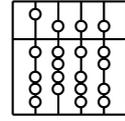


Technische Universität München
Fakultät für Informatik

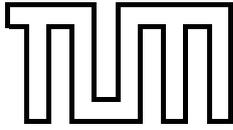


Diplomarbeit

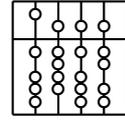
Intensity Based Rigid 2D-3D Registration Algorithms for Radiation Therapy

**In collaboration with Siemens Corporate Research Inc.,
Princeton, USA**

Wolfgang Wein



Technische Universität München
Fakultät für Informatik



Diplomarbeit

Intensity Based Rigid 2D-3D Registration Algorithms for Radiation Therapy

**In collaboration with Siemens Corporate Research Inc.,
Princeton, USA**

Wolfgang Wein

Aufgabensteller: Prof. Gudrun Klinker, Ph.D. (TUM)
Prof. Nassir Navab, Ph.D. (SCR)

Betreuer: Dipl.Inf. Martin Bauer (TUM)
Ali Khamene, Ph.D. (SCR)

Abgabedatum: 15. Dezember 2003

Ich versichere, dass ich diese Diplomarbeit selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 15. Dezember 2003

Wolfgang Wein

Abstract

One of the main applications for computers in medicine is to digitally merge patient data that originates from different sources. For our 2D-3D Registration problem, this means finding the right spatial alignment of a three-dimensional Computed Tomography (CT) data set and one or two X-Ray images from the same patient. An automatic algorithm has to create a digitally reconstructed radiograph (DRR) for a specific orientation in the CT data set, compare the result with the X-Ray image, alter the position and orientation of the DRR and repeat the process until the optimal alignment is reached. A key issue is to assess the alignment of these two images in every step. I present a comprehensive examination of all intensity-based similarity measures known in literature, and use them with a number of different optimization algorithms.

We developed a prototypic application for both manual and automatic registration. The performance is evaluated for different data sets, and validation is done using marker-based ground truth information. The results show that we are able to do very precise and fully automatic registration within a couple of minutes. We propose to use this technique to find the correct patient position on the treatment couch for radiation therapy. This would replace the common, often very inconvenient immobilization methods.

Zusammenfassung

Eine der wichtigsten Anwendungen für Computer in der Medizin ist, Patientendaten von verschiedenen Quellen miteinander zu verschmelzen. Bei 2D-3D Registrierung möchte man die richtige räumliche Ausrichtung für einen dreidimensionalen CT-Datensatz und ein oder zwei Röntgenbilder vom selben Patienten finden. Ein entsprechender Algorithmus muß von einer bestimmten Ausrichtung in den CT-Daten aus ein rekonstruiertes Röntgenbild berechnen, es mit dem realen Röntgenbild vergleichen und entsprechend die Position und Orientierung des rekonstruierten Bildes ändern. Der ganze Vorgang wird wiederholt, bis die richtige Angleichung gefunden wurde. Ein Hauptproblem dabei, ist die Ähnlichkeit der zwei Bilder zu beurteilen. Ich habe eingehend alle bekannten intensitätsbasierten Ähnlichkeitsmaße untersucht, und zusammen mit verschiedenen Optimierungsalgorithmen angewandt.

Wir haben einen Prototypen für sowohl manuelle als auch automatische Registrierung entwickelt. Die Leistung für verschiedene Datensätze wird untersucht, und die Ergebnisse mit marker-basierter "Ground Truth"-Information validiert. Die Ergebnisse zeigen, daß wir in der Lage sind, hochgenaue und vollautomatische Registrierung in wenigen Minuten durchzuführen. Wir schlagen vor, diese Technik für das Finden der richtigen Position des Patienten auf dem Behandlungstisch für Strahlbehandlung zu verwenden. Das würde die üblichen, oft sehr lästigen Fixiermethoden ersetzen.

Preface

About this work This document is my Diplomarbeit for the computer science degree at the Technische Universität München (TUM). A major part of it originates from an internship that I did at Siemens Corporate Research (SCR), Princeton, USA from April through October 2003. I was working on the project *RTReg*, which was initiated by Siemens Oncology Care Systems (OCS) in order to examine the feasibility of automatic 2D-3D Registration for Radiation Therapy.

Acknowledgements First of all, I am very indebted to Ali Khamene for all his helpful advice, his technical knowledge and his patience. He was the driving force behind this project at SCR, and it was a great experience to work with him. Many other employees helped me getting started and were always ready for technical discussions, among them Harry Sundar, Chenyang Xu, Jens Guehring and Sebastian Vogt. Many thanks to the department head Frank Sauer for his uncomplicated way of causing the necessary actions, not to mention his generosity (I was invited for lunch not only once).

Thanks also to Prof. Gudrun Klinker for recommending me to SCR and conducting this thesis with me, and to Prof. Navab for being my counterpart at SCR and for his visionary ideas. Martin Bauer both supported me remotely during the summer and helped me to finish my work back home at university, many thanks to him as well.

Last but not least, all the other international interns at SCR provided a great atmosphere for both exchanging technical issues and fun. Cheers!

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Overview about Registration	1
1.3	Intensity-Based 2D-3D Registration	4
1.4	The Radiation Therapy Application	5
1.5	Outline of this thesis	8
2	Digitally Reconstructed Radiographs	9
2.1	Theoretical Background	9
2.2	Volume Rendering Techniques	10
2.2.1	Ray Casting	10
2.2.2	Shear-Warp Factorization	11
2.2.3	Precomputing a Transgraph	11
2.2.4	GPU Rendering with 2D Textures	12
2.2.5	GPU Rendering with 3D Textures	12
2.3	DRR Rendering	12
3	Intensity-Based Similarity Measures	15
3.1	Measures using only Image Intensities	16
3.1.1	Sum of Intensity Differences	16
3.1.2	Correlation	18
3.2	Measures using Spatial Information	19
3.2.1	Pattern Intensity	19
3.2.2	Gradient Correlation	19
3.2.3	Gradient Difference	20
3.2.4	Sum of Local Normalized Correlation	21
3.3	Histogram Based Measures	22
3.3.1	Entropy of the difference image	23
3.3.2	Mutual Information	23
3.3.3	Correlation Ratio	25
3.3.4	Probability Distribution Estimation	26
3.4	Suggested Class Hierarchy	27
4	Optimization Algorithms	29
4.1	Overview	29
4.2	Best Neighbor Search	31
4.3	Gradient Descent Method	32
4.4	Powell's Direction Set Method	33

