 280 caching has been improved dramatically. Therefore it is not always the best solution to use shared memory. The overhead produced by the filling process could outweigh the benefits. This is especially valid for calculations performed on three dimensional data. Another drawback of the use is their insufficient memory size for computation of three dimensional memory blocks.

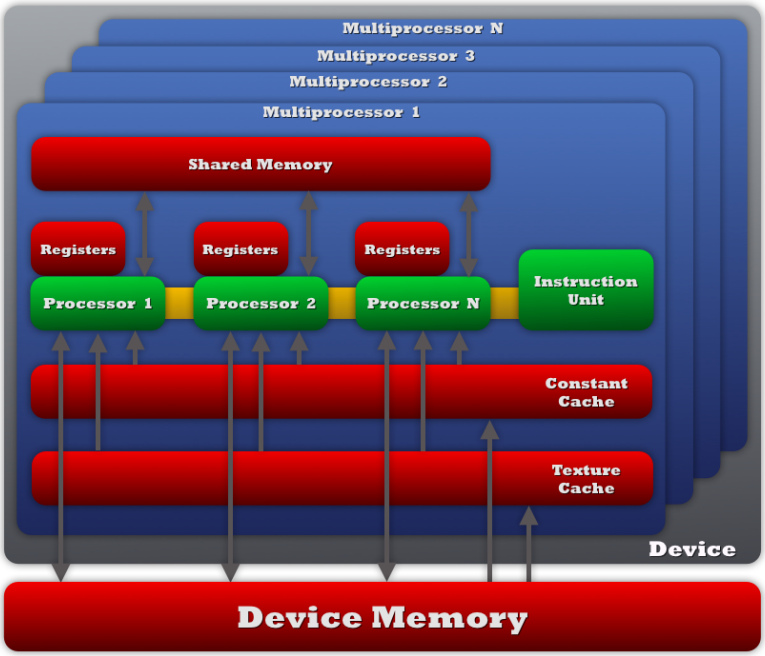


Figure 4.2: Memory architecture of modern GPU's

4.3.3 Implementation on the Graphics Processing Unit

Our variational algorithm was implemented on the GPU using the CUDA framework [28]. Except a minor preprocessing step the whole algorithm is executed on the GPU.

Due to the limited memory space on current GPU's we had to reduce the volume grid size. This is done in the preprocessing step. Therefore we used the ITK framework [1]. An interpolation in physical space is applied to reduce the volume grid to a size of maximum $(128 \times 128 \times 128)$ voxels. The spacing has to be isotropic (i.e. the same in all three directions). All the remaing steps are executed on the GPU.

During the initialization step image pyramids are built. Further we want to explain the alternating minimization schema applied on every pyramid level. Let $l = 1 \dots L$ denote the warp iteration number (of the moving image I_M using the displacement field $\mathbf{u}_0$), $n = 1 \dots N$ denotes the iteration number of the altering minimization schema and $k = 1 \dots K$ the iteration number of the projected gradient alternating descent algorithm.

1. $\mathbf{u}_0^{l+1} \leftarrow \mathbf{u}^l$

2. $I_M^{l+1} \leftarrow WA(I_M, \mathbf{u}_0^{l+1})$

3. Perform N iterations of

   (a) $\mathbf{v}^{l+1,n+1} \leftarrow TH(\mathbf{u}^{l,n}, I_M^{l+1})$

   (b) Perform K iterations of

   i. $\mathbf{u}^{l+1,n+1,k+1} \leftarrow \mathbf{v}^{l+1,n+1} + \theta \text{ div } \mathbf{p}_d^{l,k,n}$ for $d \in \{1,2,3\}$

   ii. $\widetilde{\mathbf{p}}_d^{l+1,n+1,k+1} \leftarrow \mathbf{p}_d^{l,k,n} + \frac{\tau}{\theta} \nabla \mathbf{u}^{l+1,n+1,k+1}$ for $d \in \{1,2,3\}$

   iii. $\mathbf{p}_d^{l+1,n+1,k+1} \leftarrow \frac{\widetilde{\mathbf{p}}_d^{l+1,n+1,k+1}}{\max\left\{1.0\ ,\ \widetilde{\mathbf{p}}_d^{l+1,n+1,k+1}\right\}}$ for $d \in \{1,2,3\}$

where $WA(\cdot)$ represents the warping procedure of the moving image I_M, $TH(\cdot)$ the thresholding step from Equation 4.23. It is important to mention that all variables have to be stored on the GPU to achieve the best runtime performance. We perform a fixed number of iterations on every pyramid level. Pock [31] made use of shared memory in step 3 b). Due to problems associated with shared memory and three dimensional grid data (see section 4.32) we refused this option. We achieved much higher performance on modern GPU's (up to 5× on the GTX 280) without using shared memory. As mentioned the improved caching on the GTX 280 surpasses the performance benefit of the use of shared memory (this is just valid for three dimensional grid data).

4.4 Evaluation Experiments & Results

As mentioned previously the algorithm was implemented on the GPU. Beforehand a preprocessing step has to be done. Therefore we use the ITK framework [1]. Due to the limited memory space on current GPU's we had to reduce the volume grid size. Therefore we applied a interpolation in physical space. We reduced the volume grid size to $(128 \times 128 \times 128)$ voxels.

