

Optimization of registration pipeline for head MR and CT images

Karoliina Tapani

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 20.5.2016

Thesis supervisor:

Prof. Lauri Parkkonen

Thesis advisor:

D.Sc.(Tech.) Eero Salli

Author: Karoliina Tapani

Title: Optimization of registration pipeline for head MR and CT images

Date: 20.5.2016

Language: English

Number of pages: 8+74

Department of Neuroscience and Biomedical Engineering

Professorship: Biophysics and Biomedical Engineering

Supervisor: Prof. Lauri Parkkonen

Advisor: D.Sc.(Tech.) Eero Salli

In this thesis work, the aim was to find a robust, optimal rigid registration process to accurately and automatically align computed tomography (CT) and magnetic resonance (MR) images of the brain. For patients undergoing, for example, stereoelectroencephalography (epilepsy patients) or implantation of stimulating electrodes in the brain (Parkinson's patients), it is crucial to be able to combine information from low-dose CT and MR with great precision. Registration was performed with SimpleITK interface to the image registration framework of the United States National Library of Medicine Insight Segmentation and Registration Toolkit (ITK). In the optimization process an existing SimpleITK example was used as a basis for the registration algorithm, which was then optimized one block at a time beginning with the initial alignment. Registration accuracy was determined by comparing the automatic transform of our registration algorithm to the transform of a semiautomatic registration performed with a semiautomatic ITK based software, ipcWorkstation, which is used and developed in HUS Medical Imaging Center. As a result, a robust rigid registration algorithm was developed. The maximum registration errors with the final algorithm were less than 2 mm for 7 out of 15 and less than 4 mm for 12 out of 15 patients. The algorithm performs registration up to initial rotations of 45 degrees. The fast development of automated registration algorithm presented in this thesis appears promising to be used for other applications as well. This kind of block-wise optimization pattern could be used to optimize the registration either for images of other parts of the body or for other imaging modalities such as positron emission tomography (PET) and MR.

Keywords: image registration, mutual information, rigid registration, automation, magnetic resonance imaging, computed tomography

| | | |
|---|-----------------|-----------------|
| Tekijä: Karoliina Tapani | | |
| Työn nimi: Kohdennusohjelman optimointi pään magneetti- ja tietokonetomografiakuville | | |
| Päivämäärä: 20.5.2016 | Kieli: Englanti | Sivumäärä: 8+74 |
| Neurotieteen ja lääketieteellisen tekniikan laitos | | |
| Professuuri: Biologinen fysiikka ja Lääketieteellinen tekniikka | | |
| Työn valvoja: Prof. Lauri Parkkonen | | |
| Työn ohjaaja: TkT Eero Salli | | |
| <p>Tämän diplomityön tarkoituksena oli löytää optimaalinen ja automaattinen tietokonetomografia- ja magneettikuvien kohdennusmenetelmä. Kohdennus suoritettiin käyttäen hyväksi SimpleITK-ohjelmakirjastoa, joka perustuu ITK kuvakohdennus ohjelmakirjastoon (engl. the United States National Library of Medicine Insight Segmentation and Registration Toolkit). Optimointi aloitettiin SimpleITK-esimerkin pohjalta, jonka parametreja optimoitiin osa kerrallaan lähtien liikkeelle kohdennuksen alustuksesta. Kohdennustarkkuus määritettiin vertaamalla optimoidulla kohdennusohjelmalla saatua automaattista muunnosmatriisia puoliautomaattisella menetelmällä saatuun muunnosmatriisiin. Puoliautomaattinen muunnos tehtiin HUS-Kuvantamisessa kehitetyllä ipcWorkstation-ohjelmalla, joka myös perustuu ITK-ohjelmakirjastoon.</p> <p>Työn tuloksena saatiin luotettavasti toimiva jäykän kuvakohdennuksen suorittava algoritmi, joka pohjautuu SimpleITK:n Python-kirjastoon. Seitsemällä 15:stä potilaasta suurin kohdennusvirhe oli alle 2 mm ja 12:lla 15:stä potilaasta alle 4 mm. Kohdennus onnistuu jopa 45 asteen lähtökohtaisilla kulmaeroilla.</p> <p>Työssä käytettyä nopeaa algoritmikehitystekniikkaa voitaisiin käyttää optimointiin muillekin sovelluksille. Tulevaisuudessa algoritmioptimointia osa kerrallaan voisi hyödyntää kohdennusparametrien optimointiin jonkin muun vartalon alueen rakenteellisten kuvien kohdennukseen tai eri kuvamodaliteettien kohdennukseen, kuten esimerkiksi positroniemissiotomografia- ja magneettikuvien kohdennukseen.</p> | | |
| Avainsanat: kuvakohdennus, mutuaali informaatio, jäykkä kohdennus, automatisaatio, magneettikuvantaminen, tietokonetomografia | | |

Preface

This thesis work was performed at HUS Medical Imaging Center. This opportunity to work for HUS was offered by Professor and Chief Physicist Sauli Savolainen, from which I am very appreciate of. A special thanks to my instructor Eero Salli for his extensive support and advice. I wanna thank my supervisor Professor Lauri Parkkonen for great instructions and advice, which guided me throughout this thesis process. I also wanna thank other young researchers and hospital physicists at HUS Medical Imaging Center for their help. Last but not least, a great help and support was given by my fellow students, family and friends during the entire process.

Helsinki, 19.5.2016

Karoliina T. Tapani

Contents

| | |
|--|------------|
| Abstract | ii |
| Abstract (in Finnish) | iii |
| Preface | iv |
| Contents | v |
| Symbols and Abbreviations | vii |
| 1 Introduction | 1 |
| 2 Background | 3 |
| 2.1 Principles of medical image registration | 3 |
| 2.1.1 Definitions and terms | 3 |
| 2.1.2 Transformations, measures and optimization | 4 |
| 2.1.3 Interpolation | 5 |
| 2.1.4 Registration accuracy | 6 |
| 2.2 Rigid image registration applications | 6 |
| 2.3 Imaging modalities | 8 |
| 2.3.1 Magnetic resonance imaging | 8 |
| 2.3.2 Computed tomography | 10 |
| 3 Methods | 13 |
| 3.1 Insight Segmentation and Registration Toolkit: registration framework | 13 |
| 3.2 Metrics | 15 |
| 3.3 Transforms | 18 |
| 3.4 Interpolation | 19 |
| 3.5 Optimizers | 21 |
| 3.6 Multiresolution framework | 23 |
| 3.7 Initial Alignment | 24 |
| 3.8 Masking and Windowing | 24 |
| 3.9 Data | 24 |
| 3.10 Computation of registration errors and evaluation of time performance | 25 |
| 3.11 Optimization process | 27 |
| 4 Results | 29 |
| 4.1 Parameter optimization for Euler transforms | 29 |
| 4.1.1 Initial alignment | 29 |
| 4.1.2 Optimizers | 29 |
| 4.1.3 Similarity metric | 33 |
| 4.1.4 Multiresolution strategy | 33 |
| 4.1.5 Interpolators | 36 |
| 4.1.6 Number of bins and sampling | 38 |
| 4.1.7 Windowing and Masking | 39 |

| | | |
|----------|---|-----------|
| 4.1.8 | Accuracy of the algorithm with Euler transforms | 39 |
| 4.1.9 | Robustness and reproducibility | 40 |
| 4.2 | Parameter optimization for versor transforms | 44 |
| 4.3 | The final algorithm | 50 |
| 5 | Discussion | 54 |
| 5.1 | Parameter optimization | 54 |
| 5.2 | Clinical applicability | 58 |
| 5.3 | Limitations | 59 |
| 5.4 | Future prospects | 60 |
| 6 | Conclusions | 61 |
| | References | 62 |
| | Appendices | 68 |
| A | Metric evaluate -method angles | 68 |
| B | Bin number and sampling percentage combinations | 69 |
| C | Table of ITK parameters | 70 |
| D | The SimpleITK example | 74 |

Symbols and Abbreviations

Symbols

| | |
|------------------|---------------------------|
| Ω_A | image domain A |
| Ω_B | image domain B |
| $\Omega_{A,B}^T$ | intersecting image domain |
| \in | set membership |
| γ | gyromagnetic ratio |
| \hat{T} | estimate of T |
| \equiv | equivalent |
| $P^*(z)$ | Parzen window |

Operators

| | |
|---------------|------------------------------|
| T | image transformation |
| T^{-1} | inverse image transformation |
| Σ | summation |
| $\arg \max_T$ | arguments of the maxima |
| \int | integral |
| ∂ | partial derivative |

Abbreviations

| | |
|----------|---|
| AIR | automated image registration |
| ANTS | advanced normalization tools |
| CGDLS | Conjugate Gradient Descent Line Search |
| CT | computed tomography |
| FAIR | flexible algorithms for image registration |
| FLE | fiducial localization error |
| Flirt | FMRIB's Linear Image Registration Tool |
| FMRIB | Oxford Centre for Functional MRI of the Brain |
| f.r.e.e. | Flexible Registration and Evaluation Engine |
| FRE | fiducial registration error |
| GD | Gradient Descent |
| GDLS | Gradient Descent Line Search |
| ITK | National Library of Medicine Insight Segmentation and Registration Toolkit |
| JHMI | Joint Histogram Mutual Information |
| MERIT | module in MeVis Image Registration Toolkit |
| MI | mutual information |
| MITK | Medical Imaging Interaction Toolkit |
| MMI | Mattes mutual information |
| MRI | magnetic resonance imaging |
| PET | positron emission tomography |
| RSGD | Regular Step Gradient Descent |
| TRE | target registration error |
| VTK | Visualization Toolkit |
| v4 | version 4 |

1 Introduction

Due to the complementary nature of magnetic resonance (MR) and computed tomography (CT) images, it is desirable to combine the information gained from each imaging modality in a proper manner (Maintz & Viergever, 1998). Ever since the MR scanners were introduced to many hospitals and the use of different tomographic imaging modalities became routine, it has been common to take multiple tomographic images from patients such as CT images and MR images, in order to diagnose and plan the treatment (Hill et al., 2001). Multimodal image registration is needed to utilize information from both imaging modalities. Image registration consists of matching the positions of features in two images in a common coordinate space and enables comparison of the intensities of these features (Hill et al., 2001). Due to the rigid nature of the skull, especially rigid-body registration to align images of the head and brain became quite popular. A registration measure for multimodal images was first introduced by Woods and colleagues (1993). It was based on the minimization of the standard deviation of PET voxel intensities corresponding a specific value in another modality. In optimal alignment, there would be little variation in grey values.

The next big step in multimodal image registration was the idea of Collignon and colleagues (1995b) and Studholme and colleagues (1995) to use entropy as a measure. A widely used information measure is Shannon entropy (Shannon, 1948). In image registration, the Shannon entropy for joint distribution between the grey values of two images can be used as the function to be minimized (Pluim et al., 2003). Quickly after the discovery to use entropy in image registration three different groups (Collignon et al., 1995a; Wells et al., 1996; Viola & Wells III, 1997) introduced at the same time a measure based on entropy: *mutual information*.

The aim of this thesis was to optimize registration parameters for rigid registration of MR and CT images of the head. There are many studies utilizing mutual information in CT to MR rigid image registration (Collignon et al., 1995a; Wells et al., 1996; Studholme et al., 1996; Maes et al., 1997; Studholme et al., 1999; Thévenaz & Unser, 2000). They all suggest that mutual information can be used in an automated manner, since no pre-segmentation or other pre-processing is needed. Many of them also take advantage of the multiresolution strategy (Studholme et al., 1996; Wells et al., 1996; Maes et al., 1997; Thévenaz & Unser, 2000). Škerl and colleagues (2007) evaluated different similarity metrics for instance for the rigid registration of CT and MR images of the head. They suggest that one of the most accurate metrics in multimodal image registration is mutual information.

At the moment, the two patient groups, with which rigid image registration is needed at HUS (The Hospital District of Helsinki and Uusimaa) Medical Imaging Center, are the surgery candidates suffering from epilepsy as well as the Parkinson's patients undergoing the implantation of deep brain stimulators. In these patient groups, image registration is complicated compared to standard MRI-CT registration for several reasons. Firstly, CT is usually a low-dose CT revealing no soft-tissue structure within the brain. Second, electrodes are present in CT and third, they produce significant metal artefacts. Also other types of electrode implantations are

common.

At first, the epilepsy surgery candidates undergo the implantation of depth electrodes (Stereoelectroencephalography, SEEG), which are used to identify the epileptogenic focus and network (van Rooijen et al., 2013). The correct identification of the location of focus or foci is critical in order to perform successful neurosurgical treatment. Before SEEG implantation, high-resolution MR images are taken to carefully plan the insertion of electrodes. In order to utilize the location information from recordings, it is important to know the exact anatomical locations of depth electrodes. This is managed by postimplantation imaging. At HUS, the postimplantation imaging is performed with CT. These pre- and post-implantation images then need to be registered. van Rooijen and colleagues (2013) suggested that mutual information performed better compared to other measures of similarity.

Another clinical routine procedure is implantation of deep brain stimulation electrodes for Parkinson’s patients (Benabid et al., 1988; Deep-Brain Stimulation for Parkinson’s Disease Study Group and others, 2001; Odekerken et al., 2013). In order to exactly locate the electrodes delivering high-frequency electric pulses, usually implanted in the basal ganglia, accurate image registration is needed (D’Albis et al., 2015). As with epilepsy patients, the planning of electrode trajectories for surgery is conducted based on preoperative MR images.

This thesis work was conducted to meet the needs of HUS. The registration parameters were therefore optimized for the patient groups described above. In this thesis, we used SimpleITK (Lowekamp et al., 2013) image registration library, which is based on the National Library of Medicine Insight Segmentation and Registration Toolkit (ITK) (Avants et al., 2014a) registration framework v4. To our knowledge, no optimization study for version 4 (v4) has been published in the literature yet, which would have considered the difficulties in registration resulting from low-dose CT and electrode artefacts. There are multiple image registration applications, some implemented on the basis of ITK algorithms (Klein et al., 2010; Wolf et al., 2005; Pieper et al., 2004; Avants et al., 2009; Fedorov et al., 2012) and others not (Modersitzki, 2009; Boehler et al., 2011; Dickhaus et al., 2004; Floca & Dickhaus, 2007; Jenkinson & Smith, 2001; Woods, Grafton, Holmes, et al., 1998; Woods, Grafton, Watson, et al., 1998). However, they all offer registration options and since there are many parameters and in each group multiple options, for a user it would require thorough knowledge of the parameters and their relations. Therefore, it would be desirable to have a robust registration algorithm, which would not need the user to define the registration parameters, so that only the final results would need to be checked by the user.

2 Background

The aim of this thesis was to find optimal image registration method for the rigid registration of MR and CT images of the head within the ITK v4 registration framework. Therefore, the basics of medical image registration as well as MR and CT imaging are introduced in the following sections.

The last chapter introduces some available open-source rigid image registration applications, from which a few are largely based on the ITK algorithms and modules and some have developed their own. From the wide usage of ITK algorithms it is obvious, that it dominates the medical image registration field at the moment.

2.1 Principles of medical image registration

The overall idea of image registration is presented in this section, which is mostly based on the publication by Hill and colleagues (2001).

2.1.1 Definitions and terms

The term *registration* consists of two parts. On the one hand, it means matching the positions of a feature in two images into one image or coordinate space. On the other hand, in addition to relating two positions of one feature, it allows us to consider the intensities of these features from two different images. With this interpretation, the terms interpolation and resampling have to be taken into account. The registration transformation can be called mapping, when the geometry of registration is considered. In addition to spatial mapping, which is considered the transformation of a position x from one image to another, overlaying the two images requires knowledge about resolution and image sampling. Since in medical imaging the two images are from a real object, they have limited fields of view, which might quite likely be different in size. If the two images are thought to be mappings of points in the patient within their fields of view (i.e. image domains Ω), the intensity values in those locations x_A and x_B can be presented as follows;

$$\begin{aligned} A : x_A \in \Omega_A &\mapsto A(x_A) \\ B : x_B \in \Omega_B &\mapsto B(x_B) . \end{aligned} \tag{1}$$

To overlap and compare two images A and B, we need to recover the spatial transform T , which maps the locations x_A and x_B of one object in two images into locations in overlap image domain $\Omega_{A,B}^T$. This overlap domain can be defined as,

$$\Omega_{A,B}^T = \{x_A \in \Omega_A \mid T^{-1}(x_A) \in \Omega_B\} . \tag{2}$$

The image transformation T maps both position and intensity of a point in two images from one image to another, also taking into account issues regarding sampling and resolution. Since the images have a discrete nature, interpolation between sample positions is needed, as well as the information about differences in sample spacing. Therefore, transformation T has to take into account the discrete sampling.

Image fusion is a related but different operation. When image registration means determination of the mapping function between two images, image fusion is the process of combining information from two or more images into a single image. Naturally, the image registration is a prerequisite for a meaningful image fusion.

2.1.2 Transformations, measures and optimization

In registration of rigid structures such as bony structures without joints, the transformation needed is also rigid. This means that no scaling, shearing or non-affine transformations are needed in order to align two images. Since the optimization problem in this thesis involves intrasubject registration of multimodal images of the head, rigid-body registration can be used. Transformations can be categorized according to the number of parameters introduced, i.e. the degrees of freedom (Pluim et al., 2003). In rigid-body registration, there are six degrees of freedom (e.g. six parameters). In a 3-dimensional image transformation, there are three translation and three rotation parameters. It can be regarded as a special case of an affine transformation, where the scaling values are unity and skews zero. In registration of more deformable structures, such as pelvis, more degrees of freedom are needed. Usually in these cases even affine transformations are not enough, since in addition to stretch and skew most organs deform in a more complex way. In such cases, more complicated, elastic transformations are needed.

Some medical image registration algorithms are based on identifying similar features such as points, lines or surfaces, which are present in both images. In this case, the transformation T is determined iteratively until the features of interest are aligned and the algorithm converges. Other registration algorithms are based on image intensities. In the case of multimodal image registration there is no simple relationship between intensities in images A and B. One useful technique is to maximize the amount of shared information in two images. Therefore, as images are registered correctly, we are reducing the amount of information needed to present the structures present in both images. The most used measure of information is Shannon's entropy (Shannon, 1948),

$$H = - \sum_i p_i \log p_i , \quad (3)$$

where p_i are probabilities of each symbol supplying information. Maximum entropy is reached, when each symbol has an equal probability of occurring. The joint entropy of two images is the combined information of both images. In the case of multimodal image registration, the smaller the joint entropy is compared to the sum of individual entropies of images A and B (i.e. marginal entropies) (Equation 4), the less independent the images are. In the case of completely unrelated images, the joint entropy is equal to the sum of marginal entropies.

$$H(A, B) \leq H(A) + H(B) \quad (4)$$

Joint entropy is a measure of the amount of information in two images altogether. Studholme and colleagues (1995) and Collignon and colleagues (1995a) were the first

to suggest joint-entropy-based registration for multimodal images at the conference of Information Processing in Medical Imaging in 1995. In the registration process, the algorithm tries to find the transformation T which maximizes the joint entropy. In case the images differ only by noise, the noise blurs the joint histogram of images, which leads to an increase in entropy. Due to the fact that interpolation during transformation might smooth the image and at the same time reduce noise, the sharpening of the joint histogram decreases entropy.

As a function of transform in image registration, the registration measure defines an n -dimensional function, with n degrees of freedom (Pluim et al., 2003). The transform that aligns the images is the optimum of that function. Image registration is an optimization process, in which the metric value is updated iteratively according to the transformation estimate until the algorithm converges to the user defined tolerance value. Unfortunately, there are generally multiple local optima within the parameter space. Falling to a wrong local optimum could result from incorrect interpolation or changes in image contents of the two overlapping images (Pluim et al., 2003). For instance, these changes might result from post-operative brain deformations. The number of local optima can be reduced in multiple ways, for instance by smoothing the images, using higher order interpolation or by broadening the bins of the intensity histogram.

Another important aspect of the registration function is the effect of capture range of the optimum to the choice of optimizer (Hill et al., 2001). Since in most voxel similarity measures the perfect alignment is not the global minimum but one of the local minima, a proper initial alignment is important. This reduces the probability that the transformation would escape from the right local optimum capture range to the wrong global one. The number of small optima in the parameter space can be removed by image smoothing before registration. This kind of hierarchical approach in registration is common, in which multiple resolution levels are used to both boost computation and avoid wrong local minima.

2.1.3 Interpolation

After transformation, even if the voxel dimensions are the same, the discrete image grids are not comparable, since they are not perfectly aligned. Therefore, interpolation between grid points is needed. Probably the most common interpolation method in medical image registration is trilinear interpolation. However, this method includes errors in accurate comparison of registered images, since it low-pass filters the interpolated image. Furthermore, interpolation errors can induce modulation errors to the similarity measure. These errors become less important with multiple resolution approach, where the lower resolution images are already blurred to begin with.

More accurate interpolation methods are based on sinc functions. These are for example b-spline, cubic spline and windowed sinc functions with different windowing functions such as Hamming window. These sinc-based interpolation methods are computationally somewhat intensive, thus they are not usually used in the optimization process. However, they can be used in the final resampling of, for example a CT image, as well as in the registration of CT and MR images (Johnson et al., 2015).

2.1.4 Registration accuracy

It is also important to consider the registration accuracy. In medical image registration, it is crucial to get the alignment as accurate as possible, especially when the final result is used to plan the surgical management of the patient, to estimate the effect of treatment or the progress of disease. There are many ways to examine accuracy, but maybe the most common registration error definitions are target registration error (TRE) and fiducial registration error (FRE) (Fitzpatrick, West, & Maurer Jr, 1998). These measures need either the *gold standard* for comparison, or fiducial markers, which can be either invasive or non-invasive.

TRE computation requires a gold standard transform T_g , to which the transform of the registration algorithm, T , is compared to. TRE varies with location in the image (x_A) and is computed as,

$$TRE(x_A) = |T(x_A) - T_g(x_A)|. \quad (5)$$

It is common to express TRE as a mean or maximum of the distribution. However, in most clinical cases T_g is not known, so TRE can not be used to define registration error.

If there are fiducial markers implanted in the patients brain to the target area, such as the tumour or on the surface of the skull, FRE can be used to define how well the registration is performed (Fitzpatrick, West, & Maurer Jr, 1998). FRE can be derived from the residual error in fitting of fiducial point set. However, what really matters in registration according to Fitzpatrick, West, & Maurer Jr (1998) is the TRE distribution.

In this thesis, we have carefully semi-automatically registered the CT and MR images and this transformation is then regarded as the 'gold standard'. Registration itself was performed automatically, but we manually tuned the registration parameters until a visually excellent registration was achieved throughout the brain. This is a very lengthy process which cannot be done as a clinical routine. The transform provided by our automatic algorithm was then compared to this gold standard by computing distances between point sets transformed both according to the semi-automatical transformation and according to the automatically-defined transformation. In addition to this registration error method, visual inspection is a common and in many cases the only available method to define the accuracy of registration. Nevertheless, this method is dependent on the observer, which results in possible false negatives (images that are accurately enough registered, but the observer classifies them as failures) and false positives (images that are not accurately enough registered but the observer classifies them as success).