4.5	Multiresolution Strategy	34
5	Registration	37
5.1	Nature of the Transformation	37
5.1.1	Euler Angles	37
5.1.2	Unit Quaternion	39
5.1.3	Rotation Axis Representation	40
5.1.4	Parameter Scaling	40
5.2	Use of 2D Transformations	41
5.3	Region of Interest	43
5.3.1	Image Borders	43
5.3.2	Collimator Settings	44
5.3.3	Content Based Mask	44
6	Implementation	47
6.1	User Interface	47
6.2	Our DRR algorithm	48
6.3	The Class SimilarityMeasures	49
6.3.1	Overview	49
6.3.2	Algorithmic Aspects	50
6.4	Speed Issues	52
7	Validation	53
7.1	Camera Calibration	53
7.2	Obtaining Ground Truth	54
7.3	Expressing Alignment Errors	55
7.4	Experiments	57
7.4.1	Computer Simulations with Head Phantom (#1)	57
7.4.2	Experiments with not corresponding phantom data (#2)	60
7.4.3	Experiment with corresponding phantom data (#3 - #6)	61
7.5	Parameter Space Evaluation	65
8	Conclusion	67
8.1	Result	67
8.2	Problems/Discussion	68
8.3	Future Work	68
A	Detailed Experimental Results	71
B	Glossary	83
	Bibliography	85

1 Introduction

1.1 Motivation

The use of computers in medicine is growing rapidly every year. The progress in computer technology, especially in terms of speed and data capacity, enables the development of applications that would not have been possible some years ago. The history of digital image processing is now many decades old. Much younger than this area is processing of image data with three or more dimensions. A full medical Computed Tomography (CT) data set, for instance, can be many hundreds of Megabytes large (say 800 slices of 512 x 512 images with 16 bit pixels equals to 400 MB). Only in recent years it has been possible to work with and display volumetric data of these dimensions interactively. Right now it even becomes feasible to deal with four-dimensional data (any volumetric data created at different times).

Acquiring medical images always imposes specific costs, efforts and sometimes a burden for the patients. This includes two-dimensional imaging like X-Ray or Angiography, and three-dimensional imaging like Computed Tomography (CT), Magnetic Resonance Tomography (MRT) and PET (Positron Emission Tomography). Therefore it has always been desired to get the most out of every patient data set. A very important way of doing so is to combine the information that is available from different data sets. This is an area where computers are used very extensively today. A lot of effort is always put on how to use three-dimensional data sets under any circumstances. Especially CT and MRT scanners have become a common instrument for daily clinical practice. As they acquire volumetric data, they provide a huge amount of valuable information about the patient, making them indispensable these days. This has even increased with the visualization technology that is now available. The scans are often preprocessed and segmented into 3D objects, displayed subsequently to address the full spatial perception of physicians (see figure 1.1).

1.2 Overview about Registration

In order to use patient data that originates from different devices, or has been taken at different times, it is necessary to bring them into spatial alignment. This process is referred to as *Registration*. Maintz and Viergever [25] provide a well-accepted classification of registration problems according to nine criteria, which I would like to summarize shortly.

Dimensionality The first distinction is the number of dimensions that the images to be registered have. CT, MRT and PET are three-dimensional or volumetric modalities, where the information is looked at as either a stack of two-dimensional images (slices) or directly

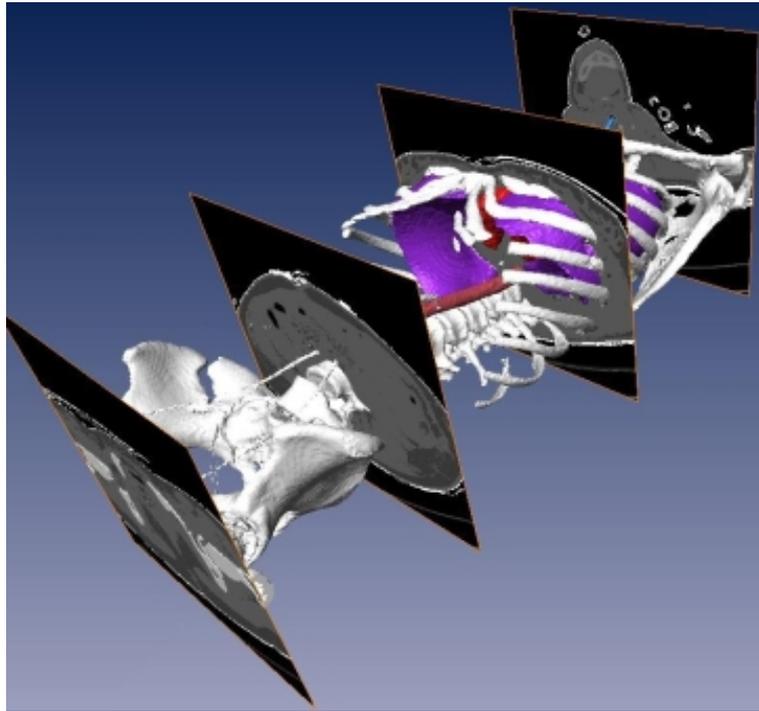


Figure 1.1: Slices from a CT scanner and reconstruction, from [9]

as volumetric elements in space (voxels), arranged in a rectangular grid. On the other hand, X-Ray, fluoroscopy and ultrasound images are two-dimensional.

A classical registration task is the fusion of CT and MR data (a so-called 3D-3D registration). This problem has been solved adequately today, and many semi- or fully-automatic applications are available. Today's computers even enable 3D-3D registration using the full amount of information in the volumes to derive the correct alignment, still in reasonable run-time. A further challenge is to fuse the data after the correct transformation is found. Various recent volume visualization methods can be used. However, transforming one volume in order to perform the found mapping usually imposes some interpolation of it, which may introduce errors or visible artifacts.

Still much more challenging today is 2D-3D registration, denoting the alignment of a volumetric data set with some projective image. A common representative of this class is registration of preoperative CT data with intraoperative X-Ray images. Most applications are in image-guided surgery or patient positioning in radiotherapy. Our problem is of this nature.

Nature of registration basis The information used for achieving spatial alignment can be extrinsic, i.e. foreign objects like fiducial markers are introduced into the image space (and therefore the patient), or intrinsic. In the latter case landmarks can be defined based on the patient's anatomy and used for the alignment. In any of these cases, corresponding points in both images are available, which simplifies the registration very much. Consider-

ing two sets of points in 3D to be registered, the solution is very easy. First, the centers of the two point sets are computed and one set is translated accordingly. Then this point set is rotated iteratively until the sum of the squared distances between the corresponding point pairs is minimal. For this latter step, even a closed solution exists, termed the "Orthogonal Procrustes Problem". At least three points in each set are needed (not placed on a line), but usually more are defined for averaging out errors.