4.4.1 Performance

In order to measure the performance of the GPU implementation we compared the runtime of our method on three different machines:

1. Apple MacBook Pro (notebook) with a 2.5 GHz Core2-Duo CPU, 4 GB of main memory and the **GeForce 8600M GT** `GPU` (512 MB) running the Mac OSX Leopard operating system.

2. personal computer equipped with a 2.13 GHz Core2-Duo CPU, 2 GB of main memory and a **GeForce 8800GTX** `GPU` (768 MB) running a 32-bit Linux operating system

3. personal computer equipped with a 2.66 GHz Quad Core2-Duo CPU, 4 GB of main memory and a **GeForce GTX 280** `GPU` (1 GB) running a 64-bit Linux operating system

We also measured the execution time using the Matlab implementation of our variational algorithm on machine 1.
We used four different isotropic volume grid sizes (16^3, 32^3, 64^3 and 128^3) and the runtime associated by them. In all cases we used the following parameters: four pyramid layers, 15 warping iterations($L = 15$), 13 alternating minimization iterations (outer iterations, $N = 13$) and 8 projected gradient descent iteration (inner iterations, $K = 8$). The results are shown in Table 4.1.

volume size (in voxels)	Matlab	8600M GT	8800GTX	GTX 280
$16 \times 16 \times 16$	168 s	10 s	3s	2 s
$32 \times 32 \times 32$	363 s	30 s	14 s	6 s
$64 \times 64 \times 64$	2354 s	187 s	72 s	12 s
$128 \times 128 \times 128$	57679 s	1502 s	548 s	59 s

Table 4.1: Execution time in seconds of our variational method performed on different machines.

4.4.2 Real Data

Due to the inability to generate appropriate synthetic data we did the evaluation just on real clinical data sets. Therefore we used three data sets. One was used from the `CT` data sets of the maxilla region, one `CT` thorax scan (provided by Martin Urschler) and one `MRI` crural (leg) data set (provided by Werner Trobin). Every data set consisted of two scans which were taken at different points in time.
Every data set was resampled to reach a maximum grid size of ($128 \times 128 \times 128$) voxels

and isotropic spacing. We compared our method (affine flow registration) with two very popular methods in current literature: TV-L^1 flow registration and the demons algorithm. The TV-L^1 flow was implemented by using the method of Pock et al [32]. We improved the run time by implementing the method on the GPU. We used the framework of Urschler [40] to perform the demons registration. More details about the two methods are presented in Chapter 2.3.1.4 and 2.3.1.5.

In order to compare the results of the three different methods we used four quantitive evaluation measures: Root-Mean-Square of Intensity Differences, Median Absolute Deviation of Intensity Differences, Normalized Mutual Information and Maximum Absolute Intensity Difference:

1. **Root-Mean-Square of Intensity Differences** RMS_{int}:
   The RMS of the intensity differences uses the pixel-wise intensity differences over the overlapping region of the image domain. It is defined as:

$$RMS_{int} = \sqrt{\frac{1}{N_1\ N_2\ N_3} \sum_{\mathbf{x}\in\Omega} \left(I_F(\mathbf{x}) - I_M(\mathbf{x})\right)^2} \tag{4.24}$$

   We use the whole image domain to calculate the RMS.

2. **Median Absolute Deviation of Intensity Differences** MAD_{int}:
   This measure is more robust in the presence of outliers as the RMS measure. The median absolute deviation is define as:

$$MAD_{int} = \text{Median}(d(\mathbf{x}))\ , \qquad d(\mathbf{x}) = |I_F(\mathbf{x}) - I_M(\mathbf{x})| > 0 \tag{4.25}$$

   We exclude all difference intensity values which are equal zero.

3. **Normalized Mutual Information** NMI_{int}:
   The normalized mutual information is a measure from information theory, which relates the information content of two images by probability distributions of their intensity values. More details are provided in Chapter 2.2.2.3.

4. **Maximum Absolute Intensity Difference** MAX_{int}:
   The maximal absolute intensity difference on the overlap region of the image grid. Therefore we use a robust maximum. It is defined as the absolute intensity difference

that is larger than 95 % of all other values.

$$MAX_{int} = \text{Max}\left(|I_F(\mathbf{x}) - I_M(\mathbf{x})|\right) \tag{4.26}$$

4.4.2.1 Compared Algorithms

In this section we want to describe the evaluation parameter setup of all three methods. For all three nonlinear registration methods (Demons, TV-L^1 Flow and Affine Flow) we used a multi resolution approach with 4 layers. The following settings were used for all three experiments:

1. **Demons**:
   60 outer iterations, 40 inner iterations, sigma = 1

2. **TV-L^1 Flow**:
   4 warp iterations, 150 outer iterations, 6 inner iterations, $\tau = 1/4$, $\lambda = 10$, $\theta = 0.1$

3. **Affine Flow**:
   4 warp iterations, 100 outer iterations, 6 inner iterations, $\tau = 3/112$, $\lambda = 30$, $\theta = 0.01$

4.4.2.2 Results

Tables 4.2, 4.3 and 4.4 are showing the results of the intensity-based measures of the three registration methods.
The illustrations in Figures 4.3, 4.4 and 4.5 are showing the axial and transversal results of the affine flow registration algorithm for all three data sets. a) is therefore showing an axial slice of the moving image $I_M(\mathbf{x})$, b) the axial slice of the fixed image $I_F(\mathbf{x})$, c) the axial slice of the warped image $I_M(\mathbf{x} + \mathbf{u})$, d) an axial slice of the absolute difference between the fixed image $I_F(\mathbf{x})$ and the moving image $I_M(\mathbf{x})$, e) an axial slice of the absolute difference between the fixed image and the warped image (after the registration), f) is showing an axial slice of the flow field of the (the blue channel is used to display the magnitude of the displacement field in x-axis direction, the green channel for the magnitude of the displacement field in y-axis direction and the red channel for the magnitude of the displacement field in z-axis direction), g)-l) are illustrating the same, just on transversal slices.