2.2 Rigid image registration applications

In addition to ITK (Ibanez et al., 2003), there are other open-source libraries which enable rigid medical image registration, such as Automated Image Registration (AIR) (Woods, Grafton, Holmes, et al., 1998; Woods, Grafton, Watson, et al., 1998) – source code in C, ANTS (Advanced normalization tools) (Avants et al., 2009) – source code

in C++ (based on ITK), Elastix (Klein et al., 2010) – source code C++ (based on ITK), FAIR (flexible algorithms for image registration) (Modersitzki, 2009) – source code Matlab, FMRIB’s Linear Image Registration Tool (Flirt) (Jenkinson & Smith, 2001) – source code in C++, the Flexible Registration and Evaluation Engine (f.r.e.e.) (Dickhaus et al., 2004; Floca & Dickhaus, 2007) – (based on ITK), MeVis Image Registration Toolkit module, MERIT – source code C++ (Boehler et al., 2011), the Medical Imaging Interaction Toolkit (MITK) – source code in C++ (based on ITK and VTK) (Wolf et al., 2005) and 3D Slicer (Pieper et al., 2004; Fedorov et al., 2012) – mostly C++ (based on ITK and VTK (Visualization Toolkit (Schroeder, 2006))). Many of them are based on ITK algorithms and some include in addition visualization tools.

3D Slicer (Pieper et al., 2004; Fedorov et al., 2012) is one of the most important medical open-source image processing applications. It consists of multiple medical imaging algorithms based on ITK and includes registration modules, though it contains only limited number of the transforms, metrics and optimizers offered by ITK. Moreover, it does not offer the possibility to combine registration parameters freely. The Slicer registration module provides options for similarity metrics, transformation parameters and so forth. It is not directly optimized to any specific application and thus requires from the user knowledge about the image registration techniques. It provides the possibility to write ITK-based extensions, which can be then used straight from the command line or from the graphical user interface. This feature was also utilized in this thesis work.

Johnson and colleagues (2007) optimized ITK parameters for 3D multimodal rigid image registration by extending one of the ITK registration example (for version 3). They used mutual information and added masking amongst other things to the example script. This algorithm, BRAINSFit, was then adapted to the 3D Slicer module, as well as an ordinary command-line script. It was used as a standard image registration protocol at University of Iowa, at least until 2007. BRAINSFit includes optional transforms and multiple other choices for the registration. Furthermore, 3D Slicer has a land-mark based registration module, which can be used as a preprocessing step for automated registration.

Another widely used application based on ITK algorithms is Elastix (Klein et al., 2010), which has no graphical interface, but it enables user to freely combine registration parameters. Elastix supports the addition of new user-developed modules. Another command-line-operating registration framework based on ITK is the Flexible Registration and Evaluation Engine (f.r.e.e.) (Dickhaus et al., 2004; Floca & Dickhaus, 2007). Like Elastix, this application does not provide a graphical user interface. Unlike other available applications, f.r.e.e. includes automatic parameter optimization. This extension enables the user to automatically optimize some of the registration parameters for a specific application. In addition to ITK algorithms it offers other self developed components. The AIR framework (Woods, Grafton, Holmes, et al., 1998; Woods, Grafton, Watson, et al., 1998), like many others, does not have any visualization tools and since it is intended for use inside another application, it is not provided with data import or export classes.

Flirt (Jenkinson & Smith, 2001), developed in Oxford University, provides a

graphical user interface in addition to the command line based intra- and intermodal affine image registration. Users are free to choose registration parameters, such as the metric (in intermodal registration either correlation ratio or mutual information based metrics), interpolation and degrees of freedom (for rigid registration 6). Flirt utilizes a global optimization method and user is provided with the choice of how precisely the search space is iterated through in the optimization process.

The MITK is an extended application to ITK algorithms and VTK visualization methods (Wolf et al., 2005). In addition to modules offered by ITK and VTK, MITK provides the user with features relevant to interactive medical imaging software development. Unlike all of the above mentioned applications, MITK offers a graphical user interface which includes real time graphical follow up of the registration progress. MITK offers different metrics, optimizers, interpolators and transforms for user selection in rigid registration.

The FAIR software (Modersitzki, 2009) is mainly meant for educational and research purposes and not necessarily suited for clinical practice. Additionally, Boehler and colleagues (2011) suggest that its Matlab environment might limit its usage. On the contrary, MERIT software (Boehler et al., 2011) is designed for rapid image registration development and its main purpose is to be used in clinical applications. It is suggested to be robust, and it offers both rigid and non-rigid registration options. Specifically, Boehler and colleagues (2011) argue that it should be easily used by individuals without extensive knowledge about the registration parameters.

ANTS (Avants et al., 2009) is another software implemented on ITK. It is both an application-oriented toolkit and it serves as a basis for further algorithm development. In addition to being a normalization tool, it is provided with image registration tools. It enables the user to define the registration parameters from the wide selection of parameters it offers.

The Nifty Reg program developed in the University College London (Modat et al., 2010) is mainly intended for non-rigid registration, but also rigid registrations are available. It utilizes partly ITK framework to write and import data, but most of its components are independent. It does not have any graphical interface.

2.3 Imaging modalities

This thesis work aimed to optimize registration parameters for a special case, in which MR and CT images of the head are aligned. Therefore, the next two chapters introduce the basics of these two imaging modalities. In first section about magnetic resonance imaging is largely based on the book by McRobbie and colleagues (2007), and the second section about CT imaging on the book by Cierniak and colleagues (2011).

2.3.1 Magnetic resonance imaging

Magnetic resonance imaging is based on nuclear magnetic resonance of atomic nuclei. Atoms have an intrinsic magnetic property called a spin. Spins interact between

themselves and with external magnetic fields. Nuclei with a net spin tend to align in the direction of a strong external magnetic field, such as the one applied in MR device. The most used nucleus in MR-imaging is hydrogen due to its abundance in the human body. In the absence of a magnetic field the spins of protons (hydrogen nuclei) are oriented randomly but when a strong homogenous magnetic field B_0 in the direction of z -axis is applied, they align in the direction of the field and begin precessing around that axis. This precession frequency is called *Larmor frequency*. This frequency is nucleus-specific and is defined by the following equation,

$$\omega = 2\pi f = -\gamma B_0, \quad (6)$$

where γ is called the *gyromagnetic ratio*, which is the ratio of a particle's or system's magnetic moment to its angular momentum. In the external magnetic field B_0 , hydrogen has two possible magnetic moment vector orientations: parallel and anti-parallel. There are always more spins oriented parallel to B_0 . The magnitude of net magnetization is proportional to the ratio between the number of spins in the parallel and anti-parallel states. During MRI, a radiofrequency electromagnetic pulse is applied orthogonally to B_0 and the net magnetization direction is flipped by angle θ . After the pulse, the net magnetization immediately starts to recover to its original equilibrium value. Returning to equilibrium is called relaxation and there are two mechanisms characterized by the two relaxation time constants. Recovery of the longitudinal magnetization (M_z) is described by T_1 and the loss of spin phase coherence by T_2 (M_{xy}). Because there are minor inhomogeneities in the magnetic field over time, the total relaxation due to loss in phase coherence is described by T_2^* . The T_1 recovery is described by the equation

$$M_z = M_0(1 - e^{-t/T_1}) \quad (7)$$

and the T_2 decay by the equation

$$M_{xy} = M_0(e^{-t/T_2}). \quad (8)$$

In order to create an image of the brain, the signal from each voxel has to be separately measured. This can be done with the help of gradient fields, formed by three gradient magnets in x -, y - and z -directions. Since the Larmor frequency depends on the external magnetic field, applying gradient field along B_0 a linear variation in magnetic field along z -direction is created. The radio frequency pulse can only excite spins which are precessing within the same frequency band as the pulse, and this way a specific 2D slice of the image can be selected. This is called slice selection, which locates the image in z -direction. The encoding of location in x - and y -directions are determined with additional frequency- and phase-encoding gradients. Phase-encoding is performed by applying a gradient field in y -direction, which causes the precession frequencies to vary along y -axis. Frequency-encoding is achieved by applying gradient field along x -axis and as a result, the whole three dimensional MR image is produced by the unique combination of precession frequencies and phases. This information is collected to k -space (Benjamini & Hochberg, 1995), and the final image is reconstructed using the inverse Fourier transform.

2.3.2 Computed tomography

CT imaging is based on electromagnetic radiation, which penetrates through the patient and the attenuated beam is detected on the opposite side of the patient. The source of the radiation is the x-ray tube. Electromagnetic radiation is produced as a result of two events: as the electrons in the inner shells of an atom move between the shells and as a result of deceleration of a charged particle due to electromagnetic fields inside the matter. The first is called characteristic x-ray radiation and the latter *bremsstrahlung*. Characteristic radiation is discrete, since only certain levels of energy are allowed in an atom, whereas *bremsstrahlung* is continuous.

Characteristic radiation occurs, when particles with sufficient amount of kinetic energy to knock the electrons out of their orbits hit the atoms. In an X-ray tube electrons depart from the high voltage cathode and they are accelerated towards the anode. The electrons approach the anode with great energy and produce X-ray radiation as they are decelerated substantially fast in the close proximity of the anode.

These photons of X-ray radiation then penetrate the object under imaging, for example, a human head. The photon energy attenuates in two ways: via the photoelectric effect (absorption) and scattering inside the tissue. To be used in clinical practice to differentiate between tissue types, Hounsfield units describe the degree of attenuation, which reflects the tissue type, through which the radiation penetrated (Hounsfield, 1980). The units range is from -1000 to 3000 , which in human tissue are represented as gases and dense bone, respectively (Hounsfield, 1980).

In computed tomography, the image is constructed from the detected projections of these attenuated X-ray beams. The detectors are either gas chambers or scintillation detectors. The idea of today's CT imaging is based on the patent by Gabriel Frank in 1940 (Webb, 1992). The most recent, fourth-generation CT scanners use scintillation detectors, which utilize the photoelectric effect. When a photon of X-ray radiation knocks an electron out of its orbit, a flash of light is produced in the presence of phosphor. Then a photomultiplier amplifies the signal and photo-detectors convert the light signal from scintillation crystals into an electrical signal.

To obtain the X-ray beam properly, collimators are needed. One of the purposes of collimation is to decrease scattering in the measurement. Artifacts are created not only from shortcomings of the collimator but also from the presence of metal objects in the body. In a modern CT device, there is a ring of detectors, which discretely detect the X-ray beam projections. The geometry of a fourth-generation scanner is presented in Figure 1. It consists of a rotating X-ray tube and a stationary detector ring. The beams from different angles acquired during one rotation about the patient form the whole projection view (presented in grey). The fan shaped beam, hitting the detector ring at one point of the tube is presented with solid lines.

We can consider a group of measurements from different angles as a group of linearly independent equations. When the number of equations increases, it becomes hard to solve them. The method of simply solving these equations is called algebraic reconstruction technique (ART). Other more sophisticated image reconstructions

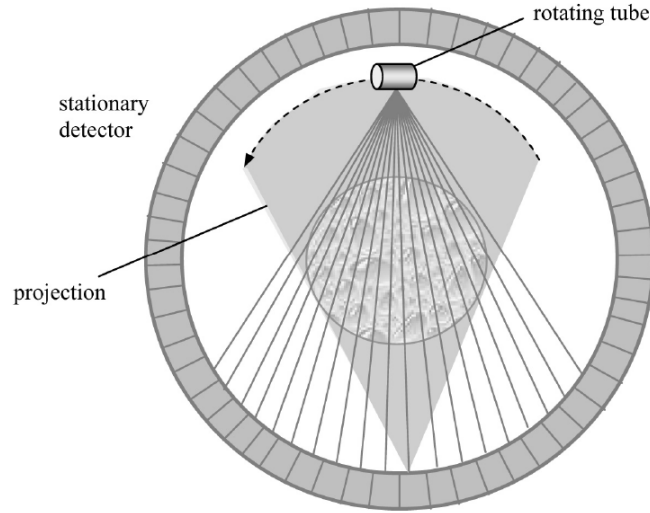


Figure 1: Geometry of a fourth-generation detector ring (Hsieh, 2009).

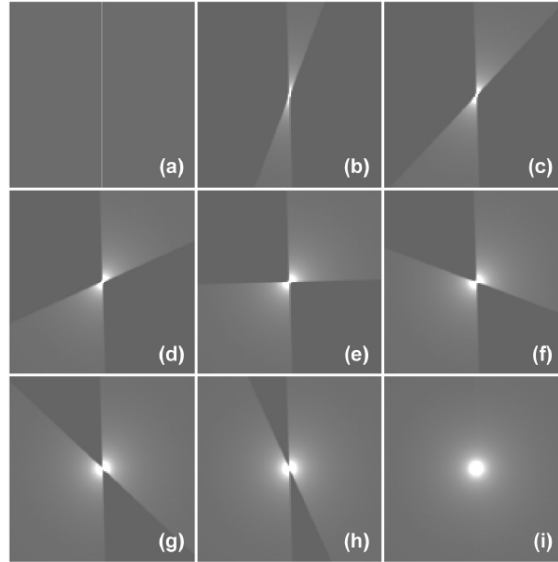


Figure 2: The idea of backprojection presented with a point object (Hsieh, 2009).

methods are more powerful to solve these problems and one solution is the so called iterative reconstruction method. In addition to these methods, there is the Fourier reconstruction method and the filtered backprojection method.

The idea in CT image reconstruction is to get an estimate of the attenuation distribution inside the object. One of the most important steps in modern CT image reconstruction is backprojection. The basic idea is presented in Figure 2. In the case of an isolated point object, we first take a vertical projection intersecting the object (Figure 2 (a)). The projection corresponding the point is an impulse function, which peak is located where the projection intersects the object. At that moment we do not know anything about the exact location of the object, either than that it is

located somewhere at the projection line. Therefore, we have to assume a uniform probability distribution on that line. Then we take multiple projections in different angles (Figure 2 (b)-(h)), and finally with an angular range from 0 to 180 degrees we have a rough estimate of the object (Figure 2 (i)) (Hsieh, 2009)

This last figure with angles from 0 to 180 is actually an impulse response of the backprojection process, which can be deconvolved with the inverse of the impulse response to recover the original object. This process is called backprojection filtering. (Hsieh, 2009)

3 Methods

The registration tool library used in this thesis is the National Library of Medicine Insight Segmentation and Registration Toolkit (ITK). The following chapters first introduce the overall ITK registration framework and then each part of the registration process separately. Thereafter, the data used in this work are described and the method used to compute registration errors. Finally, the steps of the optimization process are introduced.

3.1 Insight Segmentation and Registration Toolkit: registration framework

ITK v4, the first version released December 2011, is a publicly available templated C++ library, which offers image processing algorithms especially for the needs of biomedical applications (Lowekamp et al., 2013; Johnson et al., 2015). It was designed to be generic, flexible and extensible, in order to enable users to develop the existing modules or completely new ones in it. In order to hide some of ITK's demand-driven pipeline, an easy-to-use Python-based interface, SimpleITK (Lowekamp et al., 2013), was designed. The aim was to, for instance, facilitate rapid prototyping and scientific activities, as well as to provide users with an open-source library. Due to its simplified nature, which enables rapid developing and prototyping, this interface was chosen to be used in this thesis.

The simplest example of a registration framework is presented in Figure 3. Image registration is an optimization problem, where a moving image is brought into alignment with a fixed image by a spatial transformation (Johnson et al., 2015).

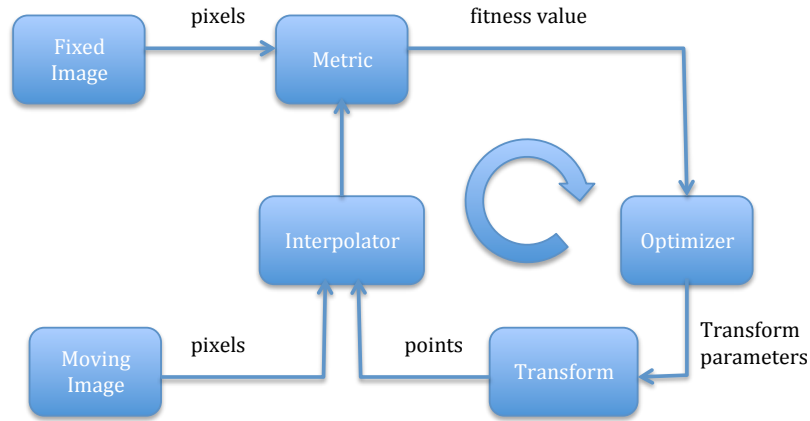


Figure 3: The basic components of a simple image registration framework (adapted from Johnson and colleagues (2015)).

The interpolator takes into account differences in sampling and resolution. The metric module tells how well the transformed moving image is matched to the fixed image. This provides a value to be optimized, the fitness value, and the search space

is defined by the transformation parameters. In our case, these are the six parameters in a rigid transformation (3 rotation parameters and 3 translation parameters).

A more flexible version to the above traditional registration framework, the ITK v4 framework, is presented in Figure 4. The main difference is the concept of virtual image domain, which means that the sampling domain is not the fixed image domain with different sampling frequency, but a whole new sampling domain consisting of arbitrary set of physical points (Avants et al., 2014b). This is where the registration and actually the metric evaluation happen. To register images in virtual domain an initial transform is set for both fixed and moving image to transform them from the physical domain to virtual domain. Nevertheless, the fixed image transformation parameters from fixed image domain to virtual domain do not effect the optimization process. The legacy registration framework is a special case, in which the virtual image domain is the fixed image domain. However, since the most practical cases including the case of CT to MR registration in this thesis, the virtual image is the fixed image, the virtual image domain is set to fixed image domain by default. (Johnson et al., 2015)

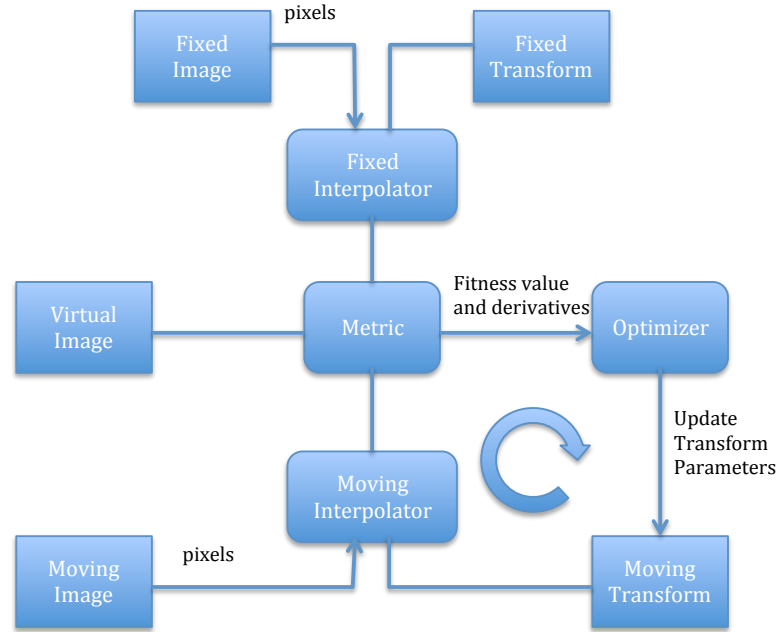


Figure 4: ITK version 4 registration framework (adapted from Johnson and colleagues (2015)). The rectangular boxes are process objects and the rounded boxes are data objects.

In ITK v4 framework (Figure 4), the metric module is the key component evaluating the intensity values of the fixed and moving image in the physical points of the virtual space. In addition to fixed, moving and virtual images, the metric module controls the fixed and moving interpolators, as well as the fixed and moving transforms. The metric cost function derives the fitness value and derivatives to the optimizer, which then guides the moving transform to update the transformation

parameters. This process is repeated until the user-defined convergence criteria are met.

ITK v4 (Avants et al., 2014a) offers several advantages over older versions. Automated scaling between translations and rotations in optimization reduces parameter optimization. A related approach automatically restricts the step sizes with gradient based optimizers. These implementations facilitate the registration optimization so that the researchers can focus on the design or comparison of registration strategies.

All transforms, metrics, interpolators and optimizers available in ITK and SimpleITK are presented in Appendix C.

3.2 Metrics

In rigid registration of intramodal images, the intensity values can be compared to find the optimal alignment. In this case, either *Mean squares* or *Correlation* metric can be used. With Mean squares metric the optimizer tries to minimize the mean squared pixel-wise intensity difference between the two images. With Correlation metric the pixel-wise cross-correlation is computed and normalized by the square root of the autocorrelation of the two images.(Johnson et al., 2015)

In addition to Mean squares and Correlation metrics, other metrics available in ITK can be found in Appendix C. Given that MR and CT images represent the same underlying anatomy, they share some mutual information (Wells et al., 1996). Therefore, mutual information is suited for registering multimodal images (Wells et al., 1996; Pluim et al., 2003). In automated image registration algorithms mutual information has been the most popular metric used since the pioneering work by Viola & Wells III (1997). A registration technique which attempts to maximize the information that one image provides from the other, was first introduced by Wells and colleagues (1996).The only assumption made in this technique is that one volume provides the most information when the two volumes are perfectly aligned.

To describe the process, two imaging volumes are defined as a *test volume* (moving image) and a *reference volume* (fixed image). A voxel of the moving image is referenced as $v(x)$ and a voxel of fixed image as $u(x)$. The coordinate transform T , relating the location of moving image voxel to fixed image voxel can be presented as $v(T(x))$. The estimate of T maximizing the mutual information can be presented as follows;

$$\hat{T} = \arg \max_T I(u(x), v(T(x))) . \quad (9)$$

In a registration algorithm, samples of reference volume locations x are drawn. Mutual information is presented in terms of entropy as follows;

$$I(u(x), v(T(x))) \equiv h(u(x)) + h(v(T(x)) - h(u(x), v(T(x))) . \quad (10)$$

The entropy, which can be interpreted as a measure of complexity, uncertainty or variability of a random variable, $h(x)$ can be presented as

$$h(x) \equiv - \int p(x) \ln p(x) dx , \quad (11)$$

while the joint entropy of random variables x and y can be presented as

$$h(x, y) \equiv - \int p(x, y) \ln p(x, y) dx dy . \quad (12)$$

In Equation 10, the first term on the right represents the entropy in the fixed image volume and the second one the entropy in the part of the moving image volume into which the fixed image volume projects. The negative term is the joint entropy of the fixed and moving image volumes. For the comprehensive explanation of information theoretic approach of entropy, see Shannon (1948). These entropies in Equation 10 are defined by integrals over the probability densities of random variables $u(x)$ and $v(T(x))$. In medical image registration, we do not have straight access to these densities, so we have to use for example the following probability density estimate $p(z)$

$$p(z) \approx P^*(z) \equiv \frac{1}{N_A} \sum_{z_j \in A} R(z - z_j) , \quad (13)$$

where N_A is the number of trials in sample A , R is the windowing function and $P^*(z)$ is widely known as *Parzen window* (Duda et al., 1973). (Wells et al., 1996)

Parzen windows are used to estimate the probability density functions in ITK Mattes Mutual information metric and Joint Histogram Mutual Information metric, which are the metrics available in SimpleITK to register multimodal images (Johnson et al., 2015). These two metrics were used in the optimization process, since they are the only similarity metrics based on mutual information available in SimpleITK. The Mattes mutual information (MMI) metric is based on minimizing the negative of mutual information value of two images. The MMI metric in ITK is an implementation of computing mutual information as in Mattes and colleagues (2001) and (2003). MMI metric takes into account joint, marginal fixed, and marginal moving image probability distributions (Mattes et al., 2003).

Another similarity metric in SimpleITK based on mutual information is Joint Histogram Mutual Information (JHMI). This method is an implementation of the publication by Thévenaz & Unser (2000). The concept of joint histogram is presented in Figure 5. In the top row of Figure 5, two miss-aligned images are presented: on the left a test image t (moving image) and on the right a reference image κ (fixed image).

Pictorial description of the paired elements $(t_0, \kappa_0), (t_0, \kappa_1), (t_1, \kappa_0), (t_1, \kappa_1)$ are presented on the bottom of the Figure 5 on the left, which contribute to the joint discrete Parzen histogram. The resulting joint histogram is presented on the right.

Figure 6 illustrates the effect of misregistration with MR to MR registration (top row) and CT to MR registration (bottom row). From left to right the misregistration is increased, which results in blurring of the signal (Hill et al., 2001).