Those errors arise mainly if the defined or detected coordinates of the used points are slightly wrong. The best results can be achieved by screwing markers onto the patient's bones. The high registration accuracy then comes with a high invasiveness, though. On the other hand, a skilled professional can use anatomical features in the images to define points, which is susceptible to errors, too.

Another class of intrinsic registration methods that involves no further objects is based on segmentation. The user selects, points, curves or surfaces in both images which a computer algorithm tries to align. Those methods always require user interaction for selecting those objects. Automatic segmentation of medical data is barely possible today and is subject to intensive research.

Finally, another class of algorithms is in use today that needs neither auxiliary objects nor user interaction, the *voxel property based* or *intensity-based* registration algorithms. They try to find the right alignment by assessing the image similarity based on the raw image data, i.e. the pixel or voxel intensities. This becomes tricky especially when different modalities are involved, as it is not sufficient to just take the sum of the squared intensity differences of the respective pixels/voxels then. Many sophisticated similarity measures have been developed today, which I present in chapter 3. Especially intensity-based 3D-3D registration works very well today, enabling fully automatic fusion of tomographic data. There are interesting other approaches that merge the above aspects. Bansal et al. [33] use a combined segmentation and registration, performed alternatingly.

Nature of transformation If a rigid-body transformation is applied to one of the images, its size and shape remains unchanged, only the position and orientation changes. A rigid transformation in 3D therefore has three translational and three rotational degrees of freedom. See section 5.1 for various parameterizations. If scaling and shearing is added, we obtain an affine transformation with 12 degrees of freedom. At last, many applications want to align images whose internal shapes and structures are changed, resulting in the use of deformable registration. This may end up with an unlimited number of parameters, and many different deformation models can be used to represent the transformation.

Further distinctions are made in the following categories: The type of *Interaction* tells if the registration is conducted interactive, semi-automatic or automatic. The *Optimization Procedure* can either be some dedicated optimization algorithm or a closed-form solution, allowing the parameters to be computed directly (equations exist mainly for point-based registration). Further characteristics are the *Modalities* involved in the registration, and the imaged *Subject* (e.g. a patient, model or atlas) and *Object* (e.g. a specific part of a patient's body).

1.3 Intensity-Based 2D-3D Registration

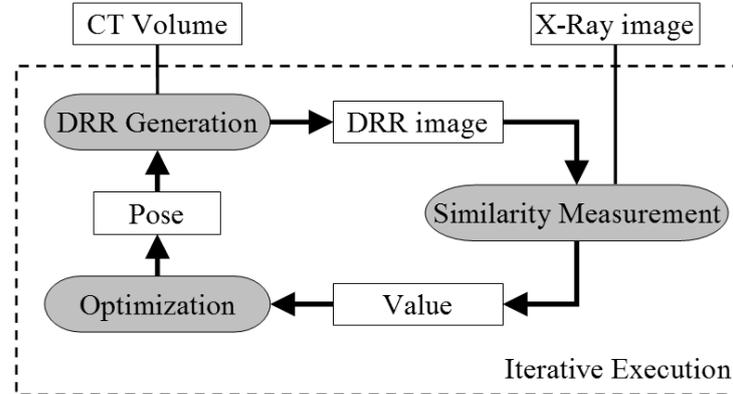


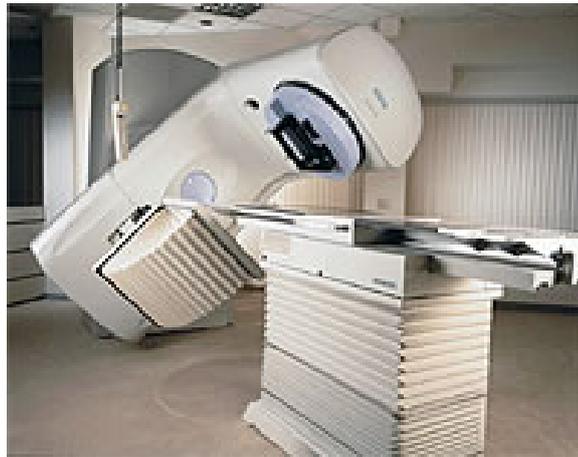
Figure 1.2: Iterative scheme of 2D-3D registration

For this kind of registration, one of the images is a two-dimensional X-Ray image, and therefore a projection of the physical space. Each point in the radiograph corresponds to a line integral of the X-Ray attenuation along the ray from the X-Ray source to the detector. The other image is a three-dimensional CT data set, where each voxel represents the X-Ray attenuation within a certain region in space. Thus X-Ray images can be “simulated” by integrating lines cast through the CT volume, resulting in a *Digitally Reconstructed Radiograph* (DRR). Registration is achieved by finding the right position and orientation with respect to the CT data, so that a DRR computed at this pose matches perfectly with the X-Ray image. This requires knowledge about the geometry of the X-Ray imaging device, so that the DRR can be generated using a correspondent (perspective) projection. The DRR is to be matched with the real X-Ray image by applying a *rigid* transformation. This means that we are looking for a translation in space (expressed by three parameters x , y and z), and a rotation in space (usually expressed with three parameters as well, e.g. θ_x , θ_y , θ_z). There are reasons for using deformable models in order to compensate e.g. for soft tissue movement. Due to the high number of parameters searched for with at the same time relatively sparse information available, this task would be even more challenging. I consider the effect of soft tissue movement negligible for now. This is justified in the majority of cases, when the images comprise enough bony (and therefore rigid) structures. The algorithm works as following (illustrated by figure 1.2):

1. The user feeds the system with a CT data set, a X-Ray image, the intrinsic parameters of the X-Ray imager and an initial starting estimate for the rigid transformation defining where the DRR is to be generated. Additional parameters for the DRR generation may be provided, e.g. in connection with the transfer function.
2. Using the perspective projection extracted from the intrinsic parameters of the X-Ray sensor and the current value of the rigid transformation, a DRR image is computed from the CT data.



(a) Accuray Cyberknife



(b) Siemens Primus

Figure 1.3: Cyberknife Stereotactic Radiosurgery System and Siemens Primus LINAC

3. The alignment of this DRR and the X-Ray image is assessed by any similarity measure. For an intensity-based registration, this measure operates directly on the image data and returns a single number, telling how well the images are matching (see chapter 3).
4. This number is used as cost function value for an optimization algorithm, which alters the rigid transformation in order to achieve better alignment. It tries to find a transformation where the DRR has the maximum similarity with the X-Ray image. This results in a search for a global optimum in a six-dimensional parameter space, as we are dealing with both position and orientation in 3D (see section 5.1 for how to parameterize the transformation). Steps 2-4 are repeated until the optimizer decides based on some abortion criteria, that the best pose has been found.