Intensity Measures			**maxilla** (upper jawbone)
RMS_{int}	*Initial*	[HU]	40.12
	Demons	[HU]	39.99
	TV-L^1 Flow	[HU]	19.4
	Affine Flow	[HU]	**18.29**
MAD_{int}	*Initial*	[HU]	0.85
	Demons	[HU]	0.82
	TV-L^1 Flow	[HU]	0.49
	Affine Flow	[HU]	**0.45**
NMI_{int}	*Initial*		0.93
	Demons		0.93
	TV-L^1 Flow		0.945
	Affine Flow		**0.95**
MAX_{int}	*Initial*	[HU]	163.36
	Demons	[HU]	89.43
	TV-L^1 Flow	[HU]	71.12
	Affine Flow	[HU]	**69.23**

Table 4.2: Clinical data evaluation results in terms of intensity-based measures on data set of the of the maxilla (upper jawbone) region.

Intensity Measures			**thorax**
RMS_{int}	*Initial*	[HU]	298.91
	Demons	[HU]	**41.51**
	TV-L^1 Flow	[HU]	67.17
	Affine Flow	[HU]	59.33
MAD_{int}	*Initial*	[HU]	257.56
	Demons	[HU]	**6**
	TV-L^1 Flow	[HU]	24.78
	Affine Flow	[HU]	14.27
NMI_{int}	*Initial*		0.88
	Demons		**0.927**
	TV-L^1 Flow		0.906
	Affine Flow		0.920
MAX_{int}	*Initial*	[HU]	794.33
	Demons	[HU]	**83.90**
	TV-L^1 Flow	[HU]	229.56
	Affine Flow	[HU]	170.35

Table 4.3: Clinical data evaluation results in terms of intensity-based measures of thoracic `CT` data sets.

Intensity Measures			**crural (leg)**
RMS_{int}	*Initial*	[HU]	1.07
	Demons	[HU]	0.79
	TV-L^1 Flow	[HU]	0.52
	Affine Flow	[HU]	**0.49**
MAD_{int}	*Initial*	[HU]	0.36
	Demons	[HU]	0.280
	TV-L^1 Flow	[HU]	0.294
	Affine Flow	[HU]	**0.270**
NMI_{int}	*Initial*		0.85
	Demons		0.87
	TV-L^1 Flow		0.86
	Affine Flow		**0.87**
MAX_{int}	*Initial*	[HU]	3.58
	Demons	[HU]	1.71
	TV-L^1 Flow	[HU]	1.52
	Affine Flow	[HU]	**1.41**

Table 4.4: Clinical data evaluation results in terms of intensity-based measures of crural `MRI` data sets.

4.4.2.3 Conclusion

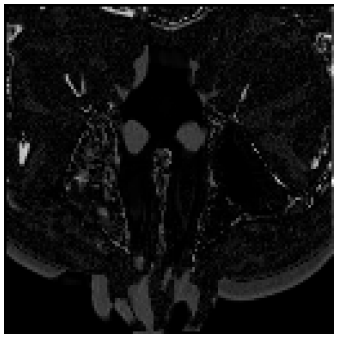
We showed that our algorithm yields to good results for all three data sets. The demons algorithm outperforms where the flow field changes are more likely to be smooth (`CT` thorax data set). However in the registration of the dental data set it fails completely. The TV-L^1 Flow yields also to acceptable results but they are not as good as those of the Affine Flow. In contrast to the Demons method both the Affine Flow and the TV-L^1 Flow are robust to outliers. An efficient hardware implementation of both methods is possible. In Figure 4.4 l) is shown that our algorithm is able to establish flow field with linear increasing flow field magnitude and does not favor piecewise constant flows like the TV-L^1 Flow.
The disadvantage of our method is its large memory requirements in order to store all v. However, to achieve maximum performance it is crucial to keep all variables inside the device (`GPU`) memory. Therefore the volume grid size has to be reduced.

(a) $I_M(\mathbf{x})$

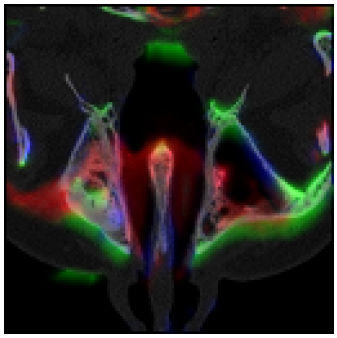
(b) $I_F(\mathbf{x})$

(c) $I_M(\mathbf{x}+\mathbf{u})$

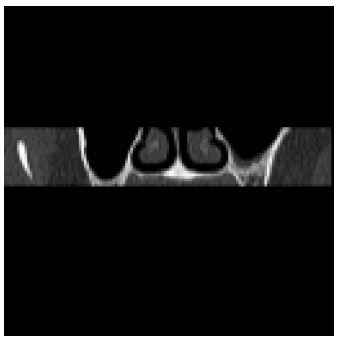
(d) $|I_F(\mathbf{x}) - I_M(\mathbf{x})|$