Part of the histogram estimation process is dividing the image intensities into discrete bins (Zhu & Cochoff, 2002). This parameter is important, since too few bins makes the registration algorithm unstable and too many introduce noise. Furthermore, less computation is needed with smaller number of bins. However, this might not necessarily decrease the computation time (Zhu & Cochoff, 2002). With both Joint

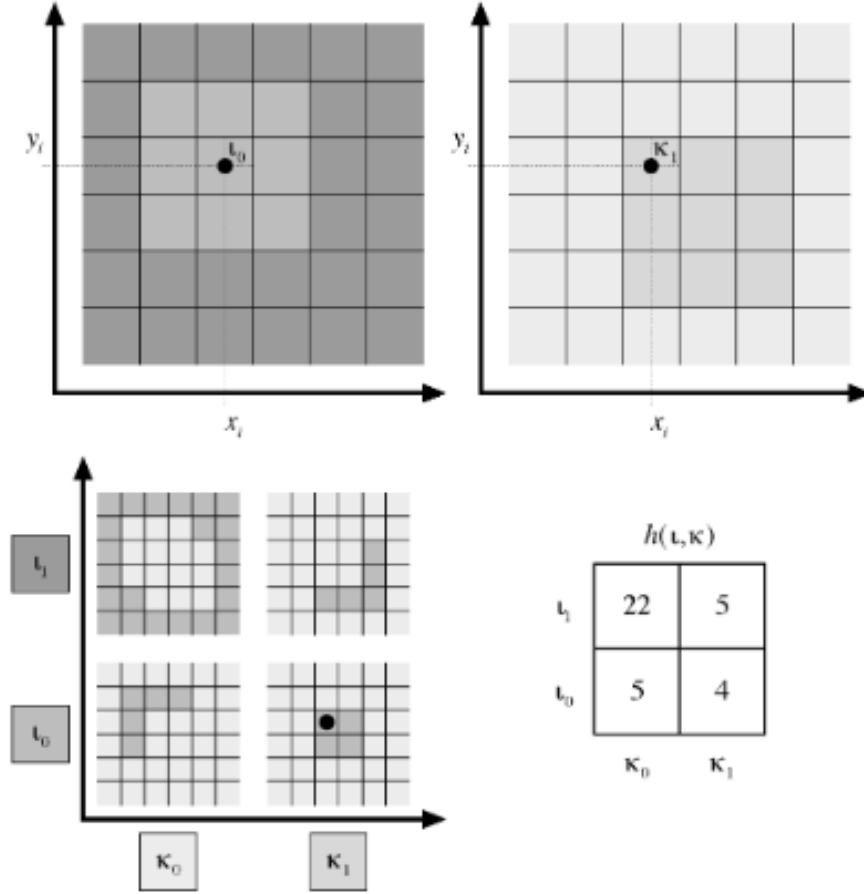


Figure 5: The idea of the computation of a joint histogram (Thévenaz & Unser, 2000).

Histogram Mutual information metric and Mattes Mutual information metric in ITK, user can define the number of bins. Usually there is no common rule on how many histogram bins is optimal, thus different numbers were tried in the work of this thesis.

Other important parameter related to metric is image sampling. It defines in how many locations x the the mutual information is computed. In SimpleITK, the sampling can be either random or regular. The user defines the sampling percentage or the number of samples. Since random sampling was widely used in ITK registration examples and also in MICCAI 2015 registration tutorial (*MICCAI 2015 : SimpleITK Registration*, 2015), we used this sampling strategy in our algorithm. The increasing number of samples used to compute the metric increases computation time yet stabilizes the algorithm. In the end of the optimization process we tried to find the optimal number of samples so that the time required by the algorithm would be minimal while the registration results would remain stable.

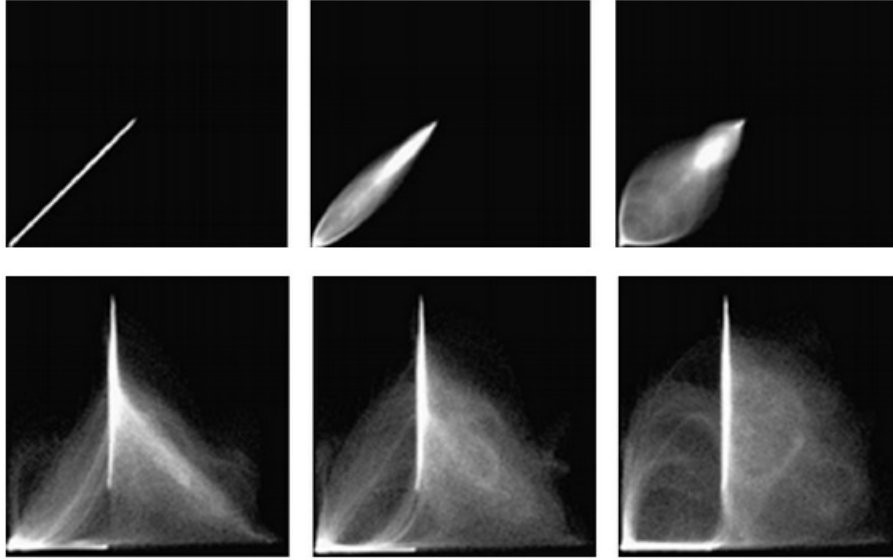


Figure 6: Joint histograms for the registration of identical MR images (top row) and CT to MR images (bottom row) of the head. On the left column, the images are aligned and towards right the misregistration increases. (Hill et al., 2001)

3.3 Transforms

When registering CT and MR images of the head, we assumed that it is enough to use a rigid transformation, since the bone structures of the head included in the registration are rigid so that affine transforms are not needed to align two images from one patient (Pluim et al., 2003). ITK provides a variety of transforms ranging from the simple translation and rotation transforms to general affine and kernel transforms (Johnson et al., 2015). In the case of image registration, these transformation parameters serve as the search space for the optimizer. Thus, the aim is for the optimizer to find those parameters, which result in best registration outcome. The more transform parameters the longer the registration takes.

In addition to affine transformations without scale and skew parameters, for rigid registration there are two suitable transforms in SimpleITK: *VersorRigid3DTransform* and *Euler3DTransform*. By definition, *versors* are rotational parts of *quaternion*. They can be also defined as *unit-quaternions* (Hamilton, 1969). Quaternions are an extension to complex numbers and used to present rotations. In older versions of ITK a special optimizer was required to perform with versor transforms. In ITK v4, *UpdateTransformParameters* method is implemented in *Transform* class, which enables the use of versor transforms with *RegularStepGradientDescentOptimizer*. In ITK Software guide it is suggested, that *VersorRigid3DTransform* would be the best transform to deal with rigid transformations in 3D, as well as to align images with great initial rotational difference. (Johnson et al., 2015)

In ITK Euler3DTransform, translations in three dimensions are performed following the rotations about axis x, y and z each one at a time. The center of rotation can be defined by the user. This center of rotation stays the same during optimization.

This transform is not recommended to be used with very large rotations and the initialization of the registration and especially the angles is crucial. (Johnson et al., 2015)

However, since the versor transform was not suitable with gradient descent optimizers in ITK v3, we decided to use this Euler transform in our registration algorithm with most optimizers, assuming that automated optimizer scaling would deal with large initial angles. Additionally, we optimized the versor transform with Regular Step Gradient Descent optimizer separately.

3.4 Interpolation

The transform maps fixed image points to moving image. An iterator walks through the points in fixed image and maps those coordinates to moving image. When the transform maps fixed image points to moving image and the metric compares the fixed image values to moving image values, interpolation is needed since the mapping usually falls on a non-grid position (Johnson et al., 2015). This is visualized in Figure 7.

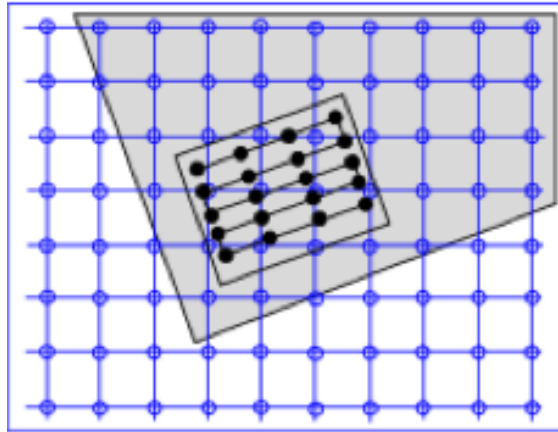


Figure 7: Moving (blue) and fixed image (black) grids after transformation. Transforms often map the fixed image points to non-grid positions in the moving image. (Johnson et al., 2015)

Four different interpolators are available in ITK and SimpleITK: *NearestNeighborInterpolateImageFunction*, *LinearInterpolateImageFunction*, *BSplineInterpolateImageFunction* and *WindowedSincInterpolateImageFunction*. For Windowed sinc interpolation there are different windowing functions available and we chose to test Hamming windowed sinc interpolation. The interpolators are used in the optimization, where they are utilized in smoothing the optimizer search space. Since they are called several times during optimization, user must compromise between simplicity of computation and smoothness of optimization. When they are used to interpolate the image in the end, the speed is less of a problem, since they are only called once. (Pluim et al., 2003; Johnson et al., 2015)

Nearest Neighbor interpolation is computationally the most simple one, since it does not require floating point computation (Johnson et al., 2015). It simply assigns the intensity of the nearest point to the non-grid position, in which the coordinate of transformed image falls. Linear interpolation, which is the most popular interpolation method (Pluim et al., 2003), assumes linear changes in intensity between grid points, which makes the interpolated intensities spatially continuous. In B-Spline interpolation the image intensities between grid points are presented by B-Spline basis functions, which are polynomial functions (Unser et al., 1993). Different order polynomials are used, and each higher order is computed by multiplying the previous order polynome function with the zero order polynomial. The intensity in non-grid position is computed by convolving the B-Spline kernel with the B-Spline coefficients associated with each grid point surrounding the non-grid position (Unser et al., 1993; Johnson et al., 2015).

B-Spline interpolation of order 0 is close to Nearest Neighbor interpolation and with order 1 identical to Linear interpolation. B-splines of order 0 to 5 are currently supported in ITK. In this thesis work we used order 3, which is the default order for B-Spline interpolation in ITK. (Johnson et al., 2015)

From the sampling theory it follows, that the ideal interpolation kernel is a sinc function (Meijering et al., 1999). ITK Software guide suggests, that it is the best interpolator for digitized data. Sinc functions are computationally much demanding compared to other interpolation methods, thus they are not possibly suitable for optimization. However, they could be used in resampling the moving image after the optimal transform is found (Johnson et al., 2015). Sinc interpolation is performed by convolving the spatial discrete signal with a sinc function,

$$\text{sinc}(x) = \sin(x)/x . \quad (14)$$

Due to the infinite support of sinc functions, it has to be multiplied with some windowing function, which results in windowed sinc interpolation. The most efficient are the ones with largest windows yet also most computationally demanding (Meijering et al., 1999). Interpolation errors may cause artifacts to optimizer cost function, which could result in registration errors (Pluim et al., 2000). Therefore, one should always choose the best available interpolator. However, this results in compromising the computational efficiency.

Multiple windowing functions are available in ITK: Cosinus, Hamming, Welch, Lancos and Blackman. The function kernel (windowed sinc interpolator) $K(t)$ is constructed by the sinc function and one of the windowing functions,

$$K(t) = w(t)\text{sinc}(t) = w(t) \frac{\sin(\pi t)}{\pi t} . \quad (15)$$

We decided to use Hamming window

$$w(x) = 0.54 + 0.46 \cos\left(\frac{\pi x}{m}\right) , \quad (16)$$

where m is the window radius. Hamming window was chosen, since it is the default window used in windowed sinc interpolation in ITK (Johnson et al., 2015) and we assumed that the results with different windows would not differ much.

3.5 Optimizers

In ITK image registration, the basic input to the optimizer is the cost function or a metric object (Johnson et al., 2015). The optimizers covered in this work are the ones available in SimpleITK: Amoeba, Gradient Descent, Gradient Descent Line Search, Regular Step Gradient Descent, Conjugate Gradient Descent Line Search, LBFGSB and Exhaustive optimizer.

Amoeba optimizer is a Nelder and Mead (Nelder & Mead, 1965) simplex based method, which does not require derivative information (Maes et al., 1999). The function to be minimized has n variables and it is dependent on the function values at $(n + 1)$ general simplex vertices. The vertex with highest value is replaced by another point, which is found by reflecting the point over the simplex face with the lowest value. In three dimensions ($n + 1 = 4$) the simplex is a tetrahedron, which moves in the parameter space trying to minimize the cost function. The method is regarded effective and computationally compact. (Nelder & Mead, 1965)

Gradient Descent method is a straightforward optimization method incorporating gradient information to minimize the cost function (Maes et al., 1999). At each iteration the current position in the parameter space is updated according to

$$p_{n+1} = p_n + L \frac{\partial f(p_n)}{\partial p_n}, \quad (17)$$

in which L is the *learning rate*, which is a fixed scalar governing the step size and $p(n)$ is the previous position (*Insight Segmentation and Registration Toolkit*, 2016). The next point is chosen in the direction of the steepest gradient (Maes et al., 1999). The optimizer runs through the user defined number of iterations without convergence checking.

Gradient Descent Line Search is a gradient descent method with a golden section line search (*Insight Segmentation and Registration Toolkit*, 2016). The gradient descent optimizer is followed by searching for the learning rate by a line search. The position is updated according to,

$$p_{n+1} = p_n + L_g \frac{\partial f(p_n)}{\partial p_n}, \quad (18)$$

in which L_g is the *learning rate by golden section line search*. The range in which the golden section line is searched for is defined by the user with lower and upper limit, where epsilon defines the resolution of the search. With smaller values more accurate results can be achieved, but with the cost of increased computation time (*Insight Segmentation and Registration Toolkit*, 2016).

In Regular Step Gradient Descent Optimizer, moving in the parameter space is performed in the direction of the gradient with step size which is computed with a bipartition scheme (Johnson et al., 2015). This optimizer can be used to optimize Versor Transform parameters. The new rotation versor is produced by composing the current rotation with the gradient rotation and the translational parameters are updated as usual in vector space. The definition of versor gradients follows Hamilton (1969).

As opposed to moving in the direction of the steepest gradient, Conjugate Gradient methods move in the direction of the conjugate to the previous direction with respect to the function that is to be minimized (Maes et al., 1999). In ITK, the direction is defined by Polak-Ribiere conjugate gradient d and line search is used to find the best value for the learning rate (Johnson et al., 2015). The position is updated by,

$$p_{n+1} = p_n L_g d . \quad (19)$$

In gradient descent based optimizers the learning rate, which governs the step size except with Regular Step Gradient Descent, can be set by the user, estimated after the first or after each iteration (*Insight Segmentation and Registration Toolkit*, 2016; Johnson et al., 2015). The user sets the maximum step size in physical units and the learning rate is estimated so that each voxel's change is less than this user defined step size in physical space.

LBFGSB optimizer is a modified version of Limited memory Broyden, Fletcher, Goldfarb and Shannon minimization (Johnson et al., 2015). The basic idea is to make the algorithm efficient with simple bounds (Byrd et al., 1995). The gradient projections from LBFGSB matrix are used to approximate the Hessian matrix of the function to be minimized. Once the search direction is found, line search is performed subject to the bounds and the current function value is computed.

Exhaustive optimizer fully samples the parameter space grid, with the resolution defined by the number of steps in optimization (Johnson et al., 2015). The grid in the parameter space is centered on the initial position. A side of a region x in the grid is defined both by the number of steps (n) and the scaling factors (s) in each dimension,

$$x = l * (2 * n + 1) * s , \quad (20)$$

in which l is the step length. In other words, exhaustive optimizer goes through all possible solutions with the resolution defined by the number of steps and scaling. This is quite time consuming and in the work of this thesis, where the aim is to find a robust algorithm even for cases, where the original images are far apart, this might not be the optimal choice due to the great computational effort needed.

ITK v4 offers automated parameter scaling to facilitate transform parameter optimization (Avants et al., 2014a). After each iteration and small parameter update, the change in magnitude of physical space deformation brought about by the transformation is analyzed. The effect of changing parameter p_i for one unit can be defined in different ways. One of these is based on maximum shift of voxels and another on the average norm of transform Jacobians (Jenkinson & Smith, 2001). In SimpleITK, these parameter scaling functions are called by *SetOptimizerScalesFromIndexShift* and *SetOptimizerScalesFromJacobian*, respectively. In case the data is not isotropic, and index based scaling can not be used (Johnson et al., 2015), *SetOptimizerScalesFromPhysicalShift* is useful. It defines the change according to shift in physical coordinates. The effect of unit change in rotation is multifold compared to unit change in translation and this difference appears as long narrow valleys in the parameter search space. This makes the optimization difficult, and the automated scaling provides help to this problem. (Johnson et al., 2015)

If automated parameter scaling is not used, users are encouraged to compute the scaling factors (s) either according to,

$$s = 1/(10 * \sqrt{(x^2 + y^2 + z^2)}) , \quad (21)$$

in case the scale between translations and rotations needed to align the images is not known and the same scaling factor is used for all dimensions (*ITK Image registration*, 2010), or according to

$$\begin{aligned} s[0] &= 1.0/(10.0 * P[0] * I[0]) \\ s[1] &= 1.0/(10.0 * P[1] * I[1]) \\ s[2] &= 1.0/(10.0 * P[2] * I[2]) \end{aligned} \quad (22)$$

if the scaling factor is wished to be computed separately for each dimension ([*Insight-users*], 2007). P represents pixel spacing and I image size in all three dimensions.

3.6 Multiresolution framework

The aim of the multiresolution framework is to increase accuracy, robustness and speed. Registration is first performed in the coarsest level using less voxels to construct the histogram (Maes et al., 1999). This spatial mapping determined at the coarsest level is then used as an input to the next, finer level. This process is visualized in Figure 8. The process is repeated until the finest level is reached. With courser scales some local minima are removed, which makes registration more robust and accurate. Computation at coarser levels is faster followed by less iterations needed in the finer levels (Maes et al., 1999). (Johnson et al., 2015)

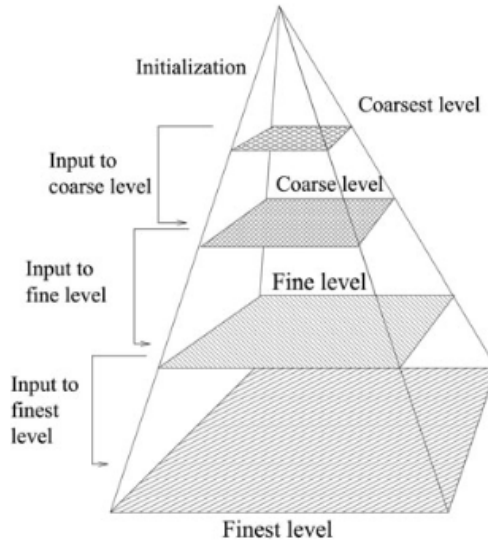


Figure 8: An illustration of the multiresolution levels used in image registration (Nejati & Pourghassem, 2014).

In addition to subsampling the image at coarse levels, smoothing is introduced by the registration filter. In ITK multiresolution framework the user defines the number

of levels, *ShrinkFactors* and *SmoothingSigmas* vectors (Johnson et al., 2015). With smoothing in low-resolution, fluctuations in the metric values can be avoided as well as getting trapped to local minima (*Insight Segmentation and Registration Toolkit*, 2016).

3.7 Initial Alignment

Registration initialization is crucial to avoid ending up out of the metric capture range. This results in the error message: *"Too many samples outside moving image buffer"* and registration process to end, since there is not enough overlap between the two images, so that the optimizer could still bring them into alignment. To this problem there is helper class *CenteredTransformInitializer*, which aligns either physical image centers or grey scale moments of both images.

The first method based on image geometry assumes that the objects to be registered locate in the center of the image. The distance between the centers is then passed to initial transform as translation. The second method computes the moments of grey scale images and aligns these centers of mass, which is then used as a center of rotation. The distance between mass centers is then passed as translational parameter to the initial transform. The moments method assumes that the two image objects have similar moments. Therefore, it might not be the optimal choice in multimodal image registration. (*Insight Segmentation and Registration Toolkit*, 2016; Johnson et al., 2015)

3.8 Masking and Windowing

With masking it is possible to ignore some irrelevant information from the images to be registered (Avants et al., 2014a). A spatial object mask can be set for the fixed image to outline the pixels used to compute the image metric. The spatial object mask can be a binary mask defined for only fixed image or for both fixed and moving images separately. (*Insight Segmentation and Registration Toolkit*, 2016)

In Mattes mutual information metric intensities are rescaled between 0 and 1 (Mattes et al., 2001, 2003). In CT images the electrodes introduce very high intensities to the images which can not be used in the registration, since they do not exist in MR images. All intensities over the intensities of dense bone (more than 3000 (*Hounsfield Unit*, 2016)), like the signals from electrodes and wires, are wasting histogram bins. In ITK, several windowing functions are available.

3.9 Data

In this thesis, we used data from multiple sources. To test and evaluate the registration parameters, we downloaded Retrospective Image Registration Evaluation (RIRE) - project data, which can be accessed via the link: <http://hdl.handle.net/1926/194>. We used MP-RAGE anatomical T1 images and CT images of nine patients from RIRE-project.

In order to have data representing more closely the needs of HUS, we used anonymous MR and CT images of both SEEG and Parkinson's patients to tune the registration parameters. The data was made anonymous by removing all patient information from the data. We had four datasets from Parkinson's (two patients) and 3 from SEEG patients. All together we had 16 datasets, which were used in the optimization process and accuracy testing.

To find registration parameters which would be rather robust to initial rotations, we deflected the CT images acquired from RIRE data library with different angles. The aim was to cover some large initial angles, which are possible to patients suffering for severe diseases such as stroke, which could results in uncontrollable head tilting for instance. According to neuroradiologist Antti Korvenoja in HUS Medical Imaging Center, in practice angles of 90° about z-axis and 45° about x-axis and y-axis are possible (Korvenoja, 2016). Having this as basis, we deflected the data with angles and translational components presented in Table 1. The centered transform initializer (Johnson et al., 2015) deals with this by aligning either the geometric centers or centers of mass.

Table 1: The deflected CT image rotations about each axis and translational component along z-axis. "Rand" refers to random number in the range in the brackets.

| Patient | z | x | y | Trans in cm |
|-------------|---------------|-------|-------------|-------------|
| patient_101 | 90 | -45 | rand(10,15) | rand(5,10) |
| patient_102 | -90 | 45 | rand(10,15) | 100 |
| patient_103 | 45 | 45 | rand(10,15) | rand(5,10) |
| patient_104 | -45 | 45 | -45 | 100 |
| patient_105 | 22.5 | 45 | rand(10,15) | rand(5,10) |
| patient_106 | -22.5 | 22.5 | 22.5 | 100 |
| patient_107 | rand(-15,-10) | 45 | rand(10,15) | rand(5,10) |
| patient_108 | 45 | 22.5 | 45 | 100 |
| patient_109 | -45 | -22.5 | rand(10,15) | 100 |

3.10 Computation of registration errors and evaluation of time performance

In order to define the registration accuracy, we compared the automatic registration with semiautomatic registration performed with an in-house developed graphical software from HUS Medical Imaging Center. The software is called ipsWorkstation and is based on ITK image registration library and VTK visualization library.

The semiautomatically registered CT images were saved in ipcWorkstation before resampling, so that the headers of images would be in same coordinates. The aim was to compare this transform to the automatically computed transform, which is the basis of computing target registration errors (TRE) (Hill et al., 2001).