1.4 The Radiation Therapy Application

Radiation Therapy (more precise: External Beam Radiation Therapy) is the controlled use of high-energy X-Ray radiation for cancer treatment. A linear accelerator (LINAC) delivers a specific dose of radiation to the patient's target tissue. This destroys the cancer cells ability to reproduce themselves, causing the body to get rid of them. Sophisticated treatment planning has to be done in advance in order to make sure that the right area is treated with the right amount of radiation, and that the surrounding tissue is not significantly affected. For the treatment planning, a CT scan is the most important source of information. The radiation oncologist analyzes the CT data set and segments the target area, for instance a tumor. Knowing the exact spatial location of the tumor in the CT coordinates, the patient has to be positioned precisely with respect to the treatment spot of the linear accelerator, so that only the tumor tissue is affected by the full radiation exposure. Various immobilization techniques are used to assure that the patient is and stays in place during the treatment [3]. Usually the linear accelerator is being rotated around one axis during the treatment. This

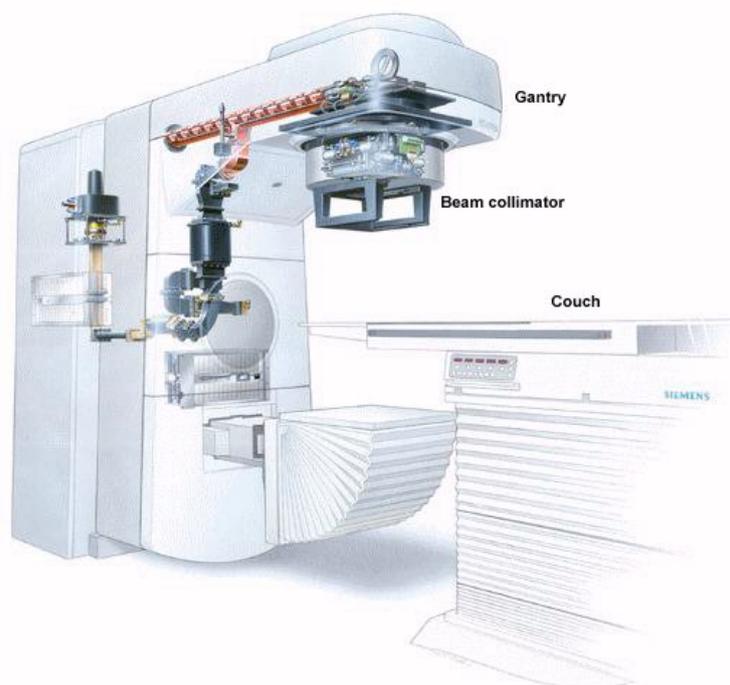


Figure 1.4: Schematic view of the Primus accelerator

makes sure that only the target site in the patient's body always gets the full radiation, and the surrounding tissue is only affected by a non-critical dose. A X-Ray detector normally is integrated in the accelerator arrangement.

The *CyberKnife Stereotactic Radiosurgery System* (Accuray Inc.) is a ultra-compact linear accelerator mounted on a robot with many degrees of freedom (figure 1.3(a)). Using this device, there is even more flexibility to repeatedly change the orientation of the treatment ray, sparing sensible parts of the body from getting any radiation at all. Two separate orthogonal X-Ray source and detector couples are statically attached to the device. Many different registration approaches have been used with this machine, e.g. [38, 37]. To a big extent, fiducial markers implanted into the patient's body are used for a stable registration. David A. LaRose [24] applies markerless 2D-3D registration on a head phantom using the integrated imaging sensors in the Cyberknife. This is the only equivalent approach that I am aware of to evaluate this kind of registration for patient positioning in Radiotherapy.

For our experiments, we could use a *Siemens Primus* linear accelerator at the University of Pennsylvania (UPenn) Hospital in Philadelphia (figure 1.3(b)). The gantry (accelerator source) can be rotated around one axis, in a way that one spot in the treatment room is always in focus of the ray. This is called the Isocenter of the accelerator. Lasers are affixed to all walls and the ceiling in the room, crosscutting in the Isocenter. Therefore skin markers on the patient, aligned with the lasers may be used as a non-invasive way to reposition the patient, in case multiple treatments are desired. Naturally, the precision of the skin markers is very poor, though.

The patient is placed on the treatment bench, which can be translated in all spatial dimensions and one rotational axis, so that the target site corresponds with the Isocenter (figure 1.4). The machine can use its X-Ray source in another low-energy mode in order to create portal images of the area located around the treatment center. Those pre-treatment images can be helpful for the proper alignment of the patient as well.

The patient has to be placed on the bench so that its position and orientation is known with respect to the coordinate system of the linear accelerator. This requires first of all that the patient is attached to the bench in a stable and defined way. Often molded forms previously customized for the specific part of the patient's body are used. Those can be used at the same time for retrieving the relative position of the target area. Another very common tool is the *Stereotactic Frame*, mainly used for neurosurgery. It consists of a light-weight frame which is tightly attached to the patient's skull at four spots (figure 1.5). Reference markers on the frame are visible both in CT and X-Ray images, enabling very accurate tracking of the patient position. Furthermore an additional arc for surgeon's tools can be mounted on the frame (see right picture), which allows retrieving the tool's coordinates by reading the angles from the arc. However some discomfort is caused by wearing the very firmly attached frame from imaging to surgery or treatment, and it may be hampering for certain operations [13]. As the frame has to be attached during the whole procedure, it prohibits the scheduling of multiple treatments, which has been shown to be often beneficial for the curative treatment.

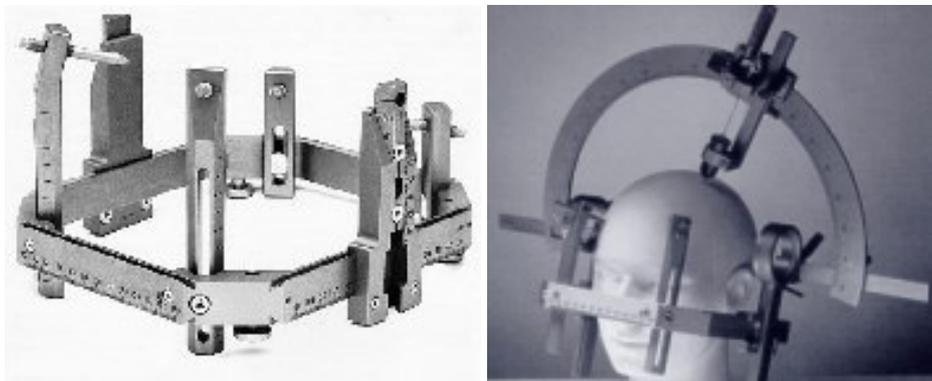


Figure 1.5: Stereotactic Frame

The primary goal of our project *RTReg* is now to achieve correct placement of the patient, solely based on the information from the existing CT data set and one or more portal images done by the linear accelerator. No additional markers or other devices attached to the patient should be used. Our registration should be fully automatic, i.e. the user just provides an initial estimate of the current patient position and launches the computation. No segmentation of both CT or portal data should be required. Therefore we use the raw information provided in the CT and portal images for our registration. This results in a so-called *intensity-based* (often denoted as "voxel-property based", too) 2D-3D registration algorithm.