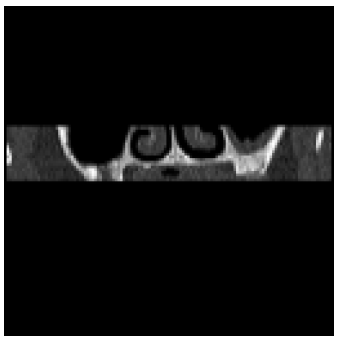
(e) $|I_F(\mathbf{x}) - I_M(\mathbf{x}+\mathbf{u})|$



(f) $\mathbf{u}$

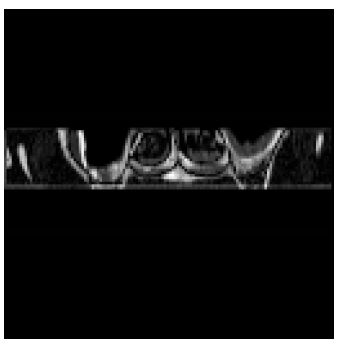


(g) $I_M(\mathbf{x})$

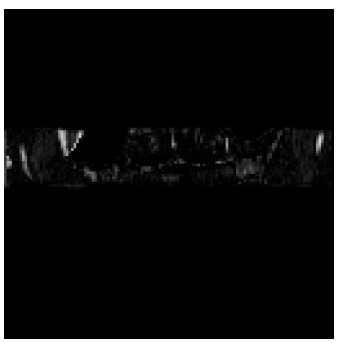


(h) $I_F(\mathbf{x})$

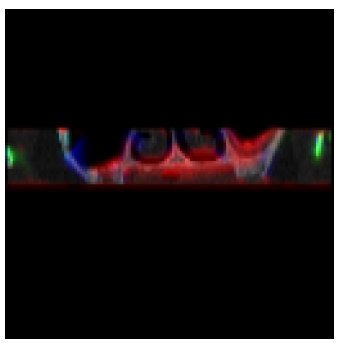
(i) $I_M(\mathbf{x}+\mathbf{u})$



(j) $|I_F(\mathbf{x}) - I_M(\mathbf{x})|$

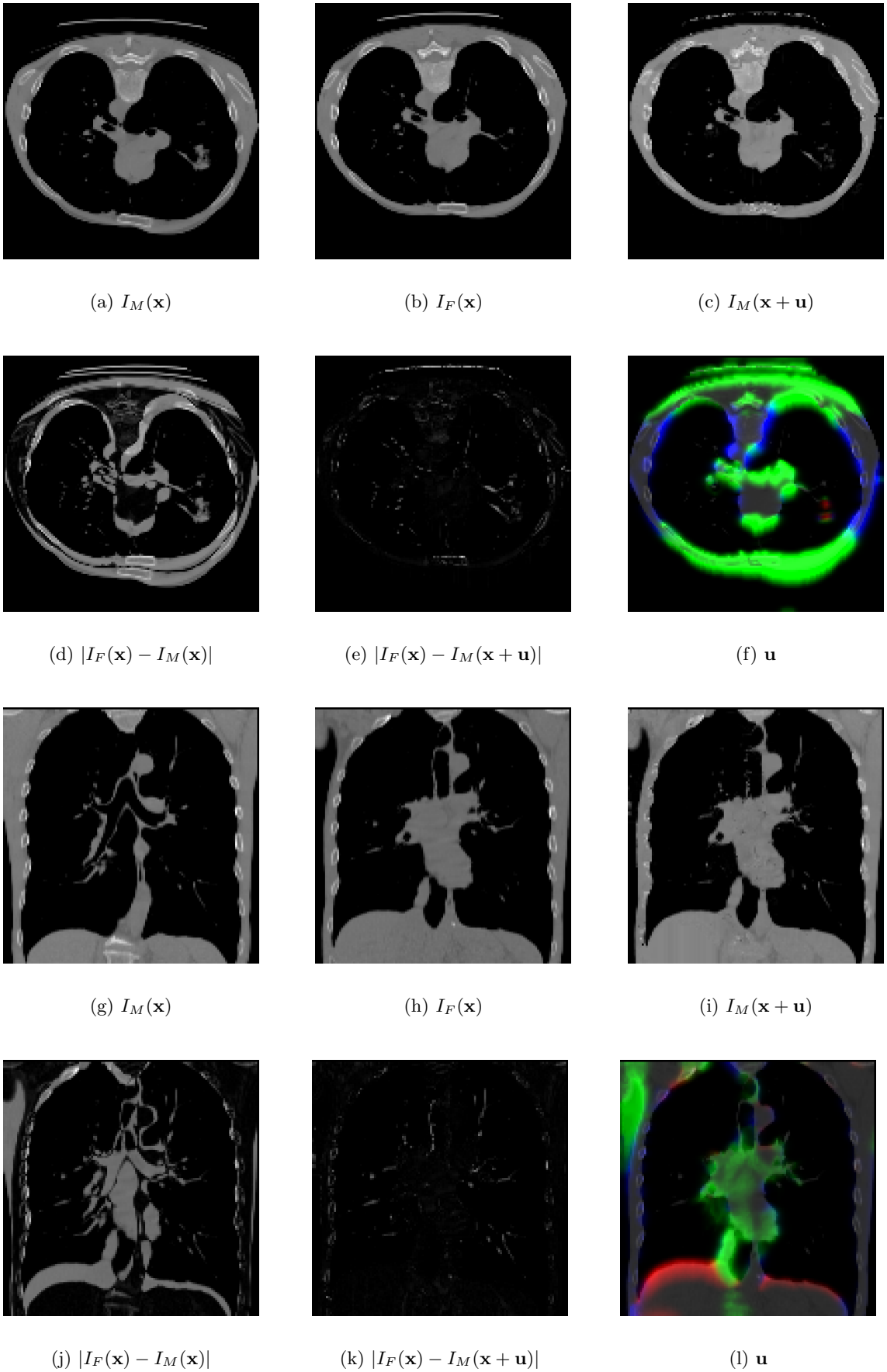


(k) $|I_F(\mathbf{x}) - I_M(\mathbf{x}+\mathbf{u})|$



(l) $\mathbf{u}$

Figure 4.3: Results of the affine flow registration on data sets of the maxilla region.



(a) $I_M(\mathbf{x})$ (b) $I_F(\mathbf{x})$ (c) $I_M(\mathbf{x}+\mathbf{u})$

(d) $|I_F(\mathbf{x}) - I_M(\mathbf{x})|$ (e) $|I_F(\mathbf{x}) - I_M(\mathbf{x}+\mathbf{u})|$ (f) $\mathbf{u}$