In order to compare these transforms, a grid image was created in such a way that there are different intensities in the sub grids of the image (Figure 9. The

center of the grid is positioned approximately in the middle of the CT image. The grid image was transformed both with the semiautomatic transform and with the automatic transform. For both transformed grid images, the center coordinates of each subgrid having a different intensity (ranging from 1 to 217) were computed resulting in 217 coordinates. The Euclidean distances between coordinates between semiautomatically and automatically transformed grid coordinates were calculated. These distances are considered in our case the target registration errors (TREs), which are measures of registration accuracy. Figure 9 presents an example of one of the grids, which is placed in the middle of the CT image.

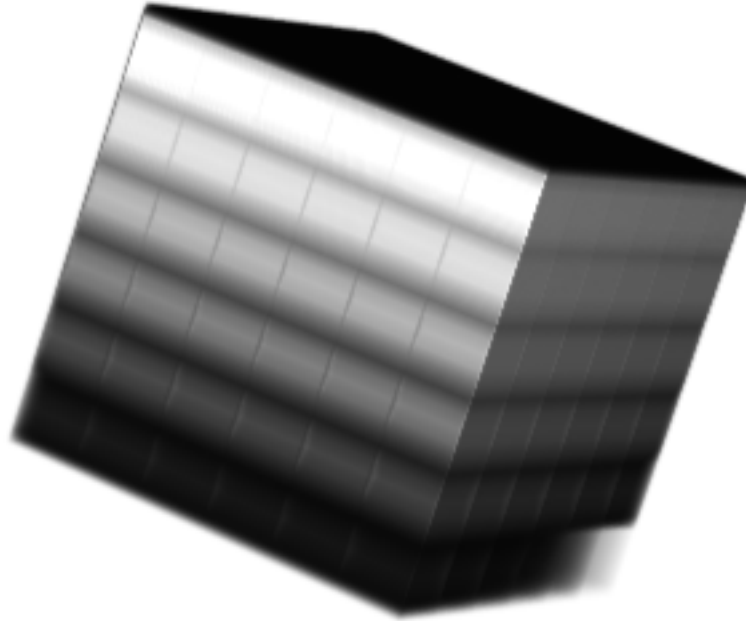


Figure 9: 3D grid. Subgrids have intensities ranging from 1 to 217.

To calculate the grid coordinates, and ITK-based Convert3D-tool was used. Convert3D is a tool offering complementary command-line image processing features to ITK-SNAP (Yushkevich et al., 2006).

The time used by a specific part of the Python script is determined with Python's `time.process_time()`. It has no relationship to the real-world time and it can be only used to measure time intervals, thus it is a unit-less value. Therefore, it can be used to measure relative performance, since it is not dependent on computational resource used by other users for example. According to Python documentation time module for version 3.5.1, it returns "the value (in fractional seconds) of the sum of the system and user CPU time of the current process" (*The Python Standard Library*, 2016).

3.11 Optimization process

The optimization process progressed as follows: We used a SimpleITK image registration example from MICCAI 2015 conference (*MICCAI 2015 : SimpleITK Registration*, 2015) and optimized one part of the registration algorithm at a time. After the optimal parameters and options were found for the registration block which was investigated, they were "locked" to be used in further analysis. Thereafter, the next block was optimized.

Once all the blocks were optimized, the number of samples used to compute the similarity metric in the image was gradually reduced. Furthermore, the number of bins for the metric computation was also reduced of the optimization process. This kind of manner to proceed is based on Insight users discussions, where users optimizing registration parameters were encouraged to first use a higher number of bins with abundant number of samples and after other parameters have been settled, reduce the number of both. ([*Insight-users*], 2007).

In the beginning of optimization process, mutual information (MI) values were used as an index of registration success. MI values are not as interpretable as for example the TRE values, but this choice was made since the computation of point based registration errors described in Methods part is rather time consuming. In the end, registration errors were used to define the accuracy of registration.

The table of all available parameters in ITK and SimpleITK optimization can be found in Appendix C.

In order to follow the progress of the optimization with ITK version 4, Insight implemented a *Observer/Command* design pattern (Vlissides et al., 1995). It provides methods to obtain the activity of the *ImageRegistrationMethodv4* class. This is especially practical, when the registration parameters are optimized for a special application. It is possible, that the registration takes extensive amounts of time to run, without providing any acceptable result. Since the registration process is driven by the optimizer, this might result from the wrong choice of optimizer parameters. (Johnson et al., 2015)

For instance, the current metric value can be obtained by calling the *Metric evaluate* -method. Additionally, in the end of registration the optimizer stopping condition can be printed. For instance, if the convergence criteria is met before the maximum number of iterations is run, this command reveals the number of iterations used for optimization in the finest resolution level. By adding commands to the registration method, also visual monitoring, such as following the optimizer path in the metric space is possible. It is possible to add commands, which are executed when a certain event occurs, such as a new iteration round. (*Insight Segmentation and Registration Toolkit*, 2016)

An example of the visual monitoring is presented in Figure 10. The progress of optimization is presented in terms of metric value (in this case MMI) and target registration errors. Three resolution levels are visible on the left. This presents very clearly, that the similarity metric functions are different on each level, thus they are not comparable.

Since it would be rather time consuming to visually follow the registration

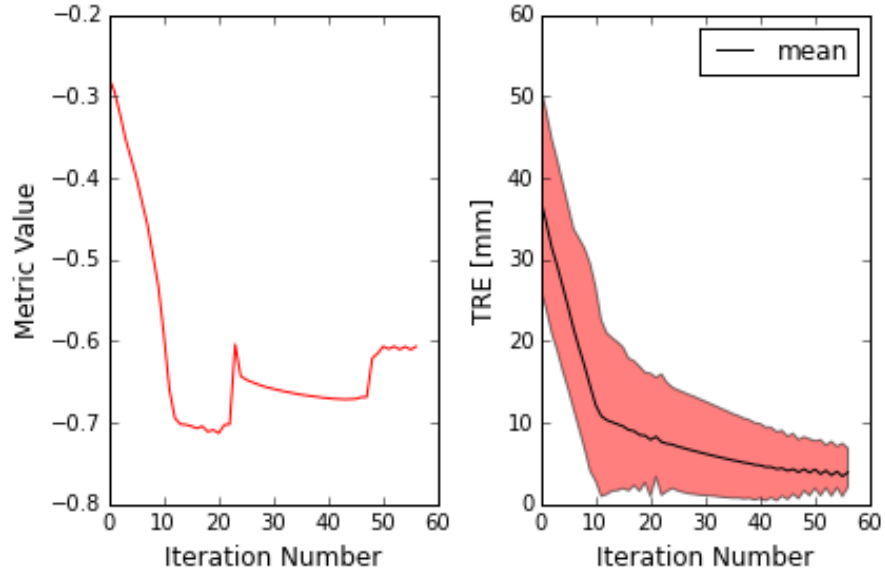


Figure 10: Exmaple of registration monitoring in SimpleITK. On the left, alterations in the metric value are being monitored. On the right, the corresponding TRE values are shown. (*SimpleITK Notebooks*, 2016)

progress for several patients with multiple registration parameters, we did not use these commands in the optimization of parameters. Furthermore, we did not have the possibility to visualize the target registration errors after each iteration, due to the high number of registrations needed in parameter optimization. To the final algorithm we added printing of metric values after each iterations, to enable the user to follow the behaviour of the similarity metric and overall registration progress.

4 Results

First, the intermediate results during the optimization process are presented. Each block of the registration algorithm is optimized separately, and the results from previous block are used in the next. Thereafter, the final registration algorithm is presented and its performance is discussed.

4.1 Parameter optimization for Euler transforms

We chose to begin the optimization process with the parameters based on SimpleITK example presented in MICCAI conference (*MICCAI 2015 : SimpleITK Registration*, 2015). First, we optimized the initial transform. Thereafter, the best performing initial alignment method was used in optimization of the optimizer parameters and so forth. In our initial case, the only difference compared to MICCAI example was that we used all samples of the image to compute the similarity metric.

The MICCAI conference SimpleITK example can be found in Appendix [D](#).

4.1.1 Initial alignment

The first part of the registration to be optimized was the initial alignment. If the initial alignment is not good enough, it might not be in the capture range of the similarity metric, and the registration might fail (*ITK Image Similarity Metrics*, 2016). Therefore, users are encouraged to find a good initial alignment to begin with (*[ITK]/[ITK-dev]Optimizing composite transforms and center of transform*, 2014). In initial transformation optimization we used the same parameters as in MICCAI tutorial (*MICCAI 2015 : SimpleITK Registration*, 2015), except we used all image samples to compute the metric and naturally the initial alignment mode was varied. We tried Centered transform initializer with both available methods: moments (aligning image moments) and geometry (aligning physical image centers), Metric evaluate method and Exhaustive method. Metric evaluate method and Exhaustive method were based on SimpleITK Notebook 63 (*SimpleITK Notebooks*, 2016). Additional angles to the ones used in the Notebook example were used in Metric evaluate method (Appendix [A](#)).

The moments method did not perform the registration for a few patients at all, so it was not considered to be an option in further analysis. From Figure [11](#) it seems, that Metric evaluate method would be the most robust one, if the moments method is left out. However, Metric evaluate is quite time consuming, with multiplication of factor 2.8 compared to the centered transform initializer with geometry. The centered transform initializer with geometry had the mean value of MMI (-0.4294) close to the mean of Metric evaluate method (-0.4342), thus it was chosen to be used in further optimization processes.

4.1.2 Optimizers

The second part of the registration optimization process was to find an ideal optimizer. First, we tried to find optimal parameters for each optimizer: Gradient Descent (GD),

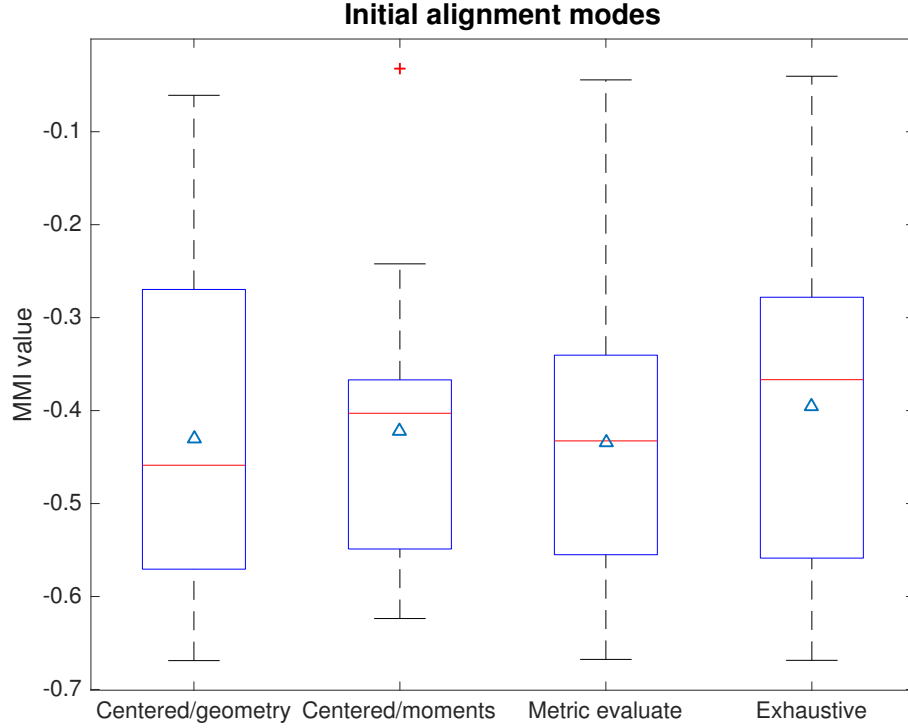


Figure 11: Final similarity metric values with different initial alignment modes. The triangles present the means over subjects of each method.

Gradient Descent Line Search (GDLS), Regular Step Gradient Descent (RSGD), Conjugate Gradient Descent Line Search (CGDLS), Amoeba, Exhaustive optimizer and LBFGSB optimizer.

With many optimizers, the parameter options chosen to be included in the optional combinations were taken from ITK registration examples (*Insight Segmentation and Registration Toolkit*, 2016). Different optimizer scalings were used; optimizer scales from physical shift, jacobian or user defined scales. The scaling factors are explained in Section 3.5. Exhaustive optimizer was left out of the analysis, since it is suggested to be only suitable when there is a limited number of possible solutions (*Understanding Optimization*, 2012) and it is rather time consuming. LBFGSB optimizer was performing poorly compared to other optimizers, since the mean MMI value for the best performing parameter combination was -0.3572. The following parameters with LBFGSB optimizer were used to get this result: gradientConvergenceTolerance=1e-5, numberOfIterations=100, maximumNumberOfCorrections=2, maximumNumberOfFunctionEvaluations=200, costFunctionConvergenceFactor=1e+7. Moreover, LBFGSB is used in ITK examples only with deformable registration, which suggests that it might not be the optimal choice for rigid registration. In addition, we were not able to find any stable parameter combination for Conjugate Gradient Descent Line Search optimizer, so that it would even perform the registration at some level. Therefore, it was also left out of further analysis.

The best performing parameters for four optimizers left (GD, GDLS, RSTG and Amoeba) were chosen by comparing the mean mutual information values across

patients of each combination.

The best performing combinations are presented in Table 2, and the accuracy of those combinations in Figure 12. Gradient Descent optimizer was computationally the most efficient one, and the multiplication factors compared to other optimizer in time consumption are: GD to GDLS 3.98, GD to RSGD 1.24 and GD to Amoeba 6.95.



Figure 12: Optimizer accuracy as mean registration error values in millimeters. Results of one patient are removed, since the registration was not successful.

Table 2: Optimizer parameters and scaling for each optimizer, which had the highest mutual information value.

| Optimizer and parameters | Parameter value |
|--|---------------------------|
| Gradient Descent | |
| Learning rate | 1.5 |
| Number of iterations | 100 |
| Convergence minimum | 1E-6 |
| Convergence window size | 5 |
| Estimate learning rate type | Each iteration |
| Maximum step size in physical units on | 0.0 |
| Scaling | Physical Shift |
| Gradient Descent Line Search | |
| Learning rate | 1.5 |
| Number of iterations | 100 |
| Convergence minimum | 1E-3 |
| Convergence window size | 5 |
| Line search lower limit | 0.5 |
| Line search upper limit | 5 |
| Line search epsilon | 0.01 |
| Maximum nuber of iterations | 20 |
| Estimate learning rate type | Each iteration |
| Maximum step size in physical units on | 0.0 |
| Scaling | Jacobian |
| Regular Step Gradient Descent | |
| Learning rate | 0.5 |
| Minimum step length | 1E-3 |
| Number of iterations | 100 |
| Relaxation factor | 0.5 |
| Gradient magnitude tolerance | 1E-4 |
| Estimate learning rate type | Each iteration |
| Maximum step size in physical units on | 0.0 |
| Scaling | PhysicalShift |
| Amoeba | |
| simplex delta | 0.01 |
| Number of iterations | 300 |
| Parameters convergence tolerance | 1E-4 |
| Function convergence tolerance | 1E-6 |
| with Restarts | true |
| Scaling | [1,1,1,0.001,0.001,0.001] |

4.1.3 Similarity metric

Next, Joint Histogram Mutual Information was used as a similarity metric with four best performing optimizers: GD, GDLS, RSGD and Amoeba. From Figure 13 it is clear, that at least with these optimizer settings JHMI metric performs worse than MMI metric. Also it is not as robust as MMI, since it could not perform the registration at all for a few patients. In 14 out of 14x4 cases (14 image volumes and 4 optimizers), JHMI metric did not perform registration properly, and these unsuccessful results were left out of Figure 13.

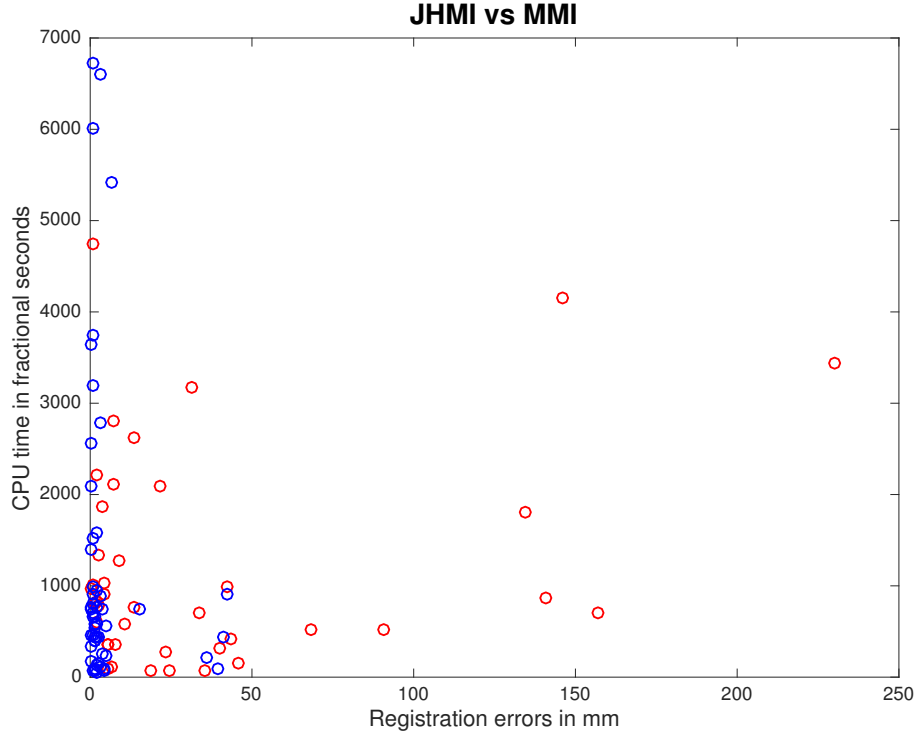


Figure 13: Registration accuracy and CPU time for Joint Histogram mutual information (red) and Mattes mutual information (blue).

4.1.4 Multiresolution strategy

After verifying that the most suitable metric for this specific registration application is Mattes mutual information, the effect of resolution levels and smoothing was tested. We used the same multiresolution strategies as Maes and colleagues (1999), in addition to one four level strategy. All multiresolution strategies with smoothing sigmas are shown in Table 3. The numbers inside brackets in strategy column represent the subsampling factors in corresponding resolution levels. The smoothing sigmas for each level represent the sigmas of a Gaussian filter used in smoothing.

For each multiresolution strategy presented in Table 3, registration accuracy with four different optimizers, as well as the CPU time used by the registration process are visualized in Figure 14. The mean accuracies for each multiresolution strategy

Table 3: Multiresolution strategies and the corresponding smoothing sigmas.

| Strategy | Smoothing sigmas |
|-----------|------------------|
| [1,1,1] | [2,1,0] |
| [2,2,1] | [2,1,0] |
| [2,2,2] | [2,1,0] |
| [3,3,1] | [2,1,0] |
| [3,3,2] | [2,1,0] |
| [4,4,1] | [2,1,0] |
| [4,2,1] | [2,1,0] |
| [8,4,2,1] | [4,2,1,0] |

are presented in Table 4. The results for one patient were left out of the visualization, since the registration was not successful and the figure is this way clearer to interpret.

| Strategy | Gradient Descent | GD Line Search | Regular Step GD | Amoeba |
|-----------|------------------|----------------|-----------------|-------------|
| [1,1,1] | 1.56 | NaN | 3.57 | 0.92 |
| [2,2,1] | 1.82 | 1.51 | 3.44 | 2.05 |
| [2,2,2] | 1.76 | 1.94 | 3.48 | 1.95 |
| [3,3,1] | 2.17 | 1.73 | 4.31 | 2.33 |
| [3,3,2] | 1.94 | 1.98 | 3.58 | 2.8 |
| [4,4,1] | 2.14 | NaN | 4.11 | 2.18 |
| [4,2,1] | 1.83 | NaN | 4.04 | 2.63 |
| [8,4,2,1] | 2.67 | 1.43 | 4.99 | 2.13 |

Table 4: Accuracy means of multiresolution strategies for four optimizers and 9 patients. The best means after the means of [1, 1, 1] strategy are printed bold. One patient with initial rotation of 90 degrees about z-axis was left out of the analysis.

GD optimizer with multiresolution strategy [2, 2, 2] and GDLS optimizer with [8, 4, 2, 1] were chosen to be used in further analysis. RSGD had the greatest registration errors, and Amoeba was substantially slower than GD or GDLS.

Multiresolution strategies

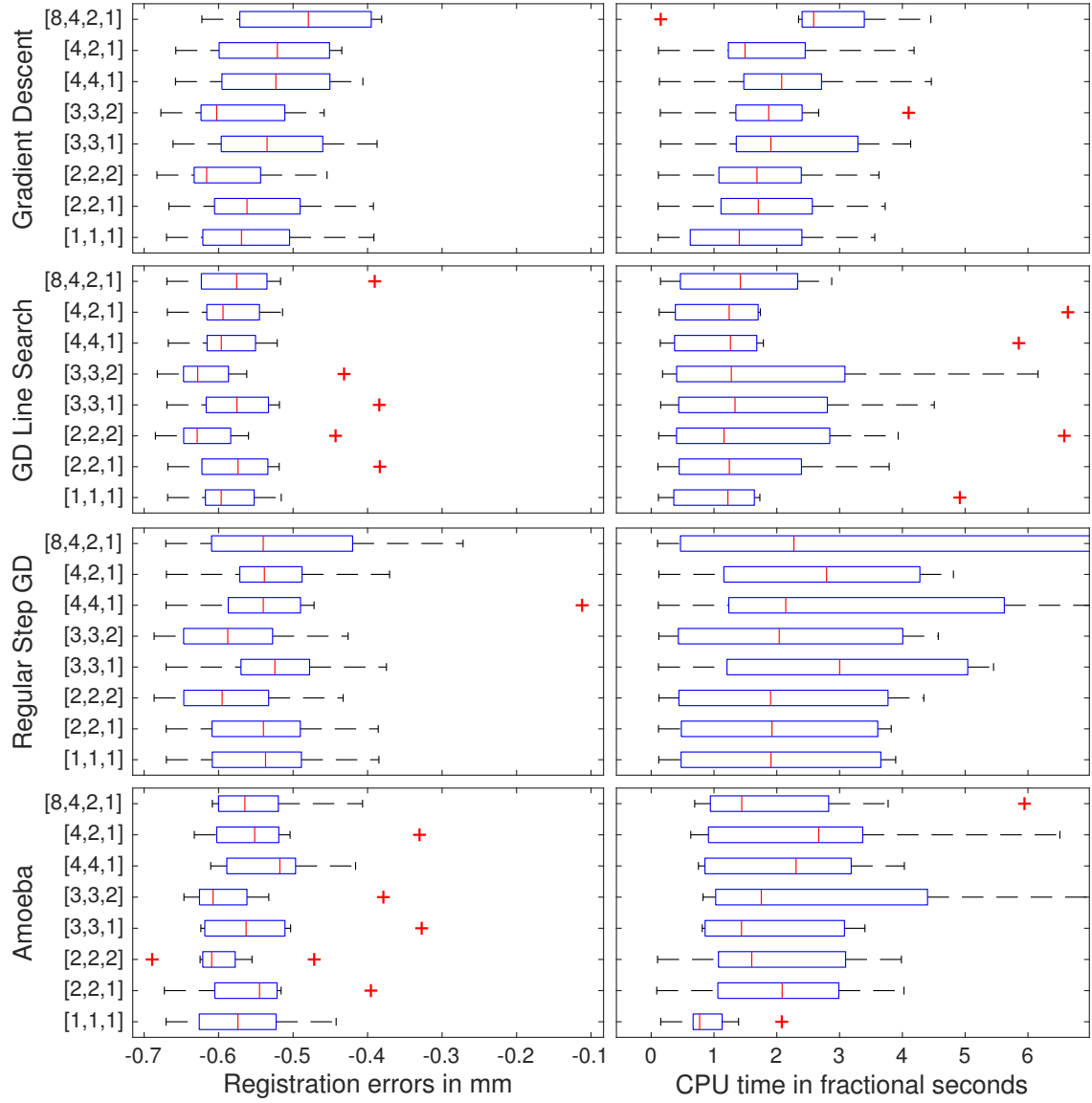


Figure 14: Different multiresolution strategies. Accuracies as registration errors for each optimizer and strategy are presented on the left and the consumed CPU time on the right.