1.5 Outline of this thesis

In the following three chapters I point out the three main issues for solving our registration problem. First, I cover the basics and existing techniques for creating Digitally Reconstructed Radiographs (chapter 2). Then I theoretically introduce all known intensity-based similarity measures in chapter 3, used for comparing a DRR with a portal image. In chapter 4 I treat different optimization algorithms that are suited for our registration. Eventually, the whole combination of the previously explained techniques in order to obtain a good registration algorithm is described in chapter 5. This includes details like the parameterization of the registration transformation, speedup using 2D transformations and Region of Interests. Some more technical details about the implementation in our RTReg application are listed in 6. Chapter 7 presents the results of our experiments with simulated, phantom and even clinical data. This leads to the conclusion and outlook on further work (8).

2 Digitally Reconstructed Radiographs

We align the CT dataset with one or more portal images by generating an artificial X-Ray image out of the CT voxel data. This image is called *Digitally Reconstructed Radiograph, DRR*. Our goal is that this image should look exactly like the portal image. In order to achieve this, we should first understand how real X-Ray images are generated (section 2.1). The different volume rendering techniques are introduced in section 2.2. Finally I present some details on using volume rendering methods specifically for DRR generation in 2.3.

2.1 Theoretical Background

The intensity value of each pixel of an X-Ray images represents the attenuation of the X-Rays coming from the source to the respective position on the detector. The basic formula is

$$\int_{ray} \mu(\bar{x}) ds = \ln \frac{N_{in}}{N_{out}} \quad (2.1)$$

where $\mu(\bar{x})$ is the linear attenuation coefficient at a given point in space, N_{in} the incident photon count and N_{out} the number of photons hitting the detector. However, this is only an approximation, as the X-Ray source is not monochromatic, but consists out of a characteristic energy spectrum. Therefore we have to use

$$\int_{ray} \int \mu(\bar{x}, E) dE ds = \ln \frac{N_{in}}{N_{out}} \quad (2.2)$$

$\mu(\bar{x}, E)$ being the attenuation for a given point in space and a specific X-Ray photon energy. Under simplifying assumptions you can now interpret the voxel intensities that a CT machine produces as the attenuation at an effective energy, i.e. the monochromatic energy that would give the same attenuation as the energy spectrum that is in fact present. The integer numbers usually saved in CT volumes are either

$$CT = \frac{\mu}{\mu_{water}} \cdot 1000 \quad (2.3)$$

resulting in the range $[0 \dots 4000]$, that can be represented with unsigned 12 bit, or

$$H = \frac{\mu - \mu_{water}}{\mu_{water}} \cdot 1000 \quad (2.4)$$

which provide values between $[-1000 \dots 3000]$ (Hounsfield units). Both formats are usually saved as 16 bit integers.

Coming back to our task of generating a DRR, we encounter the problem that the CT and the X-Ray images are taken at different energies, and therefore the CT voxel intensities can not be used directly for assembling a line integral value. A very simple approach is to scale the original CT voxel intensities with the different effective attenuation of water before summing up the DRR rays.

$$CT' = CT \frac{\mu_{water}}{\mu'_{water}} \quad (2.5)$$

This is only a very simple improvement, though, as energy dependent attenuation changes differently for different kinds of materials. Therefore a better (but more costly) way to correct for different energies is to segment the CT data for instance in metal, bone and tissue, and use different linear scalings, respectively. This involves a manual interaction for segmenting the volume, however.

No matter what method is used for correcting the different energies, the X-Ray image and DRR will always be different. It is important to keep in mind that we are dealing with different modalities. This sets further constraints on our similarity measure, as it should be able to cope with structures that are present in one image, but not the other. Thus a lot of effort is put in selecting apted similarity measures addressing this problem (chapter 3).

2.2 Volume Rendering Techniques

Generation of DRR images is one specific application of volume visualization. Volume rendering techniques can be divided in two main categories: *indirect* and *direct* rendering methods. Indirect methods first perform some preprocessing of the volumetric data in order to obtain reduced structures that are easier to visualize. Very often volume segmentation is conducted, resulting in a three-dimensional shape that can be displayed easily using surface-rendering techniques. Hence most of the computational effort is spent on this segmentation, whereas the rendering is done very efficiently by just drawing polygons.

Direct methods operate directly on the volumetric data and therefore need to process much more information for rendering. As this is very time-consuming, many sophisticated algorithms have been developed to speed up direct volume rendering. The most important of them are briefly described in the following subsections. For DRR generation, only direct methods are apted, as the full voxel information is needed in order to simulate the physical effect of X-Ray attenuation. Every voxel in the volume contributes to the image. The respective voxel intensity values are usually mapped using a *transfer function* in order to get the desired visualization effects. A nice overview to different volume rendering techniques, focusing mainly on recent hardware-accelerated methods, is given in the thesis of Rezk-Salama [35]. Engel and Ertel [6] provide a survey to the latest developments of volume rendering methods on consumer graphics hardware.

2.2.1 Ray Casting

The most straight-forward approach for DRR generation is to directly implement equation 2.1. For each pixel a ray is back-projected into the volume. As the volume is available as

discrete voxel values placed in a regular grid, a sum of all voxels that the ray intersects with is computed. In addition the voxel values are mapped using a specific transfer function in order to achieve a resulting pixel value similar to the X-Ray attenuation along the whole ray. This corresponds to the common Ray Casting technique widely used in volume visualization. The resulting images are of very good quality, as no simplifications are involved. However, a major drawback is the high computation time.