(g) $I_M(\mathbf{x})$ (h) $I_F(\mathbf{x})$ (i) $I_M(\mathbf{x}+\mathbf{u})$

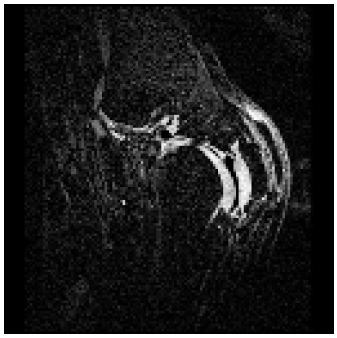
(j) $|I_F(\mathbf{x}) - I_M(\mathbf{x})|$ (k) $|I_F(\mathbf{x}) - I_M(\mathbf{x}+\mathbf{u})|$ (l) $\mathbf{u}$

Figure 4.4: Results of the affine flow registration on a clinical thoracic data set.

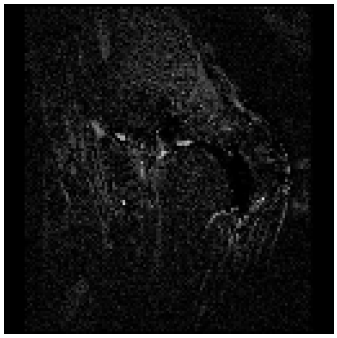
(a) $I_M(\mathbf{x})$

(b) $I_F(\mathbf{x})$

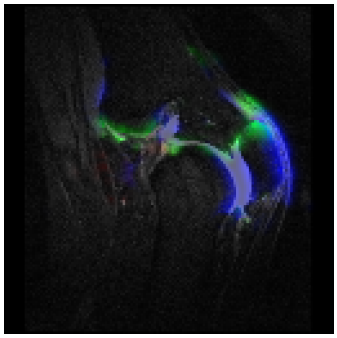
(c) $I_M(\mathbf{x}+\mathbf{u})$



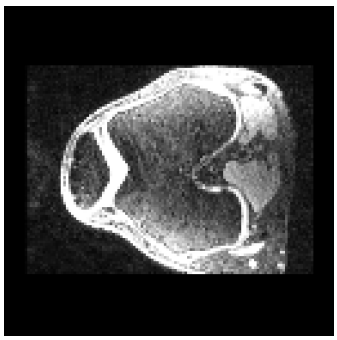
(d) $|I_F(\mathbf{x}) - I_M(\mathbf{x})|$



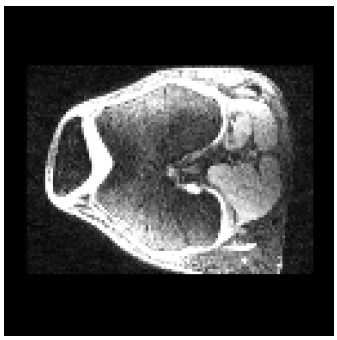
(e) $|I_F(\mathbf{x}) - I_M(\mathbf{x}+\mathbf{u})|$