4.1.5 Interpolators

The next step in registration parameter optimization was to test different interpolators. The accuracies of these interpolators are presented in Figure 15. Since it was not recommended to use sinc-interpolation in optimization process (Johnson et al., 2015), we only tested Nearest Neighbor, Linear, B-spline of order 3 (the default) and Gaussian interpolator.

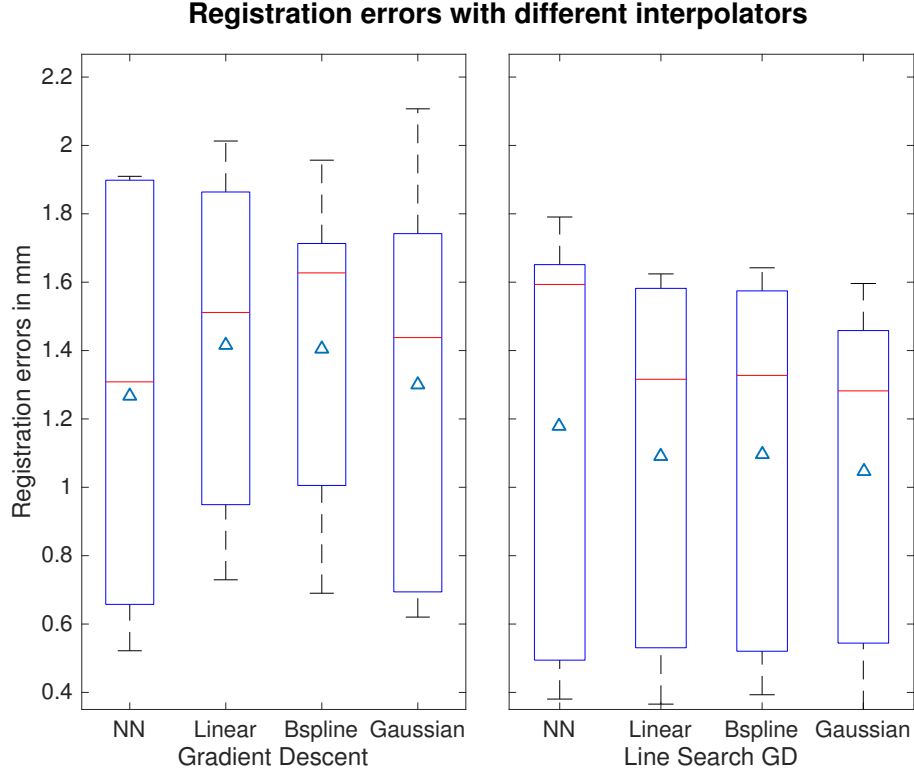


Figure 15: Registration errors introduced with four interpolators for Gradient Descent and Gradient Descent Line Search optimizers.

As can be seen from Figures 16 and 17, gaussian and b-spline interpolators are substantially more time consuming in comparison to nearest neighbor and linear interpolators. Since the difference in mean registration error between GD optimizer with nearest neighbor interpolation and GDLS optimizer with linear interpolation is only around 0.2 mm, we cannot tell which one is actually closer to the perfect alignment. This results from the comparison of automatic transform to semiautomatic transform, which is used as the ground truth, to compute registration errors. It is possible, that the semiautomatic transform is actually more inaccurate than the automatic transform. Therefore, we chose Gradient Descent optimizer with Nearest Neighbor interpolator due to its faster performance.

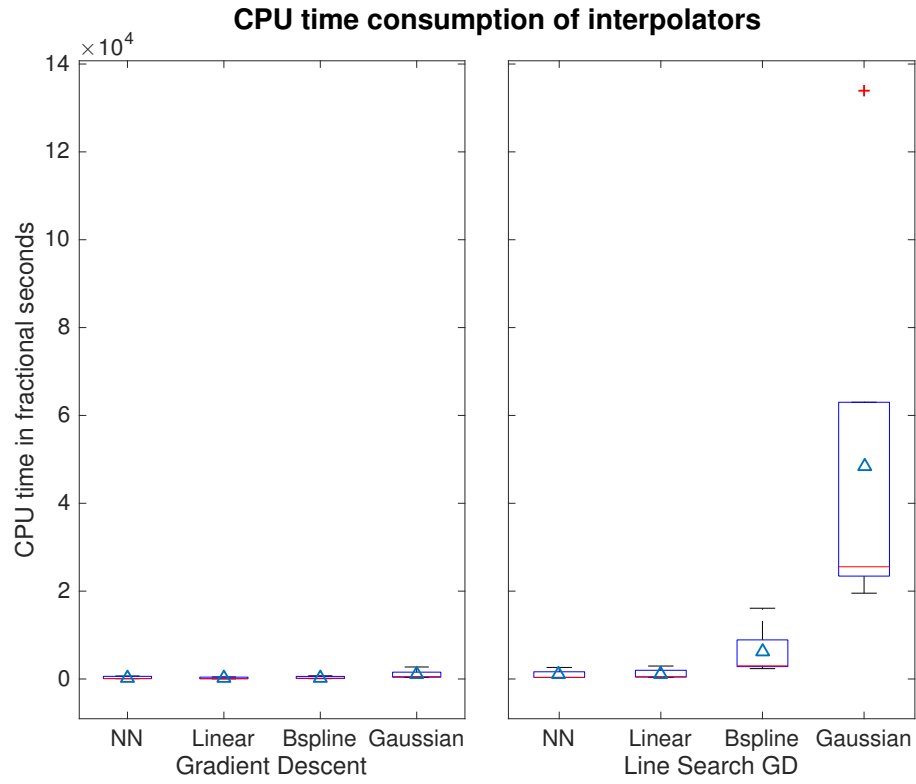


Figure 16: Consumed CPU time with four interpolators for Gradient Descent and Gradient Descent Line Search optimizers.

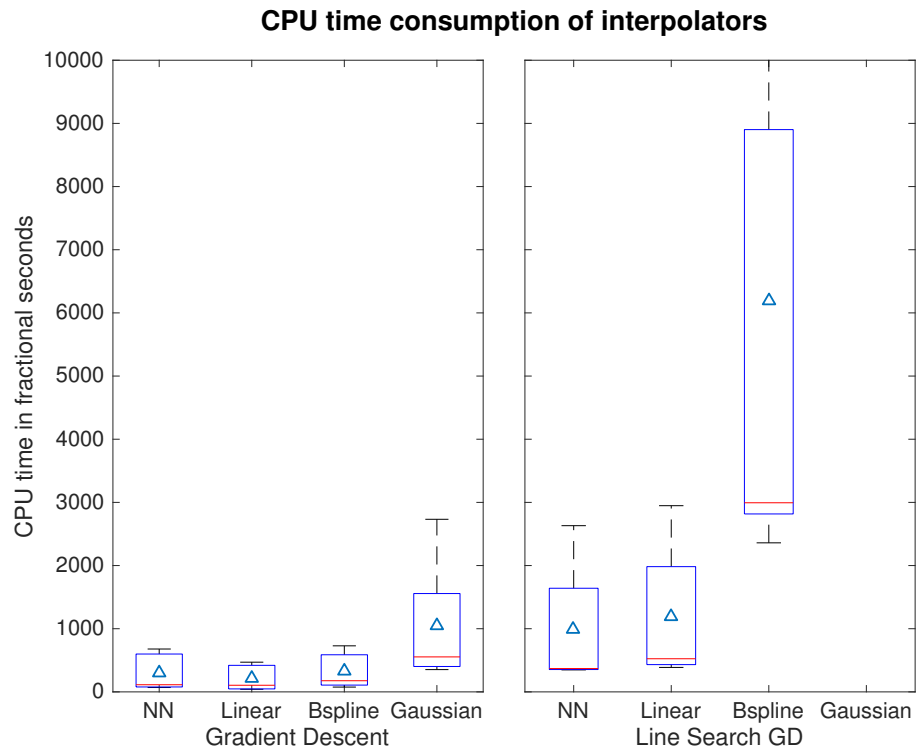


Figure 17: Enlarged CPU time consumption.

4.1.6 Number of bins and sampling

Once the best metric, optimizer, resolution level and interpolator were found, we tried how low can the number of bins and number of samples be in order to prevent the registration algorithm from failing. The aim of this step was to see, if the performance could be induced in respect of CPU time needed for registration. For the three patients used to test this part, the threshold for successful registration was 12 histogram bins and 1% of samples to compute the metric. This can be seen from Figure 18. All values for number of bins and sampling percentage tested are shown in Table 5. The CPU time consumption as a function of the number of histogram bins and as a function of sampling percentage are considered in Figure 19.

It seems that the number of bins does not have a substantial effect on CPU time consumption. The registration with 6 bins was not successful. Nevertheless, increasing the sampling percentage slightly raises the CPU time used.

Table 5: Number of bins and sampling percentages. The combinations of each of these values were tested.

| Number of bins | Sampling percentage |
|----------------|---------------------|
| 6 | 0.01 |
| 12 | 0.1 |
| 24 | 0.2 |
| 50 | 0.5 |
| | 1 |

Effect of number of bins and sampling percentage on registration errors

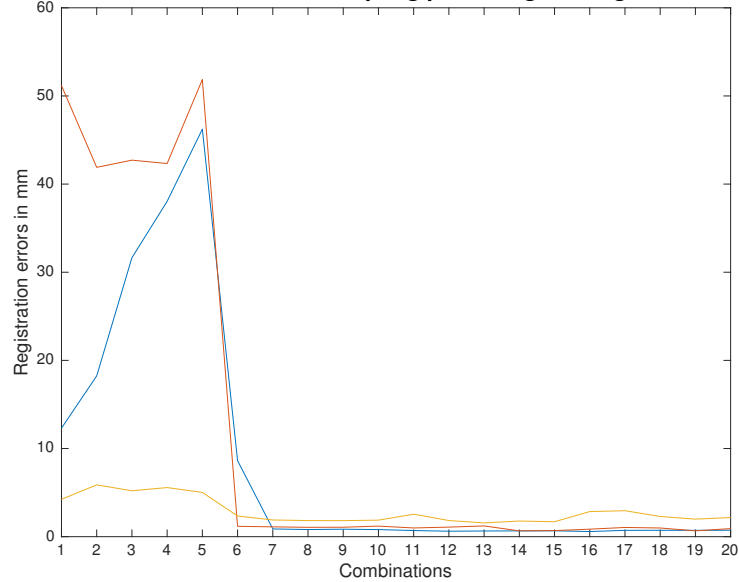


Figure 18: The registration errors with all combinations from Table 5 in Appendix B for three patients.

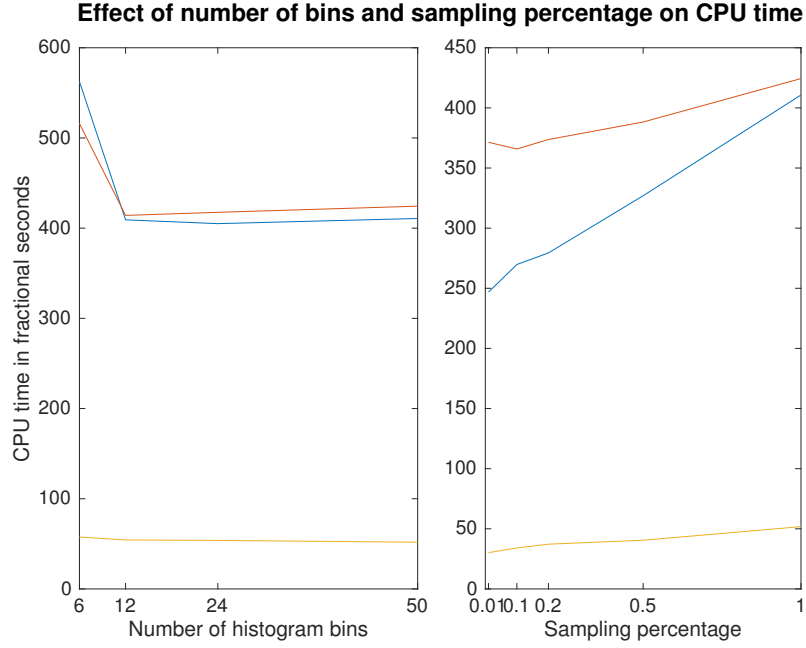


Figure 19: On the left, the effect of the number of histogram bins on CPU time when all samples are used. On the right, the effect of sampling percentage on CPU time when 50 histogram bins are used to compute the metric.

4.1.7 Windowing and Masking

In the final algorithm, all CT images are windowed so that intensities higher than 3000 are set to 3000. This procedure was performed due to the high intensities coming from metal electrodes or deep brain stimulation probes. Since the intensity of dense bone in Hounsfield units is around 3000 (*Hounsfield Unit*, 2016), we decided to use it as an intensity threshold.

In the end, we tested if masking of image intensities would make a difference in registration result. Assuming that the interface between skull and air is important for Mutual information, we masked a layer of air in addition to the brain volume. We tested for one patient how the masking effects the accuracy and CPU time of registration. However, it turned out that the mask did not considerably improve either the final accuracy nor the time performance with Gradient Descent optimizer.

4.1.8 Accuracy of the algorithm with Euler transforms

From Figure 21 and Table 6 we can see, that for 9 out of 15 patients the maximum registration errors fell under 2 mm, which was the approximate misregistration possible to visually observe by an expert (Fitzpatrick, Hill, et al., 1998). The range in CPU time used by each registration is quite wide.

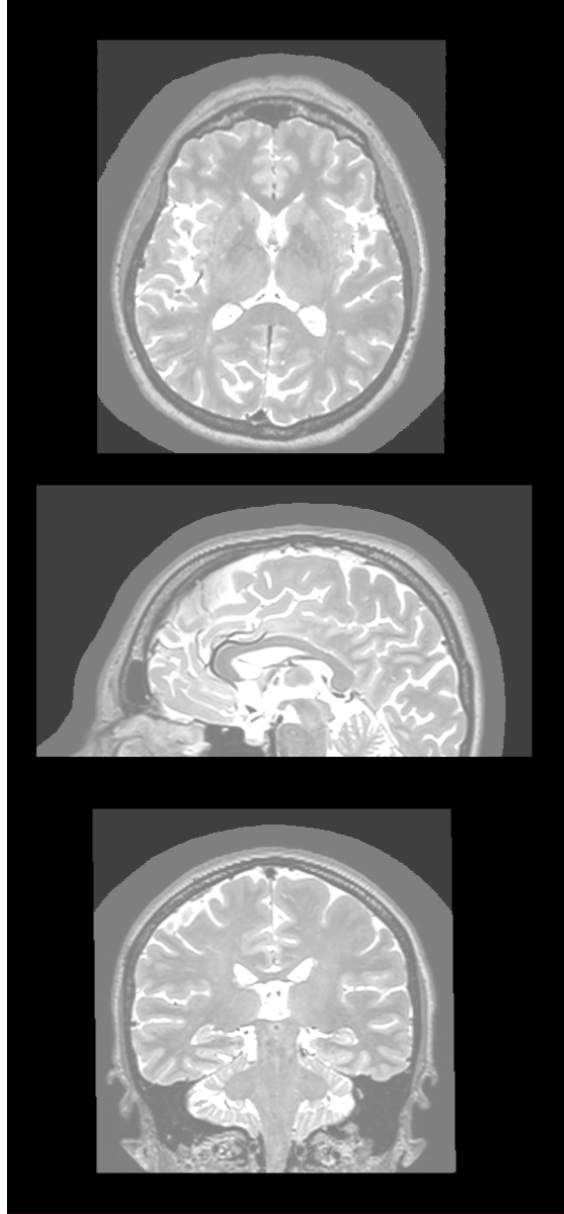


Figure 20: An example mask for one patient.

4.1.9 Robustness and reproducibility

Next, the robustness related to the initial misalignment of the final algorithm was tested. In order to test this, patient CT images of one patient were deflected from the semiautomatically aligned position with different angles presented in Table 7. The corresponding MMI values after registration are presented as well.

In Table 7, it is clearly visible that when at least one of the angles is 90 degrees, the registration algorithm can not align the images and is trapped to a wrong local optimum. The same happens if the two angles are 67.5 degrees. If the angles are 45 degrees or smaller and only one angle is 67.5 degrees, the algorithm performs the registration rather well. The combination 67.5, 67.5, 22.5 ended up in the wrong

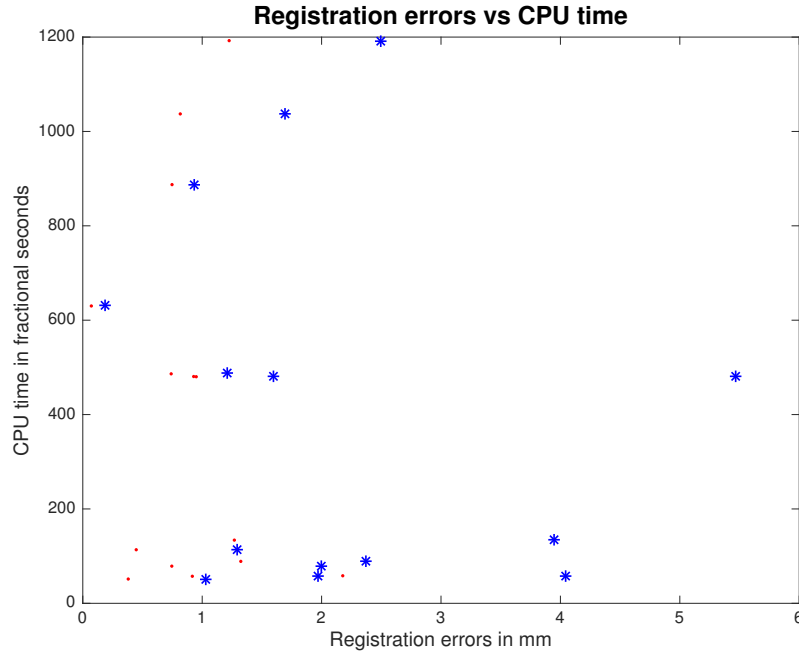


Figure 21: Registration errors with the final algorithm (with Euler transforms) for 14 patients versus the consumed CPU time. Mean registration errors are presented with red asterisks and maximum registration errors with blue asterisks. One patient with 90 degrees initial rotation was left out of results, since the algorithm could not register these images.

Table 6: Final metric values, registration errors (in mm) and CPU time (in fractional seconds) for 15 patients.

| Data set | Metric value | Mean TRE | STD TRE | Max TRE | Min TRE | CPU time |
|----------|--------------|----------|---------|---------|---------|----------|
| 1 | -0.41 | 2.18 | 0.83 | 4.04 | 0.15 | 58.37 |
| 2 | -0.32 | 1.27 | 0.91 | 3.95 | 0 | 133.91 |
| 3 | -0.37 | 0.92 | 0.5 | 1.97 | 0.01 | 57.32 |
| 4 | -0.22 | 0.74 | 0.19 | 1.2 | 0.06 | 486.33 |
| 5 | -0.27 | 0.93 | 0.24 | 1.59 | 0.07 | 480.61 |
| 6 | -0.3 | 0.07 | 0.05 | 0.18 | 0 | 630.16 |
| 7 | -0.28 | 1.23 | 0.55 | 2.49 | 0.03 | 1192.44 |
| 8 | -0.15 | 44.76 | 16.12 | 84.35 | 1.42 | 42.24 |
| 9 | -0.01 | 1.32 | 0.4 | 2.38 | 0.09 | 88.88 |
| 10 | -0.22 | 0.95 | 0.74 | 5.47 | 0.02 | 480.2 |
| 11 | -0.36 | 0.75 | 0.53 | 1.99 | 0 | 78.75 |
| 12 | -0.4 | 0.82 | 0.31 | 1.7 | 0.03 | 1037.17 |
| 13 | -0.35 | 0.38 | 0.27 | 1.03 | 0.01 | 51.46 |
| 14 | -0.28 | 0.75 | 0.09 | 0.94 | 0.06 | 887.33 |
| 15 | -0.38 | 0.45 | 0.31 | 1.29 | 0.01 | 113.54 |

Table 7: All initial angles and resulting MMI values which were tested to find the robustness threshold angles for the final algorithm.

| x | y | z | MMI |
|------|------|------|---------|
| 90 | 90 | 90 | -0,1802 |
| 90 | 90 | 67.5 | -0,1958 |
| 90 | 90 | 45 | -0,1936 |
| 90 | 90 | 22.5 | -0,1987 |
| 90 | 90 | 11.5 | -0,2060 |
| 67.5 | 67.5 | 67.5 | -0,3055 |
| 67.5 | 67.5 | 90 | -0,1948 |
| 67.5 | 67.5 | 45 | -0,3050 |
| 67.5 | 67.5 | 22.5 | -0,4127 |
| 67.5 | 67.5 | 11.5 | -0,3037 |
| 45 | 45 | 45 | -0,4168 |
| 45 | 45 | 90 | -0,1834 |
| 45 | 45 | 67.5 | -0,4106 |
| 45 | 45 | 22.5 | -0,4141 |
| 45 | 45 | 11.5 | -0,4082 |
| 22.5 | 22.5 | 22.5 | -0,4214 |
| 22.5 | 22.5 | 90 | -0,1786 |
| 22.5 | 22.5 | 67.5 | -0,4043 |
| 22.5 | 22.5 | 45 | -0,2999 |
| 22.5 | 22.5 | 11.5 | -0,4169 |
| 11.5 | 11.5 | 11.5 | -0,4098 |
| 11.5 | 11.5 | 90 | -0,1970 |
| 11.5 | 11.5 | 67.5 | -0,4084 |
| 11.5 | 11.5 | 45 | -0,4143 |
| 11.5 | 11.5 | 22.5 | -0,4110 |

local minimum.

Due to random sampling of the image, we tested the reproducibility of the registration for four patients. The sampling percentage was 0.5. For each patient we run the registration algorithm five times and the results can be seen in Figure 22. For one patient there was rather great variability in the final registration accuracy (patient 1 in Figure 22).

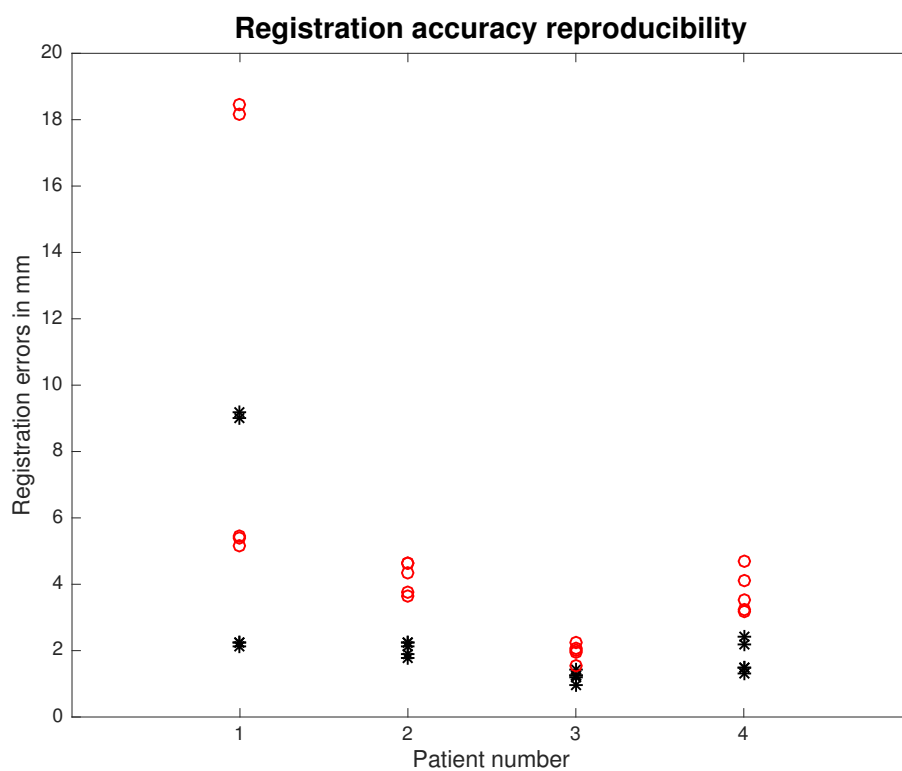


Figure 22: The reproducibility of registration accuracy for four patients. Maximum (red asterisks) and mean (black asterisks) errors are shown for repeated registrations.