2.2.2 Shear-Warp Factorization

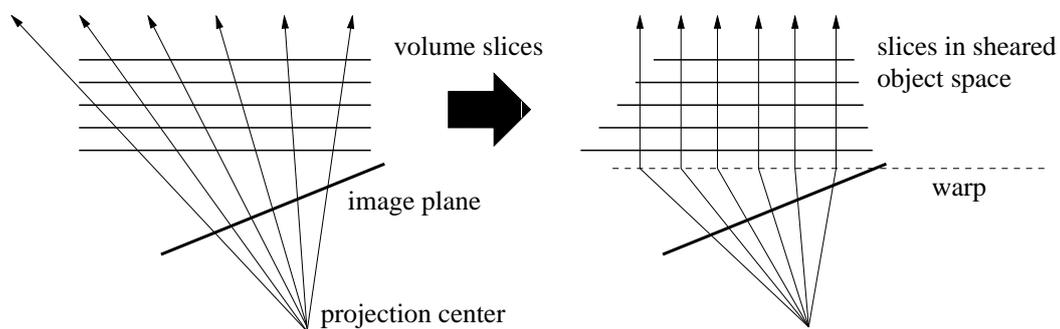


Figure 2.1: Basic operation of the shear-warp algorithm

This algorithm transforms the volume to an intermediate coordinate system called the "sheared object space". In this space all viewing rays are parallel to the third coordinate axis. This involves shearing, scaling and resampling the volume slices. Those slices are composed together in front-to-back order resulting in an intermediate 2D image. The final step is to "warp" this image in order to transform it to the original image space (figure 2.1). See [22, 23] for further details. The rendering is very efficient, as the voxels in the intermediate slices correspond with the scanlines of the final image, and can be composed immediately. On the other hand this algorithm produces artifacts under certain circumstances, and therefore may be problematic if used for accurate registration. However, there are registration applications successfully using this rendering method [47].

2.2.3 Precomputing a Transgraph

This approach uses a data structure which stores gray level values of a big number of DRR rays in an efficient data structure [24]. These ray values are precomputed in a number of different viewing directions around the initial estimate. Rendering of a DRR is therefore reduced to just select the corresponding pixel intensities for the current pose. Precomputing takes about 5 minutes in [21], and generally can be done using any known volume rendering technique. The faster the selected rendering method is and the more time is available, the bigger is the fraction of the parameter space that can be precomputed.

2.2.4 GPU Rendering with 2D Textures

Using more advanced graphics hardware, each volume slice can be loaded into the texture memory of the computer's graphics card. The GPU then rapidly renders the final image by drawing semitransparent planes with the respective textures. As a preprocessing step, additional slices are generated from the volume in different orientations. The set of slices which has the orientation closest to the current point of view is selected for rendering. If the angle between slice orientation and point of view is too high, however, artifacts similar to the results of Shear-Warping will be present.

2.2.5 GPU Rendering with 3D Textures

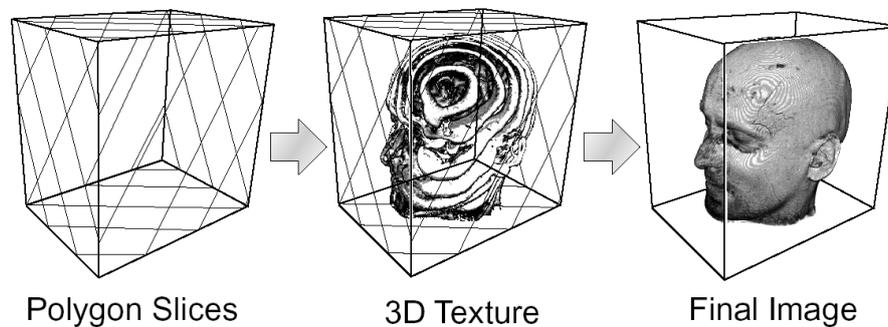


Figure 2.2: Volume Rendering using 3D Textures

Recent consumer graphics cards support an extension called "3D Texture". A whole volume can be stored in the texture memory. The user can provide a set of 3D texture coordinates, causing the GPU to generate a 2D texture interpolated on a plane in the 3D volume. For volume rendering, a set of these 2D planes is being drawn with a specified transparency, equivalent to the 2D Texture algorithm. The difference is, that the 2D planes are constructed on-the-fly out of the volume, and therefore can be oriented perpendicular to the viewing direction at any time, resulting in an image without artifacts (figure 2.2). See [24] for details. We decided to implement this algorithm for NVidia GeForce graphics cards, using the OpenGL library.

2.3 DRR Rendering

To create realistic DRRs, first of all the geometry of the original X-Ray imaging device has to be known. This is always a perspective projection, based on the pinhole camera model. It is therefore important that the used volume rendering algorithm supports the use of a perspective camera. However, many efficient implementations presume that a parallel geometry is sufficient. In order to support perspectiveness, the programming effort can rise a lot, especially if no loss in speed is desired.

Another requirement is that the largest possible precision is used to assemble the DRRs.

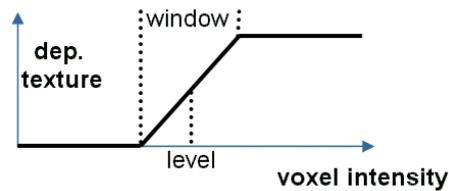


Figure 2.3: Transfer function for DRR rendering

This is not a matter of course, as most interactive volume visualization methods work on 8 bit voxels. However, the underlying CT data for DRR generation is saved with 16 bits per voxel (see section 2.1). In order to obtain the best quality, this precision has to be kept during computation of the reconstruction.

The transfer function of a volume rendering method usually maps the voxel values to specific channel values for red, green, blue and alpha (opacity). For the DRR case only a linear gray value mapping is needed, which has to satisfy equation 2.5. Thus it is fine to use a single ramp in the transfer function and to parameterize it with three values, *window*, *level* and *transparency* (figure 2.3).

We implemented an apted volume rendering algorithm specifically designed for DRR rendering, which fulfilled the above requirements. See section 6.2 for details.

3 Intensity-Based Similarity Measures

As we are searching for the best alignment of two images by transforming one onto the other, it is a very crucial issue to assess how similar two images actually are. The two main classes of similarity measures are **feature-based** and **intensity-based**.

Feature-based measures do some processing with the images first in order to obtain significant information, which can be used to judge the similarity. This can be the position of significant landmarks, or the parametrization of certain shapes within the images, which are obtained by segmentation. However, this processing usually needs some user interaction, which is often not desirable.

On the contrary, intensity-based measures use the full raw image information. A similarity measure is derived using all intensity values in the two images. However, one may consider introducing a region of interest in order to omit non relevant image parts. Working with this kind of measure is often referred to as **voxel property based registration**, too. The main advantage is that registration can be executed right after image acquisition and definition of an initial pose. No further user input is needed for instance for selecting landmarks or setting the parameters for a segmentation. This is the kind of similarity measurement that we are going to use. Figure 3.1 shows an overview about the three classes of measures that will be covered in the following chapters.