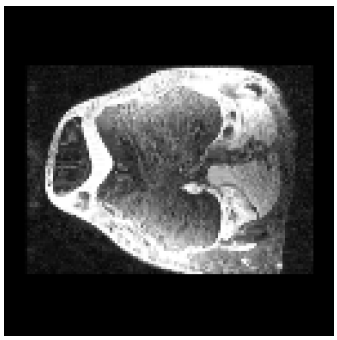
(f) $\mathbf{u}$



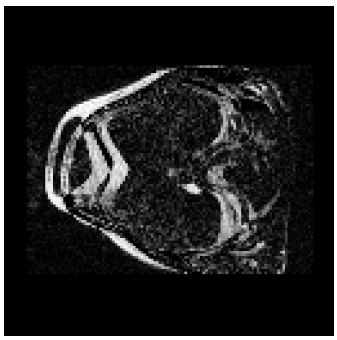
(g) $I_M(\mathbf{x})$



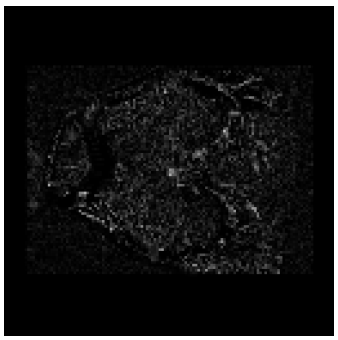
(h) $I_F(\mathbf{x})$



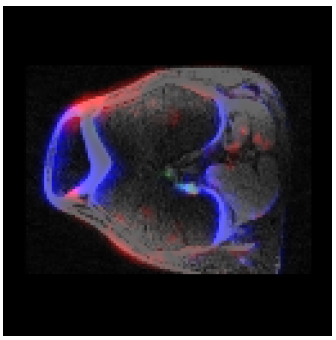
(i) $I_M(\mathbf{x}+\mathbf{u})$



(j) $|I_F(\mathbf{x}) - I_M(\mathbf{x})|$



(k) $|I_F(\mathbf{x}) - I_M(\mathbf{x}+\mathbf{u})|$



(l) $\mathbf{u}$

Figure 4.5: Results of the affine flow registration on a clinical crural (leg) data set.

Chapter 5

Conclusion and Outlook

In this thesis we presented two methods for posterior maxilla (upper jawbone) registration. The first is based on a rigid transformation model and requires manual interaction. The second uses a nonlinear registration model and does not require preselection of rigid moving structures (e.g. bones). We evaluated both methods on clinical data sets.
The Interactive Rigid Registration Algorithm performed well on clinical data sets. However it required a preprocessing step to extract only bone structures out of the data and afterwards select manually the maxilla (upper jawbone).
The Affine Flow Registration allows more freedom. It is not dependent on a preprocessing step to extract or remove specific tissues. It uses a variational formulation to estimate a solution. In contrast to other variational formulations (Horn and Schunck [15]) it allows discontinuities in the flow field. Due to the novel prior it does not favor piecewise constant flows (staircasing effect) like the method from Zach et al [43]. Due to the enormous computational effort we used a `GPU` implementation. It decreased the run time by more than 1000 times compared to the Matlab implementation.
Future work will mainly concentrate on improving the affine flow registration method. The current version of the affine flow is much more suitable to estimate local changes than global changes. This could be avoided by initializing the method with other methods which are more suitable to estimate global changes and then applying the affine flow registration. Another way to solve this issue is to improve the regularization term prior. All parameters (λ, θ) are treated as constants. As we are using a multi-resolution approach, their impact on the behavior of the algorithm is different in every pyramid layer. There-

fore an appropriate scaling strategy of these parameters could improve the performance significantly. The method should also be adapted to work with data sets acquired from different modalities, e.g. `CT-MRI`.

A strategy should also be established to generate appropriate synthetic data. This would allow to compare not just the intensity values but also the estimated flow fields of different nonlinear registration methods.

Appendix A

Acronyms and Symbols

List of Acronyms

CC	Correlation Coefficient
CT	X-ray Computed Tomography
CUDA	Compute Unified Device Architecture
DICOM	Digital Imaging and Communications in Medicine
GPU	Graphics Processing Unit
HU	Hounsfield Unit
ITK	Insight Segmentation and Registration Toolkit
MI	Mutual Information
MRI	Magnetic Resonance Imaging
NMI	Normalized Mutual Information
PDF	Probability Distribution Function
PET	Positron Emission Tomography
POI	Points of Interest
RMS	Root-Mean-Square
SAD	Sum of Absolute Differences
SPECT	Single Photon Emission Computed Tomography
SSD	Sum of Squared Differences
SVD	Singular Value Decomposition
US	Ultrasound

List of Figures

1.1 The posterior maxilla is the anatomical location where the augmented material is placed. 2
1.2 a) MRI image using T_1 mode, b) MRI using T_2 mode and c) PET scan. 4
1.3 a) Illustration of the anatomical area which is scanned and b) shows a volume rendering of a CT scan of the same area. 5
1.4 a) Demonstration of an three dimensional image grid. The dimensions are scaled with the according spatial spacing. a) is showing the whole grid where b) shows a single voxel. 6

2.1 Image registration is the task of finding a spatial transformation to map the moving image into the fixed one. 11
2.2 Illustration of a translation transformation along one axis. 12
2.3 Demonstration of an object rotated around x, y and z axis. The origin is placed in the geometric center of the object. 14
2.4 a) Illustration of a regularized displacement field and b) illustration of an unregularized displacement field. 16
2.5 An illustration of the Point-Based Alignment Process in 2D. 19
2.6 A 2D illustration of the "Head and Hat" surface alignment procedure. In this case multiple solutions are possible due to the characteristic that our anatomical structure is symmetric to rotations. 20
2.7 The basic intensity-based registration framework composed of an metric as the similarity measure, an optimization method, a transformation and an interpolation method. Taken from the ITK public-domain courseware [1]. . 22
2.8 Interpolation process during a registration procedure in 2D using physical space coordinates. Taken from the ITK public-domain courseware [1]. . . . 23
2.9 A detail of a three dimensional image grid. It demonstrates different interpolation strategies. The intensity value of point $\mathbf{C_{x,y,z}}$ has to be estimated. 24
2.10 Image pyramid showing the hierarchical multi-resolution approach. Taken from the ITK public-domain courseware [1]. 32