4.2 Parameter optimization for versor transforms

Since *VersorRigid3DTransform* was beside *Euler3DTransform* another suitable transform for 3D rigid registration, separate analysis was made to optimize registration parameters with it. For the first two analyses of optimizer parameters and metric we used only five patients, one epilepsy patient, one Parkinson’s patient and three RIRE project datasets with initial angles z : -90 , x : 45 , y : $10-15$, z : -45 , x : 45 , y : -45 , z : -22.5 , x : 22.5 , y : 22.5 .

At first, optimal optimizer parameters were searched for Regular Step Gradient Descent optimizer, since it was the only suitable optimizer available in SimpleITK to be used with versor transforms (Johnson et al., 2015). Other parameters were set according to MICCAI 2015 example (*MICCAI 2015 : SimpleITK Registration*, 2015), except we used *VersorRigid3DTransform* and *CenteredVersorTransformInitializer*. The best optimizer parameter combination found was: learning rate 0.5, minimum step length $1e-3$, maximum number of iterations 300, gradient magnitude tolerance $1e-4$, relaxation factor 0.5.

Next we tested if joint histogram mutual information could be used with versor transforms instead of Mattes mutual information. JHMI did not perform registration at all in 3 out of 5 cases, thus we chose to use MMI in further analyses.

The best performing multiresolution level was searched for next. In this analysis we used data of 14 patients. As can be seen from Figure 23, strategies $[2, 2, 2]$ and $[3, 3, 2]$ are temporally most efficient, and when mean registration errors are considered, $[2, 2, 2]$ has the lowest mean value, 1.7356 mm, as the next lowest mean with $[3, 3, 2]$ is 1.9593 mm. Thus, multiresolution strategy $[2, 2, 2]$ was used in further analysis.

Next, the effect of different interpolators in registration on registration errors and CPU time consumption. The results for 14 patients are presented in Figure 24. Linear interpolation was considered to perform most efficiently.

The next phase of optimization was to drop the number of histogram bins used to compute the metric values, as well as the sampling percentage of image pixels used. It can be seen from Figure 25, that with combination of 12 bins and 10% of samples, the registration is successful. All number of bins and sampling percentages tested are presented in Table 5, and the resulting combinations in Appendix B. Therefore, 12 bins and 20% of samples were used in the final algorithm with versors. The sampling percentage was increased to confirm stability from 10 to 20. From Figure 26 it is visible that using 20% of samples does not increase the CPU time consumption extensively, so using it is reasonable.

The accuracies with the final algorithm using versor transforms are presented in Table 8.

Robustness of the final algorithm with versors is presented in Table 9 and the reproducibility of the algorithm is visible in Figure 27.

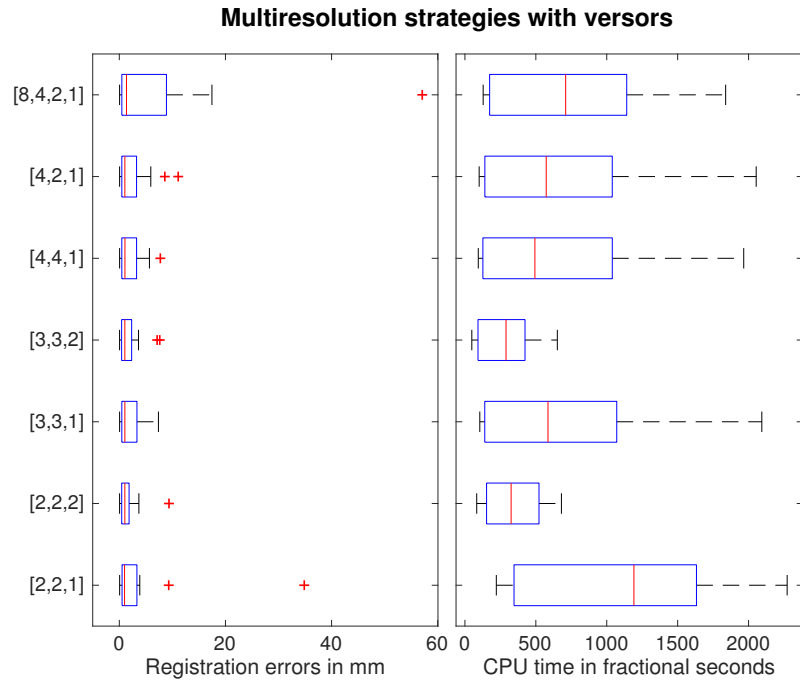


Figure 23: Registration errors and used CPU time with different multiresolution strategies with versor transforms.

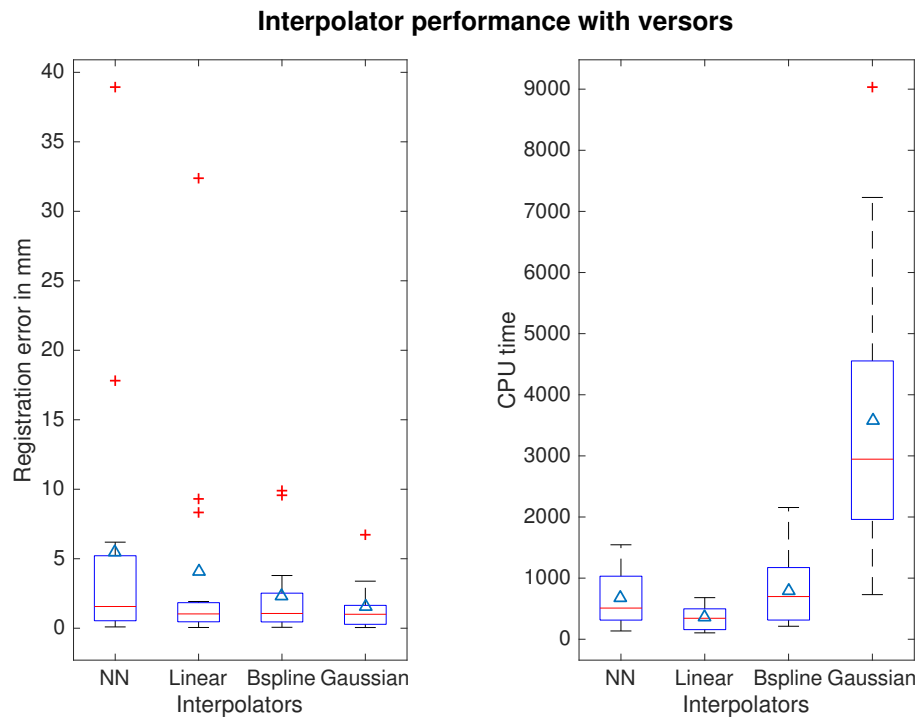


Figure 24: Registration errors and CPU time consumption with different interpolators.

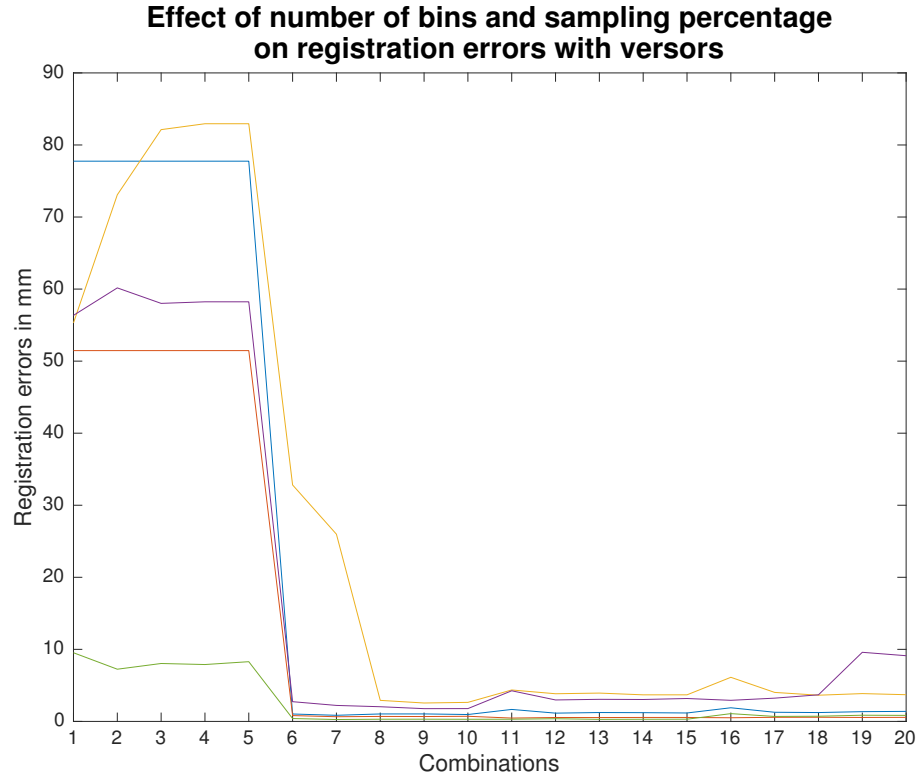


Figure 25: The effect of the combination of the number of bins and sampling percentage. Lines represent five patients.

Table 8: Accuracy of the final algorithm with versors for 15 patients.

| Data set | Metric value | TRE mean | TRE std | TRE max | TRE min | CPU Time |
|----------|--------------|----------|---------|---------|---------|----------|
| 1 | -0.45 | 1.89 | 0.57 | 3.38 | 0.1 | 59.5 |
| 2 | -0.34 | 1.67 | 1.19 | 5.32 | 0 | 117.31 |
| 3 | -0.4 | 1.09 | 0.5 | 2 | 0.07 | 54.07 |
| 4 | -0.23 | 0.63 | 0.23 | 1.24 | 0.04 | 421.92 |
| 5 | -0.28 | 0.72 | 0.09 | 0.92 | 0.06 | 412.32 |
| 6 | -0.3 | 0.09 | 0.06 | 0.26 | 0 | 365.17 |
| 7 | -0.28 | 1.11 | 0.56 | 2.54 | 0.04 | 699.48 |
| 8 | -0.16 | 43.17 | 16.17 | 85.36 | 1.42 | 42.44 |
| 9 | -0.01 | 1.71 | 0.61 | 3.23 | 0.13 | 72.16 |
| 10 | -0.23 | 0.93 | 0.76 | 3.79 | 0 | 404.19 |
| 11 | -0.39 | 0.73 | 0.51 | 1.84 | 0.01 | 53.21 |
| 12 | -0.41 | 1.04 | 0.48 | 2.45 | 0.02 | 535.55 |
| 13 | -0.38 | 0.39 | 0.12 | 0.64 | 0.02 | 55.43 |
| 14 | -0.25 | 6.27 | 3.33 | 13.45 | 0.31 | 603.83 |
| 15 | -0.42 | 0.32 | 0.22 | 0.88 | 0 | 99.18 |

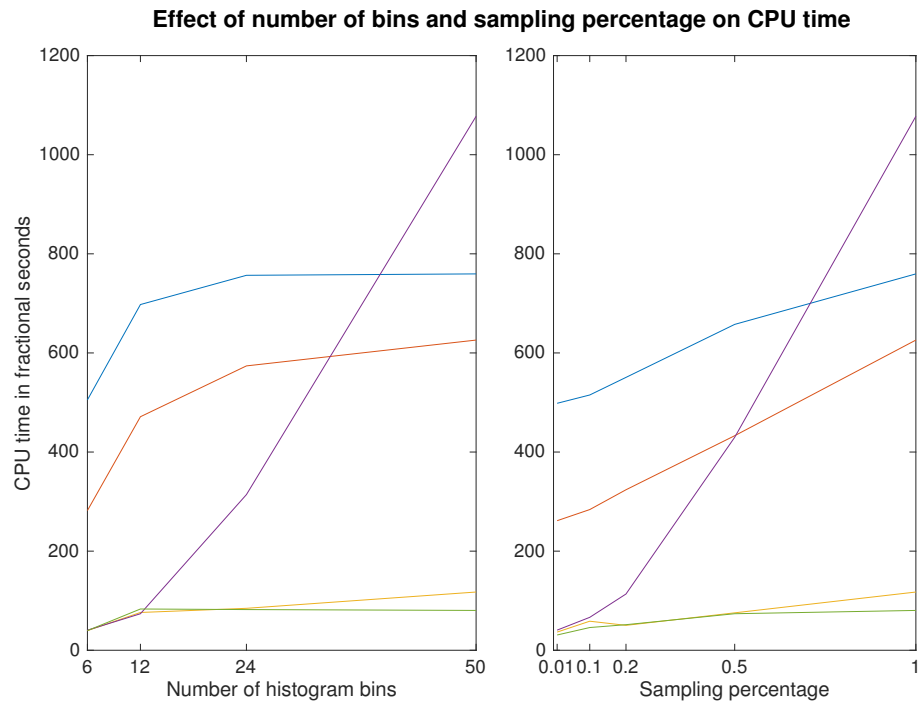


Figure 26: In this Figure, the effect of the number of histogram bins used to compute the metric value on CPU time consumption is presented on the left. On the right, the effect of sampling percentage of image pixels used to compute the metric on CPU time is presented. Lines represent five patients.

Table 9: All initial angles and resulting MMI values which were tested to find the robustness threshold angles for the algorithm with versors.

| x | y | z | MMI |
|------|------|------|-------|
| 90 | 90 | 90 | -0.19 |
| 90 | 90 | 67.5 | -0.19 |
| 90 | 90 | 45 | -0.18 |
| 90 | 90 | 22.5 | -0.20 |
| 90 | 90 | 11.5 | -0.17 |
| 67.5 | 67.5 | 67.5 | -0.32 |
| 67.5 | 67.5 | 90 | -0.19 |
| 67.5 | 67.5 | 45 | -0.32 |
| 67.5 | 67.5 | 11.5 | -0.32 |
| 67.5 | 67.5 | 22.5 | -0.30 |
| 45 | 45 | 45 | -0.45 |
| 45 | 45 | 90 | -0.19 |
| 45 | 45 | 67.5 | -0.45 |
| 45 | 45 | 22.5 | -0.45 |
| 45 | 45 | 11.5 | -0.45 |
| 22.5 | 22.5 | 22.5 | -0.45 |
| 22.5 | 22.5 | 90 | -0.08 |
| 22.5 | 22.5 | 67.5 | -0.45 |
| 22.5 | 22.5 | 45 | -0.45 |
| 22.5 | 22.5 | 11.5 | -0.45 |
| 11.5 | 11.5 | 11.5 | -0.45 |
| 11.5 | 11.5 | 90 | -0.07 |
| 11.5 | 11.5 | 67.5 | -0.45 |
| 11.5 | 11.5 | 45 | -0.44 |
| 11.5 | 11.5 | 22.5 | -0.45 |

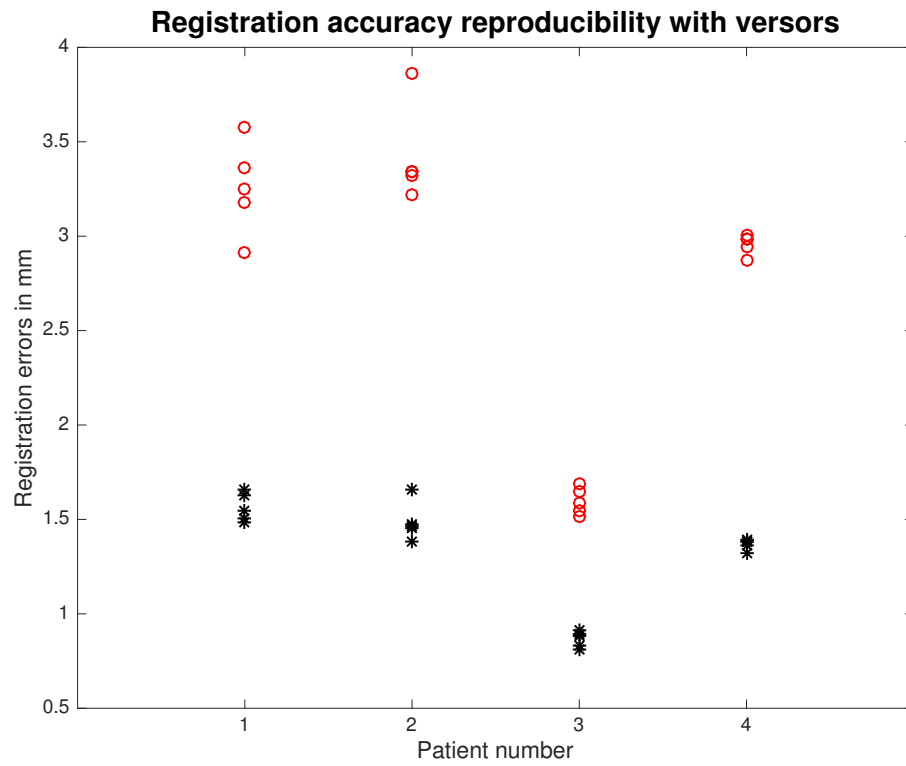


Figure 27: Reproducibility of registration results with versor transforms for four patients. The red asterisks represent maximum registration errors and black asterisks mean registration errors.

4.3 The final algorithm

The final algorithms with versor transforms and with Euler transforms are compared in Figure 28.

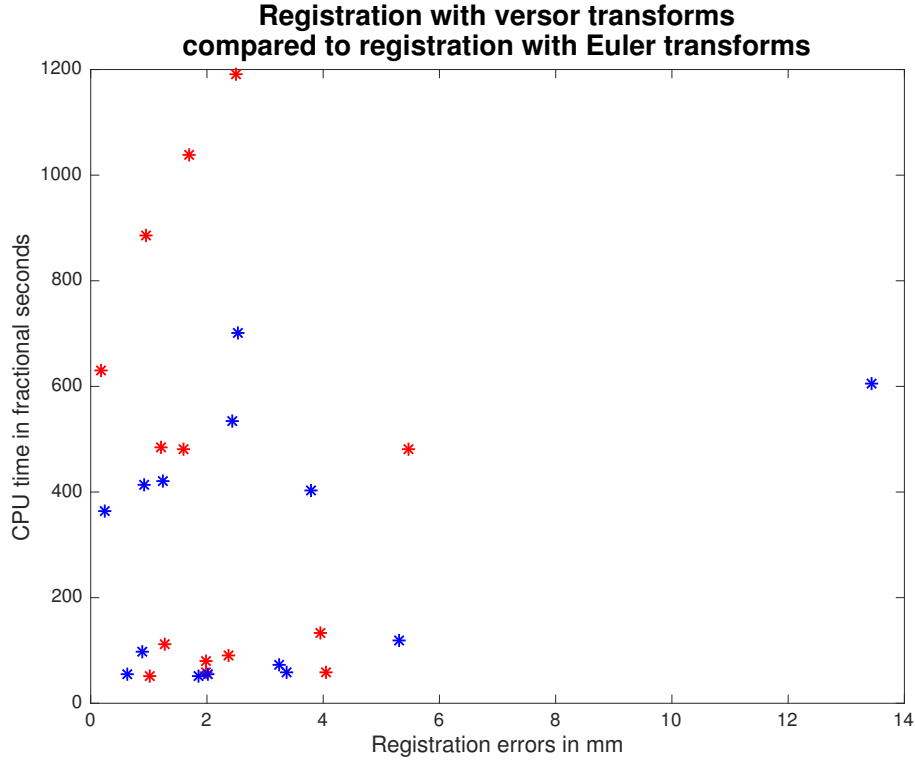


Figure 28: Comparison of the two algorithms. The maximum registration errors of algorithm with versors are presented with blue asterisks and the errors with Euler transforms with red asterisks.

With versors, with 7 out of 15 patients the maximum registration errors fell under 2 mm. With Euler transforms this ratio was 9 out of 15. From the visualization in Figure 28, data of one patient with 90 degrees initial rotation is left out, since neither of the final algorithms could perform the registration for that patient. Due to the faster performance, robustness and better reproducibility, the algorithm with versor transforms was chosen as the best performing algorithm.

In Figure 11 the initial displacement for one patient is shown. There is 45 degrees rotation about the x-axis, 10–15 degrees rotation about z- and y-axes and translation offset of 5–10 cm in z-axis. The final algorithm with Euler transforms aligned these images with the mean registration error of 0.38 mm. The resulting registered volumes are shown with the checker board filter in Figure 30.

In addition to optimized registration parameters, the final algorithm saves also a high resolution registered CT image. This is important in order to distinguish the exact location of recording or stimulating electrodes in relation to soft tissues visible in MR image. Therefore, we then tested the time allocation to different parts of the algorithm including resampling the CT image to MR image and resampling the CT

image to upsampled MR image with five patients. The high resolution MR image was sampled to have twice the resolution in all dimensions. It was noticed, that the CT image resampling for the first two patients took approximately 10 times more time compared to the last three (Table 10). In addition, for the first patient the high resolution resampling took approximately 10 times more time than lower resolution resampling, and for the second patient approximately 20 times more time.

The command line script was also extended to a 3D Slicer module, so that it can be used from the graphical user interface (Figure 31). Additionally, a loop to catch possible errors was added to the algorithm. The final program is able to save registered nifti images in DICOM format and send them straight to the to a DICOM compatible image archive.

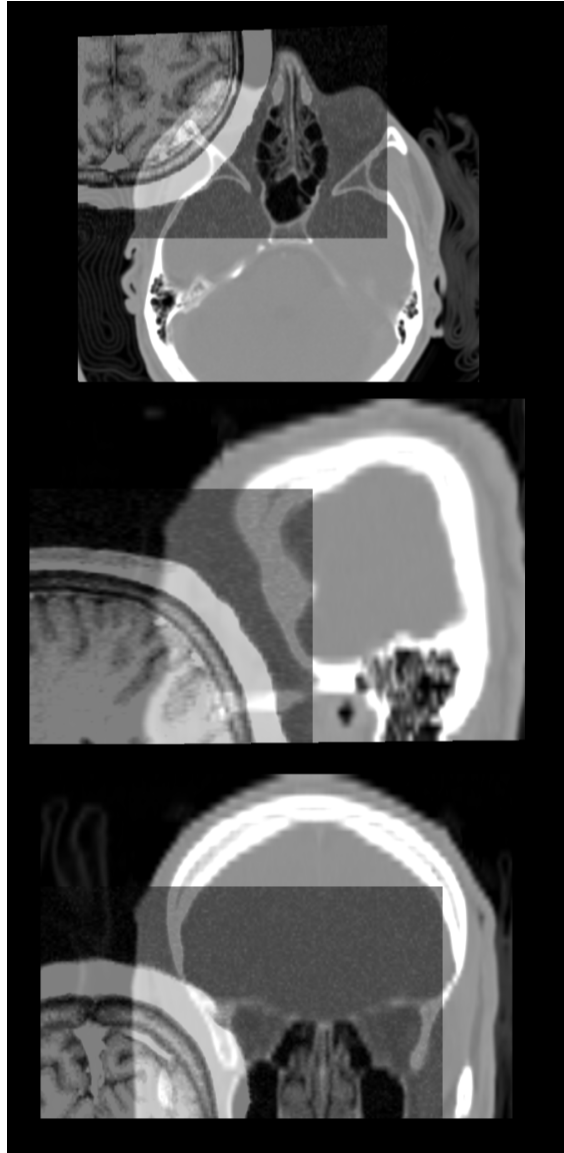


Figure 29: The initial rotated and transformed image before registration.