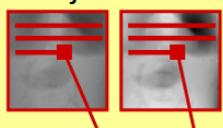
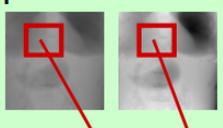
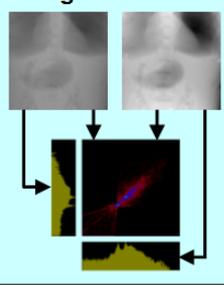
Image Similarity Measures			
Intensity Based			Feature Based
<p>Only Intensities</p>  $\sum_{y=0}^{m-1} \sum_{x=0}^{n-1} f(I_1(x, y), I_2(x, y))$ <ul style="list-style-type: none"> <input type="checkbox"/> Sum of Intensity Differences <input type="checkbox"/> Correlation 	<p>Spatial Information</p>  $\sum_{y=0}^{m-1} \sum_{x=0}^{n-1} f(W_1(x, y), W_2(x, y))$ <ul style="list-style-type: none"> <input type="checkbox"/> Pattern Intensity <input type="checkbox"/> Gradient Correlation <input type="checkbox"/> Gradient Difference <input type="checkbox"/> Local Normalized Correlation 	<p>Histogram Based</p>  <ul style="list-style-type: none"> <input type="checkbox"/> Mutual Information <input type="checkbox"/> Correlation Ratio 	<p>Point Based</p> <p>Shape Based</p> <p>...</p>

Figure 3.1: Classification of image similarity measures

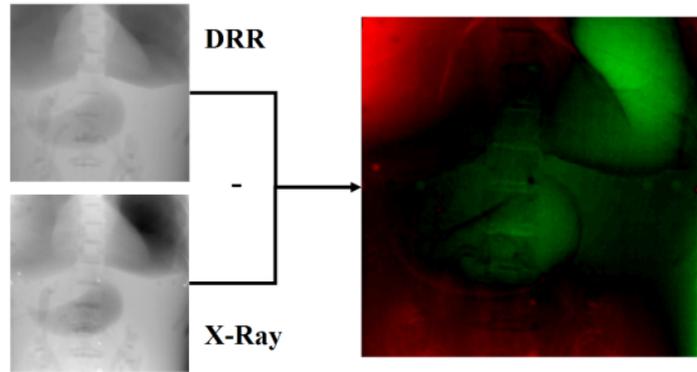


Figure 3.2: Difference image from a DRR and X-Ray image, green=positive, red=negative

3.1 Measures using only Image Intensities

This class of similarity measures compares the intensity values of both images pair-wise at the same pixel positions. Subsequently one single value is composed out of it with a certain scheme. An advantage of this kind of measure is that it can be used not only with twodimensional images, but with any kind of data in arbitrary dimensions, as no spatial information is considered.

3.1.1 Sum of Intensity Differences

One very simple similarity measure is the sum of squared differences (SSD) between two images:

$$SSD = \frac{1}{N} \sum_{(x,y \in T)} (I_1(x,y) - I_2(x,y))^2 \quad (3.1)$$

T is the overlap domain (eventually combined with a region of interest) of the images, N is the number of pixels in T . It can be shown that SSD is the optimum measure when two images only differ by Gaussian noise. Therefore this measure is mostly used in registration applications where two images from the same modality are used, otherwise this constraint is not realistic. Besides simplicity, another reason to use this measure may be that specific optimization algorithms can be used, that can minimize sum-of-squares expressions very efficiently (see section 4.1). Due to its simplicity, this measure is especially used for intra-modality registration [13].

A problem with SSD is that a small number of pixels with very large intensity differences can alter the similarity value very much. This may be the case when certain shapes are present in one image, but not in the other. This effect can be reduced by using the sum of absolute difference, SAD instead of SSD [13]:

$$SAD = \frac{1}{N} \sum_{(x,y \in T)} |I_1(x,y) - I_2(x,y)| \quad (3.2)$$

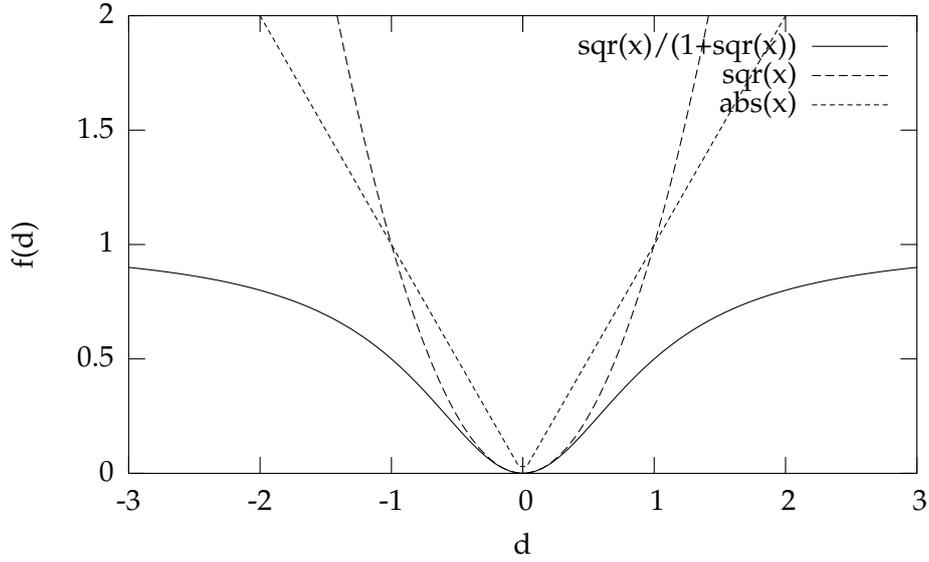


Figure 3.3: Different weighting functions for summing up differences

Generalizing, one can rewrite this kind of similarity measure like

$$\frac{1}{N} \sum_{(x,y \in T)} f(d)$$

, where $d = I_1(x, y) - I_2(x, y)$. By choosing the function $f(d)$ appropriately, one can adopt the similarity measure to behave in different ways.

I suggest to consider a $f(d) = \sigma^2 / (\sigma^2 + d^2)$ form of the addend. This would normalize the measure to the interval $[0 \dots 1]$ and limit the influence of occasional huge differences between the intensities:

$$1 - \sum_{(x,y \in T)} \frac{\sigma^2}{\sigma^2 + (I_1(x, y) - I_2(x, y))^2} = \sum_{(x,y \in T)} \frac{(I_1(x, y) - I_2(x, y))^2}{\sigma^2 + (I_1(x, y) - I_2(x, y))^2} \quad (3.3)$$

The resulting measure is the most sensible to changes when the gradient of $f(d)$ is maximal. Therefore

$$f''(d) = 0 \Rightarrow d = \pm \frac{\sigma}{\sqrt{3}}$$

Generalizing, the function $f(d)$ may be defined anyhow, as long as it satisfies the constraints for a *robust estimator*. Robust estimators [16] are widely used e.g. in computer vision and machine learning. They are basically a tool for statistical problems, where the underlying assumptions are inexact, and do especially have outliers. Robust functions thus have to be insensitive to those outliers, just as e.g. equation 3.3 is.