3.1 A survey of the Rigid Registration Model. 35

3.2 Mapping process to extract bone structures. 36
3.3 Axial view on two volume slices showing the preprocessing method. 37
3.4 Axial view on two volume slices showing noise artifacts. 37
3.5 Frontal and sagittal view of non-rigid structures in the data. 38
3.6 Result of Manualy Masking the maxilla. 39
3.7 Transversal view. 40
3.8 Volume slices of two data sets an their geometry center. 41
3.9 Our basic image registration framework. 41
3.10 a) A scalar, quotient of parallel vectors. The quotient of their magnitudes is a tensor. b) Versor, quotient of two nonparallel vectors. 43
3.11 A versor, represented as a directed arc on the surface of an unit sphere. . . 44
3.12 a) demonstration of a versor composition using directed graphs on the surface of a unit square. b) and c) are showing the versor addition operation. . 45
3.13 Versor exponentiation. The square of a versor is a versor with a doubled angle. A square root of a versor is a versor with halved angle. 46
3.14 Details of our registration framework implemented using the ITK [1] framework. 50
3.15 Synthetic data showing rotation along the z-axis. 51
3.16 Synthetic data showing rotation along the y-axis. 52
3.17 Synthetic data showing different percentages of added gaussian noise. . . . 52
3.18 A graphical visualization of the results in table 3.3. The blue bar indicates the metric value between the fixed and moving image at the initialization and the red bar after the registration process. 53
3.19 Sample slices showing the registration results on clinical CT data sets. . . . 54
3.20 Sample slices showing the registration results on clinical CT data sets. . . . 55

4.1 Computational efficiency (GFlops) and memory bandwidth (GB/s) for some nvidia GPU's, released during recent years. 64
4.2 Memory architecture of modern GPU's 65
4.3 Results of the affine flow registration on data sets of the maxilla region. . . 72
4.4 Results of the affine flow registration on a clinical thoracic data set. 73
4.5 Results of the affine flow registration on a clinical crural (leg) data set. . . . 74

List of Tables

3.1 Rotation along z-axis. 51
3.2 Rotation along y-axis. 51
3.3 Registration evaluation by adding Gaussian noise to CT data. 53

4.1 Execution time in seconds of our variational method performed on different machines. 67
4.2 Clinical data evaluation results in terms of intensity-based measures on data set of the of the maxilla (upper jawbone) region. 70
4.3 Clinical data evaluation results in terms of intensity-based measures of thoracic CT data sets. 70
4.4 Clinical data evaluation results in terms of intensity-based measures of crural MRI data sets. 71

Bibliography

[1] (2008). Insight software consortium segmentation and registration toolkit. http://www.itk.org.

[2] Aujol, J., Gilboa, G., Chan, T., and Osher, S. (2006). Structure-texture image decomposition - modeling, algorithms, and parameter selection. *Computer Vision*, 67:111–136.

[3] Baumert, B., Egli, P., Studer, S., Dehing, C., and Davis, B. (2005). Resorption accuracy of fractionated stereotactic irridation: assessment of isocenter alignement for different dental fixations by using sequential ct scanning. *Radiotherapy and Oncology*, 74:61–66.

[4] Besl, P. and McKay, N. (1992). A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256.

[5] Borgefors, G. (2003). Weighted digital distance transforms in four dimensions. *Discrete Applied Mathematics*, 125:161–176.

[6] Brown, L. G. (1992). A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376.

[7] Carter, J. L. (2001). *Dual Methods for Total Variation-Based Image Restoration*. PhD thesis, University of California, Los Angeles.

[8] Chambolle, A. (2004). An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20:89–97.

[9] Chan, T., Golub, G., and Mulet, P. (1999). A nonlinear primal- dual method for total variation-based image restoration. In *SIAM J. Sci. Comp.*, volume 20, pages 1964–1977.

[10] Chen, H. and Jain, A. (2005). Dental biometrics: Alignment and matching of dental radiographs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1319–1326.

[11] Crum, W., Hartens, T., and Hill, G. (2004). Non-rigid image registration: theory and practice. *The British Journal of Radiology*, 77:S140–S153.

[12] Danielsson, P.-E. and Lin, Q. (2001). Efficient detection of second-degree variations in 2d and 3d images. *Journal of Visual Comm. and Image Representation*, 12:255–305.

[13] Hajnal, J., Hill, D., and Hawkes, D. J., editors (2001). *Medical Image Registration*. CRC Press, Boca Raton.

[14] Hamilton, W. (1899). *Elements of Quaternions*, volume 2. Longmans Green and Company.

[15] Horn, B. and Schunck, B. (1980). Determining optical flow. *Artificial Intellignece*, 17:185–203.

[16] Ishikawa, H. (2007). Total absolute gaussian curvature for stereo prior. In *Asian Conference on Computer Vision (ACCV)*, volume 8, pages 537–548.

[17] Kamel, E., Burger, C., Buck, A., von Schulthess, G., and Goerres, G. (2003). Impact of metallic dental implants on ct-based attenuation correction in combined pet/ct scanner. *European Journal of Radiology*, 13:724–728.

[18] Labadie, R., Shan, R., Harris, S., Cetinkaya, E., Haynes, D., Fenlon, M., Jusczyk, A., Galloway, R., and Fritzpatrick, M. (2005). In vitro assessment of image-guided otologic surgery: Submillimeter accuracy within the region of the temporal bone. *American Academy of Otolaryngology-Head and Neck Surgery Foundation*, 132(3):435–442.