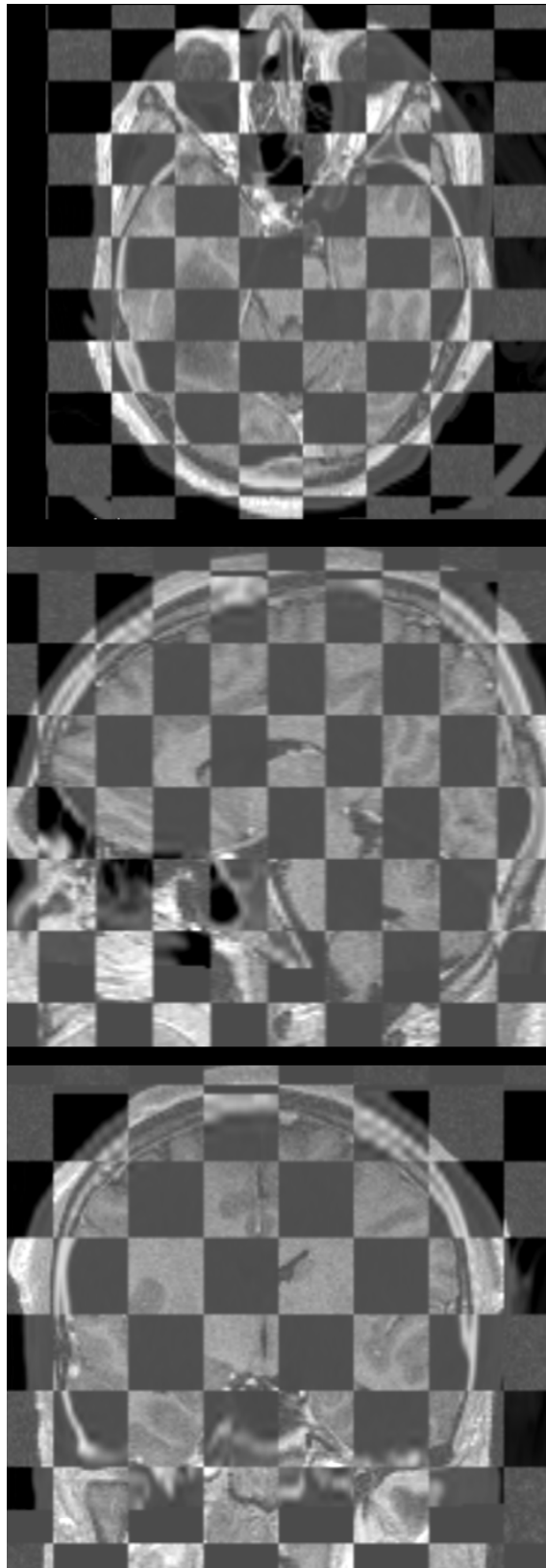


Figure 30: A checkerboard-filtered registered CT and MR images with transparency alpha value 0.5.

Table 10: CPU time consumption for different parts of the algorithm.

| Patient | Execution | Resampling | High resolution resampling |
|---------|-----------|------------|----------------------------|
| 1 | 1200.86 | 6.5 | 15503.95 |
| 2 | 507.33 | 5.95 | 10344.84 |
| 3 | 42.77 | 0.65 | 1.6 |
| 4 | 83.78 | 0.81 | 1.53 |
| 5 | 121.86 | 0.49 | 1.29 |

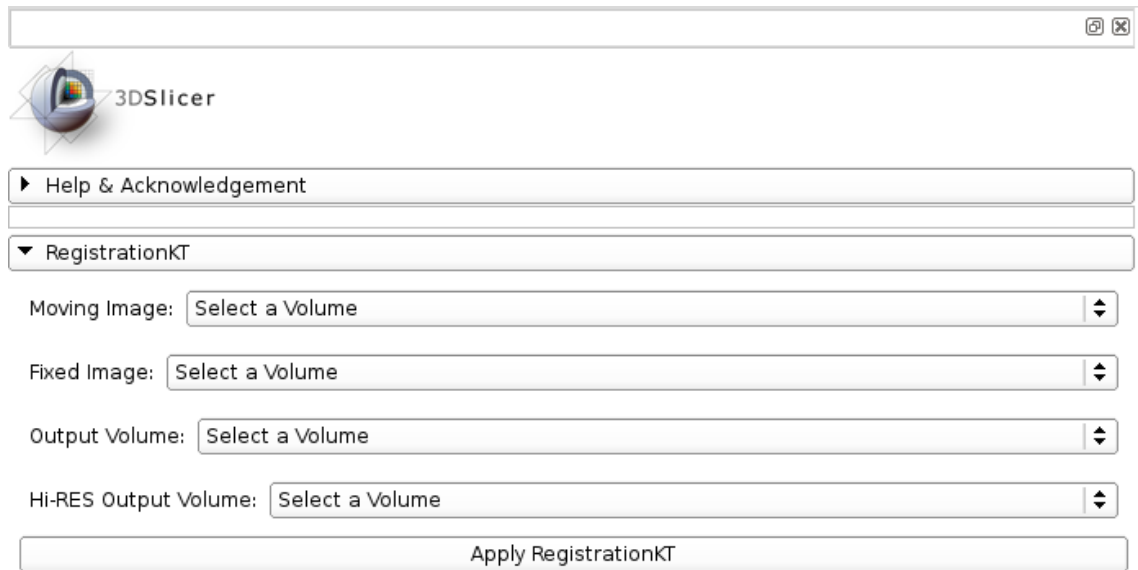


Figure 31: Screenshot of the 3D Slicer registration module.

5 Discussion

Due to the enormous number of possible registration parameter combinations, we decided to begin with the SimpleITK tutorial, which was used in an international conference of Medical Image Computing and Computer Assisted Interventions (MICCAI) 2015 (*MICCAI 2015 : SimpleITK Registration*, 2015). Having this tutorial as basis, we optimized it one block at a time. Euler transforms and versor transforms were handled in separate parameter optimization analyses. At first we discuss the optimization of parameters with Euler transforms, and thereafter with versor transforms. In the following sections, clinical applicability, limitations and future prospects are considered.

5.1 Parameter optimization

Since it is important to find a reasonable initial alignment in order to avoid optimizer getting stuck to a wrong local minimum of the metric value (*ITK Image Similarity Metrics*, 2016), we began with looking for the optimal initial transform to initialize the registration. The Metric evaluate method performed the best regarding only the final metric value, but it took almost three times more time to run compared to the centered transform initializer with geometry, which was chosen as the initial alignment method for the final algorithm. For some reason, the moments method brought the images outside the capture range of the optimizer, so that there were not enough overlapping samples to compute the metric value and the registration failed with the error message: *"Too many samples outside moving image buffer"*. The moments method is considered quite robust compared to geometry method, since it aligns the grey level mass centers of images. The advantage is obvious, in the case where the mass centers are not in the image center. However, moments method might not be valid in multimodal image registration, which could be the reason for the unsuccessful initialization in some cases (Johnson et al., 2015). The centered transform initializer with geometry was also used by Collignon and colleagues (1995a). Many algorithms, including the ipcWorkstation developed and used in HUS Medical Imaging Center, enable the use of point-based registration to initialize the registration (Eggert et al., 1997). However, these methods require user intervention, which is against the aim of this thesis to find an automated registration algorithm.

CGDLS and LBFGSB optimizers were left out of further analysis, since either we could not find any parameters which would perform the registration for even most of the patients, or the MMI values were not good enough compared to other optimizers. Exhaustive optimizer was also left out due to its substantial time consumption and its ability to find the alignment from only a finite number of solutions (*Understanding Optimization*, 2012). Compared to the article by Maes and colleagues (1999), the common optimizers were simplex based optimizer (Amoeba) and Gradient Descent optimizer. In addition to these optimizers we tested GDLS and RSGD optimizers. In their study there were no significant differences in performance between any optimizers tested. In our study, Figure 14 shows that with multiresolution strategy [4, 2, 1] for GDLS and with [2, 2, 2] for GD the registration errors are smaller than

with any of the strategies for other two optimizers.

With the parameter settings settled so far, we tried if the Joint Histogram metric would yield better registration results. However, we found out that it is rather unstable compared to Mattes mutual information with all four optimizers, at least with the these optimizer parameters optimized for MMI. MMI was also used as the similarity metric for optimization strategy comparison by Maes and colleagues (1999). In the analysis of multiresolution strategies we emphasized the registration accuracy more than CPU time consumption, since accuracy was considered more important for the final algorithm. As suspected, the strategy with one level was the most accurate one, but its CPU time consumption was multifold compared to other strategies, and too long for a registration algorithm to be used in clinical practice. Due to their best performance in accuracy, GDLS and GD were chosen for further analysis with strategies $[8, 4, 2, 1]$ and $[2, 2, 2]$, respectively. Maes and colleagues (1999) found that $[4, 2, 1]$ strategy was the most precise one. In our study, only with GD optimizer the $[4, 2, 1]$ strategy yielded accuracy close to to the mean accuracy of the best performing strategy for each optimizer. For RSGD and Amoeba $[4, 2, 1]$ was one of the worst performing strategies regarding only accuracy, and for GDLS it was not stable. The variation in registration accuracy was substantially wide for RSGD with all strategies, but its time consumption did not vary much between different multiresolution strategies. GD seems to be the most robust one, which was one of the desired features for our final registration algorithm.

Surprisingly, despite its simple computation nearest neighbor interpolation yielded smallest registration errors with GD optimizer. The most efficient interpolators in respect of time consumption are nearest neighbor and linear interpolation (Meijering et al., 1999). This is clearly visible in Figures 16 and 17. Especially with GDLS optimizer, b-spline and gaussian interpolators are substantially slower. B-spline interpolation with GD optimizer performs as fast as the nearest neighbor interpolation, but its accuracy is worse. With GDLS, b-spline and gaussian interpolators are left out of consideration due to their excessive time consumption. Maes and colleagues (1999) used partial volume interpolation in their study.

We assumed that with masking the fixed image we could reduce the number of samples needed to compute the metric and to find its optimal value. With GD optimizer, the algorithm performance did not improve with masking. However, this was only tested for one subject and in this case the original CT and MR images did not have a big translational or rotational differences. In the case, where there is a more notable misalignment to begin with, the problem with not enough overlapping samples after initialization could arise, causing the registration to fail using masking. It was also noted with GDLS optimizer, that with reduced number of histogram bins and sampling percentage, these problems with no adequate overlapping were quite frequent.

When testing the effect of the number of histogram bins on CPU time, it was noted that with 12, 24 and 50 bins, when the registration was successful with all image samples used to compute the metric value, there were no notable differences in CPU time consumption. Nevertheless, the CPU time used by the registration algorithm seems to be somewhat linearly dependent on the sampling percentage.

The value of the slope and the CPU time with sampling percentage of 0.01 appear to vary between subjects quite strongly. These differences could arise from differences in resolution and size of the images between patients, which both contribute to the total number of samples used to compute the metric during each iteration.

The MMI values are not comparable between subjects and resolution levels. They are only applicable as registration success measure, when final metric values within one patient are computed with same final subsampling factor, whilst other parameters such as the optimizer and interpolation can vary. Additionally, it should be noted that small MMI values can be achieved when the optimizer falls in the wrong local optimum. This is possible especially when there are great initial rotations. Assuming that these factors are taken into account, the smaller the MMI value, the more accurate the registration. Due to the computational power needed to calculate the registration errors, MMI values were used as a measure of registration success in the beginning of optimization process. In the end, we used registration errors as a measure of registration accuracy. It should be taken into account, that these registration errors are computed in comparison to semiautomatically registered images. If the error is around 1 mm, there is no way to tell, which is more accurate, the semiautomatic or the automatic registration. Only visual inspection with for example checker board filtering could help in recognizing, which transform is closer to optimal.

In the analysis of accuracy of the final algorithm with Euler transforms, for 9 out of 15 patients the maximal registration error was under 2 mm. However, the coordinates with which the error was computed did not cover the whole brain. In case of rotational misalignment after registration, there would be greater errors in the area of the skull than the values in Table 21. The maximal registration errors in registration of CT and MR images visible for human observer are around 2 mm (Fitzpatrick, Hill, et al., 1998).

It should be noted, that when using random sampling and sampling percentage notably lower than 100 %, the results might vary quite a lot between repeated registrations with same parameters for one subject. From Figure 22 it is clear that there is rather substantial variance in the mean registration errors between repeated registrations, which are for two patients less than 1 mm, for one patient about 2 mm, and for one patient around 7 mm. This suggests that for some patients the registration could sometimes fail merely because of the random samples falling to poor locations in respect of the similarity metric computation. In these situations I would suggest to run the algorithm again and see if they are improved after repeated registration. In parameter optimization 0.1% of samples was found to suffice, but in order to increase registration stability I decided to multiply it to 0.5% of the samples.

With versor transforms, initialization was done with *CenteredVersorTransformInitializer*. Since RSDG optimizer was the only suitable optimizer to be used with versor transforms (Johnson et al., 2015), it was chosen for further analysis. JHMI metric did not perform registration at all in most cases, thus MMI was used as similarity metric in final algorithm with versors. Since the [1, 1, 1] multiresolution strategy was so time consuming, it was not tested with versor transforms. Otherwise all strategies were tested according to Table 3. Not surprisingly, strategies which did not have

full resolution optimization were most efficient. As well as with Euler transforms, Gaussian interpolator CPU time consumption was substantially greater compared to other interpolators. With versor transforms there were stronger differences between patients in CPU time consumption than between patients with Euler transforms. However, this may result from using different patients in the analysis of the effect of number of bins and sampling percentage on CPU time.

The comparison of the accuracies and CPU time consumption of the two final algorithms is presented in Figure 28. In this Figure it seems that the algorithm with Euler transforms is slower. This probably results from greater sampling percentage used in the algorithm. Otherwise their performance seems rather similar. From the reproducibility Figures 22 and 27 can be seen, that the algorithm with versors produces more consistent results.

When comparing the robustness tables 7 and 9, it seems that with large initial angles the algorithm with versors performs better. For example with angles 45, 45, 67.5, the algorithm with versors performed notably better results with visual inspection in 3D Slicer. From Figure 28 it visually seems that the versor algorithm performs better. Due to its higher robustness and better reproducibility, it was chosen to be used in the final registration algorithm.

Since the CT images are usually taken with low dose, there is not much variation visible in the soft tissues of the brain and most information is in the anatomy of the skull (van Rooijen et al., 2013). Therefore, it is likely that the registration is considerably based on the skull, since its structures are visible in both MR and CT. Thus, it is possible that the more CT slices there are the better the registration results will be.

Since both metrics, Mattes mutual information and Joint Histogram mutual information, tested in this thesis work use Parzen windowing (Mattes et al., 2001; Thévenaz & Unser, 2000), there should be no problem with histogram binning (Pluim et al., 2003). This suggests, that windowing the image intensities before registration should not have substantial effect. However, we decided to include thresholding of CT images to 3000 HU, since there is no need to use those intensities coming from electrodes in histogram. Additionally, thresholding is computationally simple, thus it will not increase the overall computation time substantially.

Besides choosing the similarity metric and its parameters, the choice of optimizer and its parameters is important when tuning the registration parameters for a specific application, in our case the rigid registration of CT and MR images. One of the crucial parameters with optimizer is scaling. It was noted, that in most cases the automated scale estimation, either based on Jacobian norms or Physical shift, is the most suitable method. These methods allow changes to scaling after each iteration, which is a notable advantage compared to user defined permanent scales.

After the final algorithm with Euler transforms was implemented to 3D Slicer it was noticed, that in case the original images are almost aligned to begin with, the registration algorithm moved them into greater misalignment. This was assumed to result from the initial translation aligning the physical image centers. Therefore, a block was added to the final algorithm, which checks the metric value before and after initial alignment. If the registration initialization brings the moving images

further apart, the initial transform is set to identity. This was applied in the analysis of optimizer parameters with versors.

Since this check was not performed in the optimization of registration parameters with Euler transforms, it is possible that some options were left out because the optimizer could not bring the images back to capture range after initialization. However, according to Maes and colleagues (1999) there are no substantial differences between optimizers, thus it is likely that no better results would have been achieved even if this would have been taken into account already during the parameter optimization.

5.2 Clinical applicability

Since the aim was to create a fully automatic registration algorithm requiring no user interaction, the final program does not offer options in the registration parameters for the user. This facilitates the use of the registration in clinical routine and work flow. Furthermore, the graphical registration module added to the widely used medical image processing tool 3D Slicer simplifies the use even more. One of the objectives for the software was to be easy to use.

The main application of the optimized registration algorithm produced in the work of this thesis will be to register MRI and CT datasets of the epileptic patients undergoing Stereoelectroencephalography recordings as surgery candidates. The next frequently used application are probably the Parkinson's patients with stimulating electrodes implanted to their thalamus. For these patient groups it is crucial to have an accurate and robust registration program. The results from the final accuracy and robustness tests suggests that this thesis work provides an efficient registration tool to be used in clinical practice for HUS Medical Imaging Center.

It has been suggested that in case of postoperative morphological brain deformations, it would be advisable to use post-implantation MR images instead of CT images, so that these deformations would be visible (van Rooijen et al., 2013). However, the exact localization of electrodes from MR images is hard due to the high artifacts in the magnetic field. Therefore, they emphasize the importance of visual verification of the results, especially in the case of anatomical changes.

Along neuroradiologist Antti Korvenoja in HUS Medical Imaging Center (Korvenoja, 2016), patients' heads during scanning can be in rather great angles compared to the CT scanner axes. These angles could range up until almost 90° about z-axis and 45° about x- and y-axis. Our algorithm is stable until one angle of 67.5° , when other angles stay under 45° . This might limit the use with some challenging patients suffering for instance with stroke.

Studholme and colleagues (1996) suggest, that there is variance in the importance of different anatomical features existing in the images and the presence of the skull results in greatest robustness of the registration algorithm. Mutual information was the best performing voxel similarity measure beside soft tissue correlation with gradient descent optimizer and multiresolution strategy. According to Collignon and colleagues (1995a) there is never a perfect registration, not even in the aligned position due to noise and simply the nature of multimodal images having different intensities.

For instance skull and background in MR images have partially overlapping intensities, while in CT they are completely different (Collignon et al., 1995a). In addition CT and MR images never fully overlap, since CT usually has fewer slices due to exposure to radiation as opposed to MR, where the whole volume is imaged. Hence, the difference in the number of slices, i.e. also the difference in anatomical features existing in the images could effect on the registration accuracy.

5.3 Limitations

Since it is not possible to test all available parameter combinations, at lest with the framework of this thesis and available computational resources, the final algorithm might not include the absolute optimal parameters for this application. The optimization of registration parameters was performed in blocks, which results in multiple parameter combinations left untested. For instance, the optimizer parameters as well as the intialization were optimized for Mattes mutual information. If they were optimized for Joint Histogram mutual information, the performance of JHMI as similarity metric could have been better.

The robustness to rotations was defined only with combinations of five angles. The accurate threshold for initial angles could have been solved more precisely. Furthermore, if the semiautomatic registration could have been performed with all rotations, the threshold for robustness would have been easier to interpret.

Without masking the algorithm takes image samples also outside the head. It would be more efficient, if all the metric values could be computed inside the skull, which is the structure with most mutual information between low dose CT and MR images. Therefore, masking could be tested further to find optimal sampling in this special case. However, it is possible that the successful registration with mutual information with low dose CT images might require at least some of the surrounding air. This could explain the poor results with masking.

For individual cases it could have been possible to fine tune the registration parameters even further. ITK v4 offers a class for monitoring registration (Johnson et al., 2015). It enables careful monitoring of the registration progression. Unfortunately, in the framework of this thesis this was not possible, since the parameter optimization had to be mostly automatic to save time.

In order to compute the registration errors, a grid image with ascending intensities was created in the middle of the CT images. This grid does not cover the whole brain and skull. This results in reduced maximum registration errors compared to the actual errors, since the errors introduced by rotation are greatest on the scalp. A more sophisticated algorithm for registration error computation would be required to compute the errors with equal distribution everywhere inside the head.

It would be desirable to have a testing script, which could be run after version updates and such, in order to verify that the algorithm produces similar results compared to previous versions. However, there is no command for re-initialization of random seeds in SimpleITK, which results in different metric values and number of iterations every time the algorithm is run. Therefore, we were not able to produce an exact testing script for the registration algorithm.

The semi-automatically created golden standard was regarded as the ground truth. We computed the registration errors based on this semi-automatic transform. However, this golden standard is not the ground truth. The true targets in SEEG are the electrode locations. Therefore, in order to follow the definition of TRE (Fitzpatrick, West, & Maurer Jr, 1998), they should have been computed with those locations instead of grid points. Additionally, TREs estimated from the golden standard assumes that the rigid registration model is valid through the brain. This is not the case, since there are some morphological changes due to swelling for instance, which results from the implantation of electrodes.

5.4 Future prospects

The first thing to advance the method in this thesis is to make it more efficient. At the moment, the whole process including high resolution resampling takes several minutes. In addition, even though the algorithm sends the registered images automatically to a DICOM compatible image archive, it is not fully compatible with the system. It would be desirable to import the image volumes straight from the image archive for processing with the algorithm. Furthermore, the import could be automatic so that it would recognize the image volumes in need of registration. This would not need any user intervention.

In this thesis the aim was to optimize the registration parameters for CT and MR images of the head. A possible extension of this work could be including functional images, such as PET images to the algorithm. An even further extension would be that the algorithm could first identify possible need for affine transformation or even free form transformation accordingly. However, with free form transforms there is the problem of restricting the number of possible solutions and evaluating the results.

One of the future extensions to the work in this thesis could be to add a block of code, which could at least detect possible gantry tilt in CT images and apply an affine transformation to correct it. Gantry tilt can be a very small or even 30 degrees angle tilted from the normal position of the X-ray tube and detector gantry perpendicular to the patient z-axis (Bibb et al., 2014). CT DICOM headers have all the required information to take the gantry tilt into account. Some of the available software just use wrong tags (e.g. DICOM standard explicitly forbids the use of Gantry Tilt tag in any computations). This could result in some serious problems in image registration, when CT image would be distorted due to gantry tilt. Instead, the ImagePosition and ImageOrientation tags should be used. Affine transforms can take care of these distortions. Therefore, the gantry tilt extension could first recognize the angle and then either inform the user of the distortion and cease the algorithm or use affine transform to register the images.

This registration algorithm was constructed with SimpleITK. Since all ITK v4 optimizers and transforms etc. are not available in SimpleITK, using SimpleITK there are less possible parameters to test in parameter optimization. In the future, the same algorithm could be optimized for ITK with additional parameters provided with ITK v4.

6 Conclusions

As a result of this thesis work, a fully automated image registration algorithm was produced. Registration parameters were optimized for aligning CT and MR images of the head. The main applications of this algorithm in HUS Medical Imaging Center are to register multimodal images of epilepsy and Parkinson’s patients. In the case of epilepsy patients, pre-operative MR images and post-implantation CT images with stereoelectroencephalography electrodes have to be aligned to plan the surgical removal of brain tissue. In the case of Parkinson’s patients, pre-operative MR images have to be registered with post-implantation CT images with deep brain stimulation electrodes to find out the exact location of stimulating electrodes. The advantage of the algorithm produced in this thesis work is, besides automation, its ease of use for users with no image registration expertise.

The algorithm is rather robust, providing a rough registration result even with more difficult cases, such as large initial rotations. However, in these cases the result might not be accurate enough to be used in determining exact soft tissue regions to be removed in epileptic surgery. Therefore, it is always crucial to verify the results after registration. In most cases, the algorithm performs accurate registration for clinical purposes and outperforms the present registration software used in HUS Medical Imaging Center.

The temporal performance of the algorithm is highly dependent on individual cases. In addition to the actual registration process, resampling the high resolution CT image is rather time consuming, especially with high resolution CT images. If high resolution CT images are not needed, this part can be removed from the Python code. Editing the sampling percentage to 100% could yield better results in difficult cases with large initial rotations for example. If the registration results are not satisfactory with versor transforms, the algorithm with Euler transforms could be used.

In the future, this block-wise pattern of optimization could be used to optimize registration parameters for other individual applications as well. In addition to rigid registration, deformable registration parameters could be optimized for instance for multimodal images from abdomen. However, this might require ITK instead of SimpleITK in order to get access to the full range of registration parameters.

References

- Avants, B. B., Tustison, N., & Song, G. (2009). Advanced normalization tools (ANTS). *Insight Journal*, 2, 1–35.
- Avants, B. B., Tustison, N. J., Stauffer, M., Song, G., Wu, B., & Gee, J. C. (2014a). The insight toolkit image registration framework. *Frontiers in Neuroinformatics*.
- Avants, B. B., Tustison, N. J., Stauffer, M., Song, G., Wu, B., & Gee, J. C. (2014b). The insight toolkit image registration framework. *Frontiers in Neuroinformatics*.
- Benabid, A., Pollak, P., Louveau, A., Henry, S., & De Rougemont, J. (1988). Combined (thalamotomy and stimulation) stereotactic surgery of the VIM thalamic nucleus for bilateral Parkinson disease. *Stereotactic and Functional Neurosurgery*, 50(1–6), 344–346.
- Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 289–300.
- Bibb, R., Eggbeer, D., & Paterson, A. (2014). *Medical modelling: The application of advanced design and rapid prototyping techniques in medicine*. Woodhead Publishing.
- Boehler, T., van Straaten, D., Wirtz, S., & Peitgen, H.-O. (2011). A robust and extendible framework for medical image registration focused on rapid clinical application deployment. *Computers in Biology and Medicine*, 41(6), 340–349.
- Byrd, R. H., Lu, P., Nocedal, J., & Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5), 1190–1208.
- Cierniak, R. (2011). *X-Ray computed tomography in biomedical engineering*. Springer Science & Business Media.
- Collignon, A., Maes, F., Delaere, D., Vandermeulen, D., Suetens, P., & Marchal, G. (1995a). Automated multi-modality image registration based on information theory. In *Information Processing in Medical Imaging* (Vol. 3, pp. 263–274).
- Collignon, A., Vandermeulen, D., Suetens, P., & Marchal, G. (1995b). 3D multi-modality medical image registration using feature space clustering. In *Computer Vision, Virtual Reality and Robotics in Medicine* (pp. 195–204).
- Deep-Brain Stimulation for Parkinson’s Disease Study Group and others. (2001). Deep-brain stimulation of the subthalamic nucleus or the pars interna of the globus pallidus in Parkinson’s disease. *The New England Journal of Medicine*, 345(13), 956.

- Dickhaus, H., Flocq, R., Eisenmann, U., Metzner, R., & Wirtz, C. (2004). A flexible registration framework for multimodal image data. In *Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE* (Vol. 1, pp. 1755–1758).
- Duda, R. O., Hart, P. E., et al. (1973). *Pattern classification and scene analysis* (Vol. 3). Wiley New York.
- D'Albis, T., Haegelen, C., Essert, C., Fernández-Vidal, S., Lalys, F., & Jannin, P. (2015). PyDBS: an automated image processing workflow for deep brain stimulation surgery. *International Journal of Computer Assisted Radiology and Surgery*, 10(2), 117–128.
- Eggert, D. W., Lorusso, A., & Fisher, R. B. (1997). Estimating 3-D rigid body transformations: a comparison of four major algorithms. *Machine Vision and Applications*, 9(5-6), 272–290.
- Fedorov, A., Beichel, R., Kalpathy-Cramer, J., Finet, J., Fillion-Robin, J.-C., Pujol, S., ... others (2012). 3D Slicer as an image computing platform for the quantitative imaging network. *Magnetic Resonance Imaging*, 30(9), 1323–1341.
- Fitzpatrick, J. M., Hill, D. L., Shyr, Y., West, J., Studholme, C., & Maurer, C. R. (1998). Visual assessment of the accuracy of retrospective registration of MR and CT images of the brain. *Medical Imaging, IEEE Transactions on*, 17(4), 571–585.
- Fitzpatrick, J. M., West, J. B., & Maurer Jr, C. R. (1998). Predicting error in rigid-body point-based registration. *Medical Imaging, IEEE Transactions on*, 17(5), 694–702.
- Flocq, R., & Dickhaus, H. (2007). A flexible registration and evaluation engine (free). *Computer Methods and Programs in Biomedicine*, 87(2), 81–92.
- Hamilton, W. (1969). Elements of quaternions, 1899–1901 enlarged by CJ Joly, reprinted by Chelsea Publ. New York.
- Hill, D. L., Batchelor, P. G., Holden, M., & Hawkes, D. J. (2001). Medical image registration. *Physics in Medicine and Biology*, 46(3), R1.
- Hounsfield, G. N. (1980). Nobel lecture, 8 december 1979. computed medical imaging. *Journal de Radiologie*, 61(6-7), 459.
- Hounsfield unit. (2016). <http://www.fpnotebook.com/rad/CT/HnsfldUnt.htm>. (Accessed: 2016-05-07)
- Hsieh, J. (2009). Computed tomography: principles, design, artifacts, and recent advances..
- Ibanez, L., Schroeder, W., Ng, L., & Cates, J. (2003). The ITK software guide.

- Insight segmentation and registration toolkit.* (2016). <http://www.itk.org/Doxygen/html/index.html>. (Accessed: 2016-03-22)
- [*insight-users*]. (2007). <https://cmake.org/pipermail/insight-users/2007-March/021435.html>. (Accessed: 2016-02-16)
- Itk image registration.* (2010). <http://www.itk.org/Wiki/ITK/ImageRegistration>. (Accessed: 2016-03-22)
- Itk image similarity metrics.* (2016). <http://www.itk.org/Doxygen/html/ImageSimilarityMetricsPage.html>. (Accessed: 2016-03-10)
- [*itk*]/[*itk-dev*]*optimizing composite transforms and center of transform.* (2014). <http://public.kitware.com/pipermail/community/2014-July/006841.html>. (Accessed: 2016-03-10)
- Jenkinson, M., & Smith, S. (2001). A global optimisation method for robust affine registration of brain images. *Medical Image Analysis*, 5(2), 143–156.
- Johnson, H. J., Harris, G., Williams, K., et al. (2007). BRAINSFit: mutual information rigid registrations of whole-brain 3D images, using the insight toolkit. *Insight Journal*, 1–10.
- Johnson, H. J., McCormick, M. M., & Ibanez, L. (2015). The ITK software guide book 2: Design and functionality fourth edition updated for ITK version 4.9.
- Klein, S., Staring, M., Murphy, K., Viergever, M. A., & Pluim, J. P. (2010). Elastix: a toolbox for intensity-based medical image registration. *Medical Imaging, IEEE Transactions on*, 29(1), 196–205.
- Korvenoja, A. P. (2016). *Possible patient orientations*. Personal communication. (Discussion on: 2016-02-01)
- Loweckamp, B. C., Chen, D. T., Ibáñez, L., & Blezek, D. (2013). The design of SimpleITK. *Frontiers Neuroinformatics*, 7(45).
- Maes, F., Collignon, A., Vandermeulen, D., Marchal, G., & Suetens, P. (1997). Multimodality image registration by maximization of mutual information. *Medical Imaging, IEEE Transactions on*, 16(2), 187–198.
- Maes, F., Vandermeulen, D., & Suetens, P. (1999). Comparative evaluation of multiresolution optimization strategies for multimodality image registration by maximization of mutual information. *Medical Image Analysis*, 3(4), 373–386.
- Maintz, J. A., & Viergever, M. A. (1998). A survey of medical image registration. *Medical Image Analysis*, 2(1), 1–36.
- Mattes, D., Haynor, D. R., Vesselle, H., Lewellen, T. K., & Eubank, W. (2003). PET-CT image registration in the chest using free-form deformations. *Medical Imaging, IEEE Transactions on*, 22(1), 120–128.

- Mattes, D., Haynor, D. R., Vesselle, H., Lewellyn, T. K., & Eubank, W. (2001). Nonrigid multimodality image registration. In *Medical Imaging 2001* (pp. 1609–1620).
- McRobbie, D. W., Moore, E. A., Graves, M. J., & Prince, M. R. (2007). *MRI from picture to proton*. Cambridge University Press.
- Meijering, E. H., Niessen, W. J., Pluim, J. P., & Viergever, M. A. (1999). Quantitative comparison of sinc-approximating kernels for medical image interpolation. In *Medical image computing and computer-assisted intervention–MICCAI’99* (pp. 210–217).
- MICCAI 2015 : SimpleITK registration*. (2015). <http://www.itk.org/Wiki/SimpleITK/Tutorials/MICCAI2015>. (Accessed: 2016-02-16)
- Modat, M., Ridgway, G. R., Taylor, Z. A., Lehmann, M., Barnes, J., Hawkes, D. J., ... Ourselin, S. (2010). Fast free-form deformation using graphics processing units. *Computer Methods and Programs in Biomedicine*, 98(3), 278–284.
- Modersitzki, J. (2009). *FAIR: flexible algorithms for image registration* (Vol. 6). SIAM.
- Nejati, M., & Pourghassem, H. (2014). Multiresolution image registration in digital X-Ray angiography with intensity variation modeling. *Journal of Medical Systems*, 38(2), 1–10.
- Nelder, J. A., & Mead, R. (1965). A simplex method for function minimization. *The Computer Journal*, 7(4), 308–313.
- Odekerken, V. J., van Laar, T., Staal, M. J., Mosch, A., Hoffmann, C. F., Nijssen, P. C., ... Contarino, M. F. (2013). Subthalamic nucleus versus globus pallidus bilateral deep brain stimulation for advanced Parkinson’s disease (NSTAPS study): a randomised controlled trial. *The Lancet Neurology*, 12(1), 37–44.
- Pieper, S., Halle, M., & Kikinis, R. (2004). 3D slicer. In *Biomedical Imaging: Nano to macro, 2004. IEEE International Symposium on* (pp. 632–635).
- Pluim, J. P., Maintz, J. A., Viergever, M., et al. (2003). Mutual-information-based registration of medical images: a survey. *Medical Imaging, IEEE Transactions on*, 22(8), 986–1004.
- Pluim, J. P., Maintz, J. A., & Viergever, M. A. (2000). Interpolation artefacts in mutual information-based image registration. *Computer Vision and Image Understanding*, 77(2), 211–232.
- The Python standard library*. (2016). <https://docs.python.org/3/library/time.html#module-time>. (Accessed: 2016-04-15)
- Schroeder, K., W. Martin. (2006). *The visualization toolkit an object-oriented approach to 3D graphics, kitware*. Kitware.

- Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27, 379–423, 623–656.
- SimpleITK notebooks*. (2016). <http://insightsoftwareconsortium.github.io/SimpleITK-Notebooks/>. (Accessed: 2016-03-07)
- Škerl, D., Likar, B., Fitzpatrick, J. M., & Pernuš, F. (2007). Comparative evaluation of similarity measures for the rigid registration of multi-modal head images. *Physics in Medicine and Biology*, 52(18), 5587.
- Studholme, C., Hill, D. L., & Hawkes, D. J. (1995). Multiresolution voxel similarity measures for MR-PET registration. In *Information Processing in Medical Imaging* (Vol. 3, pp. 287–298).
- Studholme, C., Hill, D. L., & Hawkes, D. J. (1996). Automated 3-D registration of MR and CT images of the head. *Medical Image Analysis*, 1(2), 163–175.
- Studholme, C., Hill, D. L., & Hawkes, D. J. (1999). An overlap invariant entropy measure of 3D medical image alignment. *Pattern Recognition*, 32(1), 71–86.
- Thévenaz, P., & Unser, M. (2000). Optimization of mutual information for multiresolution image registration. *Image Processing, IEEE Transactions on*, 9(12), 2083–2099.
- Understanding optimization*. (2012). https://www.multicharts.com/trading-software/index.php/Understanding_Optimization.
- Unser, M., Aldroubi, A., & Eden, M. (1993). B-spline signal processing. I. Theory. *Signal Processing, IEEE Transactions on*, 41(2), 821–833.
- van Rooijen, B. D., Backes, W. H., Schijns, O. E., Colon, A., & Hofman, P. A. (2013). Brain imaging in chronic epilepsy patients after depth electrode (stereo-electroencephalography) implantation: magnetic resonance imaging or computed tomography? *Neurosurgery*, 73(3), 543–549.
- Viola, P., & Wells III, W. M. (1997). Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2), 137–154.
- Vlissides, J., Helm, R., Johnson, R., & Gamma, E. (1995). Design patterns: Elements of reusable object-oriented software. *Reading: Addison-Wesley*, 49(120), 11.
- Webb, S. (1992). Historical experiments predating commercially available computed tomography*. *The British Journal of Radiology*, 65(777), 835–837.
- Wells, W. M., Viola, P., Atsumi, H., Nakajima, S., & Kikinis, R. (1996). Multi-modal volume registration by maximization of mutual information. *Medical Image Analysis*, 1(1), 35–51.

- Wolf, I., Vetter, M., Wegner, I., Böttger, T., Nolden, M., Schöbinger, M., . . . Meinzer, H.-P. (2005). The medical imaging interaction toolkit. *Medical Image Analysis*, 9(6), 594–604.
- Woods, R. P., Grafton, S. T., Holmes, C. J., Cherry, S. R., & Mazziotta, J. C. (1998). Automated image registration: I. General methods and intrasubject, intramodality validation. *Journal of Computer Assisted Tomography*, 22(1), 139–152.
- Woods, R. P., Grafton, S. T., Watson, J. D., Sicotte, N. L., & Mazziotta, J. C. (1998). Automated image registration: II. Intersubject validation of linear and nonlinear models. *Journal of Computer Assisted Tomography*, 22(1), 153–165.
- Woods, R. P., Mazziotta, J. C., Cherry, S. R., et al. (1993). MRI-PET registration with automated algorithm. *Journal of Computer Assisted Tomography*, 17(4), 536–546.
- Yushkevich, P. A., Piven, J., Cody Hazlett, H., Gimpel Smith, R., Ho, S., Gee, J. C., & Gerig, G. (2006). User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. *Neuroimage*, 31(3), 1116–1128.
- Zhu, Y.-M., & Cochoff, S. M. (2002). Influence of implementation parameters on registration of MR and SPECT brain images by maximization of mutual information. *Journal of Nuclear Medicine*, 43(2), 160–166.

Appendices

A Metric evaluate -method angles

Table 11: All angles used in Metric evaluate -method in intial alignment.

| x | y | z |
|-----|---|-----|
| 0 | 0 | 90 |
| 0 | 0 | -90 |
| 0 | 0 | 45 |
| 0 | 0 | -45 |
| 90 | 0 | 0 |
| -90 | 0 | 0 |
| 45 | 0 | 0 |
| -45 | 0 | 0 |
| 90 | 0 | 90 |
| 90 | 0 | -90 |
| -90 | 0 | 90 |
| -90 | 0 | -90 |
| 45 | 0 | 90 |
| 45 | 0 | -90 |
| -45 | 0 | 90 |
| -45 | 0 | -90 |
| 45 | 0 | 45 |
| 45 | 0 | -45 |
| -45 | 0 | 45 |
| -45 | 0 | -45 |
| 90 | 0 | 45 |
| -90 | 0 | 45 |
| 90 | 0 | -45 |
| -90 | 0 | -45 |

B Bin number and sampling percentage combinations

| Combination number | Number of histogram bins | Sampling percentage |
|--------------------|--------------------------|---------------------|
| 1 | 6 | 0.01 |
| 2 | 6 | 0.1 |
| 3 | 6 | 0.2 |
| 4 | 6 | 0.5 |
| 5 | 6 | 1.0 |
| 6 | 12 | 0.01 |
| 7 | 12 | 0.1 |
| 8 | 12 | 0.2 |
| 9 | 12 | 0.5 |
| 10 | 12 | 1.0 |
| 11 | 24 | 0.01 |
| 12 | 24 | 0.1 |
| 13 | 24 | 0.2 |
| 14 | 24 | 0.5 |
| 15 | 24 | 1.0 |
| 16 | 50 | 0.01 |
| 17 | 50 | 0.1 |
| 18 | 50 | 0.2 |
| 19 | 50 | 0.5 |
| 20 | 50 | 1.0 |

C Table of ITK parameters

The following table presents all ITK and SimpleITK parameters, including the ITK v3 legacy metrics. Bolded parameters are also available in SimpleITK.

| ITK PARAMETERS | DESCRIPTION |
|--------------------------------------|--|
| TRANSFORMS | |
| Identity Transform/ Transform | In SimpleITK a generic transformation, the default is identity transformation. |
| Translation Transform | 2D or 3D, translation |
| Scale Transform | 2D or 3D, anisotropic scaling |
| Scale Logarithmic Transform | Logarithmic Scale transformation of a vector space |
| Euler2DTransform | 2D, rigid transformation with rotation represented by a Euler angle |
| CenteredRigid2DTransform | Transform applies a rigid transformation is 2D space |
| Similarity2DTransform | 2D, composition of isotropic scaling and rigid transformation with rotation represented by a Euler angle |
| QuaternionRigidTransform | Transform applies a rotation and translation to the space given a quaternion and a 3D translation |
| VersorTransform | 3D, rotation represented by a versor |
| VersorRigid3DTransform | 3D, rigid transformation with rotation represented by a versor |
| Euler3DTransform | 3D, rigid transformation with rotation represented by Euler angles |
| Similarity3DTransform | 3D, composition of isotropic scaling and rigid transformation with rotation represented by a versor |

| | |
|--|---|
| <code>Rigid3DPerspectiveTransform</code> | 3D rigid transformation followed by a perspective projection |
| AffineTransform | 2D or 3D, affine transformation. |
| BSplineDeformableTransform | 2D or 3D, deformable transformation represented by a sparse regular grid of control points. |
| <code>KernelTransforms</code> | Transform used with elastic body splines and thin plate splines (ElasticBodySplineKernelTransform, ElasticBodyReciprocalSplineKernelTransform, ThinPlateSplineKernelTransform, ThinPlateR2LogRSplineKernelTransform, VolumeSplineKernelTransform) |
| ScaleVersor3DTransform | 3D, rigid transformation and anisotropic scale is added to the rotation matrix part (not composed as one would expect) |
| ScaleSkewVersor3DTransform | 3D, rigid transformation with anisotropic scale and skew matrices added to the rotation matrix part (not composed as one would expect) |
| DisplacementFieldTransform | 2D or 3D, deformable transformation represented as a dense regular grid of vectors. |
| INTERPOLATORS | |
| Nearest Neighbor Interpolation | Intensity of the nearest grid position |
| Linear Interpolation | Assumes linear variation of intensities between grid points, spatially continuous. |
| B-Spline Interpolation | Image intensity using B-spline basis functions |
| Windowed Sinc Interpolation | Based on Fourier analysis consideration. Some windows that can be used: Welch, Cosinus, Hamming, Blackman, Lancos. |

| | |
|-----------------|---|
| Gaussian | Evaluation of Gaussian interpolation of an image. |
|-----------------|---|

METRICS

| | |
|---------------------|---|
| Mean squares | Mean squared pixel-wise difference in intensity between two images. |
|---------------------|---|

| | |
|--------------------|--|
| Correlation | Pixel-wise cross-correlation, which is normalized by the square root of the autocorrelation of the images. |
|--------------------|--|

| | |
|-------------------------------------|--|
| Mutual information by Mattes | Mutual information between images, suitable for multimodal images. Mattes et al implementation |
|-------------------------------------|--|

| | |
|---|---|
| Joint Histogram mutual information | Mutual information between images, suitable for multimodal images. Thevenaz et al implementation. |
|---|---|

| | |
|----------------------|---|
| Demons metric | Metric derivative using image derivatives, default fixed image gradients. |
|----------------------|---|

| | |
|---|--|
| ANTS neighborhood correlation metric | Normalized cross correlation using a small neighborhood for each voxel between two images. |
|---|--|

ITKv3:

Normalized correlation
 Mean reciprocal squared difference
 Mutual information by Viola and Wells
 Kullback Liebler distance metric by Kullback and Liebler
 Normalized mutual information
 Mean squares histogram
 Correlation coefficient histogram
 Cardinality Match metric
 Kappa Statistics metric
 Gradient Difference metric

OPTIMIZERS

| | |
|---------------|-------------------------------|
| Amoeba | Nelder-Meade downhill simplex |
|---------------|-------------------------------|

| | |
|---|--|
| Gradient Descent | Advances parameters in the direction of the gradient where the step size is governed by a learning rate |
| Gradient Descent Line Search | Gradient descent with a golden section line search |
| Conjugate Gradient Descent Line Search | Advances parameters in the direction of the Polak-Ribiere conjugate gradient where a line search is used to find the best value for the learning rate |
| Quasi Newton | Implements a Quasi-Newton optimizer with BFGS Hessian estimation |
| LBFGS | Limited memory Broyden, Fletcher, Goldfarb and Shannon minimization |
| LBFGSB | A modified version of the LBFGS optimizer that allows to specify bounds for the parameters in the search space |
| One Plus One Evolutionary | Strategy that simulates the biological evolution of a set of samples in the search space |
| Regular Step Gradient Descent | Advances parameters in the direction of the gradient where a bipartition scheme is used to compute the step size |
| Powell Optimizer | Powell optimization method. For an N-dimensional parameter space, each iteration minimizes(maximizes) the function in N (initially orthogonal) directions. |
| Exhaustive Optimizer | Fully samples a grid on the parameteric space. This optimizer is equivalent to an exahaustive search in a discrete grid defined over the parametric space. |

D The SimpleITK example

```

import SimpleITK as sitk

#read the images
fixed_image = sitk.ReadImage('training_001_ct.mha', sitk.
    sitkFloat32)
moving_image = sitk.ReadImage('training_001_mr_T1.mha', sitk
    .sitkFloat32)

#initial alignment of the two volumes
transform = sitk.CenteredTransformInitializer(fixed_image,
    moving_image, sitk.Euler3DTransform(), sitk.
    CenteredTransformInitializerFilter.GEOMETRY)
#multi-resolution rigid registration using Mutual
Information
registration_method = sitk.ImageRegistrationMethod()
registration_method.SetMetricAsMattesMutualInformation(
    numberOfHistogramBins=50)
registration_method.SetMetricSamplingStrategy(
    registration_method.RANDOM)
registration_method.SetMetricSamplingPercentage(0.01)
registration_method.SetInterpolator(sitk.sitkLinear)
registration_method.SetOptimizerAsGradientDescent(
    learningRate=1.0, numberOfIterations=100,
    convergenceMinimumValue=1e-6, convergenceWindowSize=10)
registration_method.SetOptimizerScalesFromPhysicalShift()
registration_method.SetShrinkFactorsPerLevel(shrinkFactors =
    [4,2,1])
registration_method.SetSmoothingSigmasPerLevel(
    smoothingSigmas=[2,1,0])
registration_method.
    SmoothingSigmasAreSpecifiedInPhysicalUnitsOn()
registration_method.SetInitialTransform(transform)

registration_method.Execute(fixed_image, moving_image)
sitk.WriteTransform(transform, 'ct2mrT1.tfm')

```