[19] Lanczos, C. (1986). *The Variational Principles of Mechanics*, volume 4. Dover Publications.

[20] Leung, C., Yiu, K., Zee, K., and Tsui, W. (2005). Image registration in intra-oral radiography. *Engineering in Medicine and Biology*, 27(3206-3209).

[21] Lüdemann, L., Warmuth, C., Plotkin, M., Förschler, A., Gutberlet, M., Wurst, P., and Amthauer, H. (2008). Brain tumor perfusion: Comparison of dynamic contrast enhanced magnetic resonance imaging using t1, t2, and t2* contrast, pulsed arterial spin labeling, and h215o positron emission tomography. *European Journal of Radiology*, pages 1–10.

[22] Maes, F., Collignon, A., Vandermeulen, D., Marchal, G., and Suetens, P. (1997). Multimodality image registration by maximization of mutal information. *IEEE Transactions on Medical Imaging*, 16(2):187–198.

[23] Maintz, A. and Viergever, M. (1998). A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36.

[24] Mischkowski, R., Zinser, M., Ritter, L., Neugebauer, J., Keeve, E., and Zöller, J. (2007). Intraoperative navigation in the maxillofacial area based on 3d imaging obtained by a cone-beam device. *Oral Maxillofacia Surgery*, 36:697–698.

[25] Modersitzki, J. (2004). *Numerical Methods for Image Registration*. Oxford University Press.

[26] Nikaido, A., Ito, K., Aoki, T., Kosuge, E., and Rawamata, R. (2007). A phase-based image registration algorithm for dental identification. In *IEEE International Conference on Image Processing*, volume 6, pages 229–232.

[27] NVIDIA (2006). Nvidia geforce 8800 gpu architecture overview. *Technical report, NVIDIA*.

[28] NVIDIA (2008). Nvidia cuda compute unified device architecture programming guide 2.0. *Technical report, NVIDIA*.

[29] Pluim, J., Maintz, A., and Viergever, M. (2000). Interpolation artefacts in mutal information-based image registration. *Computer Vision and Image Understanding*, 77:211–232.

[30] Pluim, J., Maintz, A., and Viergever, M. (2003). Mutual-information-based registration of medical images: A survey. *IEEE Transaction on Medical Imaging*, 22(8):986–1004.

[31] Pock, T. (2008). *Fast Total Variation for Computer Vision*. PhD thesis, Graz University of Technology.

[32] Pock, T., Urschler, M., Zach, C., Beichel, R., and Bischof, H. (2007). A duality based algorithm for tv-l1-optical-flow image registration. In Ayache, N., Ourselin, S., and

Maeder, A., editors, *Medical Image Computing and Computer-Assisted Intervention*, volume 10, pages 511–518. Springer.

[33] Rueckert, D., Aljabar, P., Heckemann, R., Hajnal, J., and Hammers, A. (2006). Diffeomorphic registration using b-splines. In *Medical Image Computing and Coputer-Assisted Intervention (MICCAI)*, volume 4191 of *LNCS*, pages 702–709. Springer.

[34] Sani, E., Veltri, M., Cagidlaco, M., Balleri, P., and Ferrari, M. (2008). Sinus membrane elevation in combination with placement of blasted implants: A 3-year case report of sinus augmentation without grafting material. *Oral and Maxillofacia Surgery*, pages 1–4.

[35] Sonka, M. and Fritzpatrick, J. M. (2000). *Handbook of Medical Imaging: Volume 2 - Medical Image Processing and Analysis*, volume 2. SPIE Press.

[36] Studholme, C., Hill, D., and Hawkes, D. (1999). An overlap invariant entropy measure of 3d medical image alignement. *Pattern Recognition*, 32:71–86.

[37] Svendsen, P., Quiding, L., and Landahl, I. (1980). Blackout and other artefacts in computed tomography caused by fillings in teeth. *Neuroradiology*, 19:229–234.

[38] Thirion, J. (1998). Image matching as a diffusion process: an analogy with maxwell's demons. *Medical Image Analysis*, 2(3):243–260.

[39] Trobin, W., Pock, T., Cremers, D., and Bischof, H. (2008). An unbiased second-order prior for high-accuracy motion estimation. In Rigoll, G., editor, *DAGM Symposium on Pattern Recognition*, volume 30, pages 396–405. Springer.

[40] Urschler, M. (2006). *Nonlinear Intra-Modality Registration of Medical Volume Data*. PhD thesis, Graz University of Technology.

[41] Wells, W., Viola, P., Atsumi, H., Nakajima, S., and Kikinis, R. (1996). Multi-modal volume registration by maximization of mutual information. *Medical Image Analysis*, 1(1):35–51.

[42] Yoo, T. (2004). *Insight Into Images*. A K Peters Ltd.

[43] Zach, C., Pock, T., and Bischof, H. (2007). A duality based approach for realtime tv-l1 optical flow. In *DAGM Symposium on Pattern Recognition*, volume 4713, pages 214–223. Springer.

[44] Zacharaki, E., Asvstas, P., Matsopoulos, G., Delibasis, K., and Nikita, K. (2001). An automatic regisration-fusion schema based on similarity measures: an application to dental imaging. In *EMBS International Conference*, volume 23.