3.1.2 Correlation

Linear correlation is used everywhere in mathematical and computer science problems, when the relation between two data sets have to be assessed in a fast and simple way. In our case, we want to quantize the linear dependency of the pixel values in one image to the other. This measure is called *Normalized Cross Correlation* (NCC), often referred to as *Correlation Coefficient* as well:

$$\begin{aligned} NCC &= \frac{1}{\sigma_1 \sigma_2} \frac{1}{n} \sum_{(x,y) \in T} (I_1(x,y) - \bar{I}_1) \cdot (I_2(x,y) - \bar{I}_2) \\ &= \frac{\sum_{(x,y) \in T} (I_1(x,y) - \bar{I}_1) \cdot (I_2(x,y) - \bar{I}_2)}{\sqrt{\sum_{(x,y) \in T} (I_1(x,y) - \bar{I}_1)^2} \cdot \sqrt{\sum_{(x,y) \in T} (I_2(x,y) - \bar{I}_2)^2}} \end{aligned} \quad (3.4)$$

\bar{I}_1 and \bar{I}_2 are the mean intensity values in the images I_1 and I_2 , respectively. The denominator is the product of the standard deviations in the two images.

If the one image is linearly dependent on the other one, e.g. $\forall (x,y) \in T : I_2(x,y) = aI_1(x,y) + b$, a value of one would be achieved (and -1 accordingly, if a is negative). That means for practical use, that both different contrast and brightness values in the images should not affect the similarity measure (except for differences due to rounded intensity values), which is very desirable.

If the measure is computed according to equation 3.4, the image data has to be read twice, once for calculating the mean intensities, and once for completing the summation. This can be avoided by expanding the formula:

$$NCC = \frac{\sum I_1(x,y)I_2(x,y) - n\bar{I}_1\bar{I}_2}{\sqrt{\sum I_1(x,y)^2 - n\bar{I}_1^2} \cdot \sqrt{\sum I_2(x,y)^2 - n\bar{I}_2^2}} \quad (3.5)$$

The result includes more arithmetic operations, however the images have to be read only once. Besides, as usually only one of the images is changing during registration (let's assume I_1), some of the sums have to be computed only once, \bar{I}_2 and $\sum I_2(x,y)^2$. On the other hand, this form of cross correlation is more subject to numerical problems (see Numerical Recipes [32], section 14.1).

If one normalizes the constant image to have a mean value of zero and variance of one (which is a linear mapping and thus does not change the measure value), the computation becomes very simple:

$$\begin{aligned} \tilde{I}_2(x,y) &= (I_2(x,y) - \bar{I}_2)/\sigma_2 \\ NCC &= \frac{1}{\sigma_1 n} \sum I_1(x,y)I_2(x,y) \end{aligned} \quad (3.6)$$

Depending on the magnitude of the image data and the time constraints, one may choose one of the presented formulas. For our registration problem, the correlation is computed on two-dimensional images, resulting in almost no loss of time at all. Hence we do not have to consider any speed issues and use the first equation 3.4.

One problem of this measure could be that if parts of the image are directly and inversely related to each other, they may be compensated in the measure. This may be especially

significant if different modalities are used for the registration, as different tissue types may have different intensity ranges then, and there would be a change that some of them are related in the opposite way. If this shows to be a relevant problem, one can modify the measure in order to sum up over the absolute values of the linear correlation:

$$NCC' = \frac{1}{\sigma_1 \sigma_2} \sum_{(x,y \in T)} |(I_1(x,y) - \bar{I}_1) \cdot (I_2(x,y) - \bar{I}_2)|$$

Normalized Cross Correlation has been used for various registration problems. In 3D-3D registration it is mainly limited to images of the same modality, however there are approaches using NCC on image data transformed to fourier-space [13]. It is being used and evaluated for 2D-3D registration [30, 15, 34] and slice-to-volume registration [8] as well.

3.2 Measures using Spatial Information

This class of measures still evaluates the data pixelwise, but in addition some kind of neighborhood information is involved at every position. This can be done by adding all differences within a certain radius (Pattern Intensity), or by calculating gradient images for further examination.

3.2.1 Pattern Intensity

This measure [30, 15] directly assesses the content of a difference image, $I_{diff} = I_1 - sI_2$. If the images are perfectly aligned, the difference image should contain the fewest amount of patterns, or be constant in the optimum case. Pattern Intensity now sums up over differences between neighbored pixels in I_{diff} .

$$PI_{r,\sigma}(s) = \frac{1}{|T|} \sum_{x,y \in T} \sum_{v,w} \frac{\sigma^2}{\sigma^2 + (I_{diff}(x,y) - I_{diff}(v,w))^2} \quad (3.7)$$

where $(v,w) \in T$ and $(x-v)^2 + (y-w)^2 < r^2$. Good working parameters are $r = 3$ and $\sigma = 10$ [30], the latter selecting the values in the difference image where the measure is the most sensitive (see section 3.1.1). The scaling factor s for creation of the difference image should be chosen so that the difference image has the least contrast. Note that a constant shift between the image intensities does not affect the similarity measure, as it only assesses differences in the difference image.

3.2.2 Gradient Correlation

By using horizontal and vertical Sobel templates, four gradient images dI_1/dx , dI_1/dy , dI_2/dx and dI_2/dy are created (figure 3.4). Then Normalized Cross Correlation (equation 3.4) is calculated of these horizontal and vertical gradient images, respectively. The final value of the measure is the average of the two NCCs [30].



Figure 3.4: Horizontal and vertical gradient images of a DRR image

If we now assume a case where this measure reaches the maximal value (namely 1 or -1), the gradient images have to be linearly dependent:

$$\begin{aligned}\frac{\partial I_2(x, y)}{\partial x} &= a_h \frac{\partial I_1(x, y)}{\partial x} + b_h \\ \frac{\partial I_2(x, y)}{\partial y} &= a_v \frac{\partial I_1(x, y)}{\partial y} + b_v\end{aligned}$$

a_h, b_h and a_v, b_v are constants for the horizontal and vertical gradient images, respectively. Integrating gives the following constraints:

$$\begin{aligned}I_2(x, y) &= a_h I_1(x, y) + b_h x + c_h + f_h(y) \\ I_2(x, y) &= a_v I_1(x, y) + b_v y + c_v + f_v(x)\end{aligned}$$

Here $f_h(y)$ and $f_v(x)$ are arbitrary functions. As can easily be seen, the postulated dependency between the images is now much looser compared to the original Cross Correlation.

3.2.3 Gradient Difference

This measure, proposed in [30] evaluates two difference images I_{diffH} and I_{diffV} which are calculated from gradient images (section 3.2.2). The same $1/(1 + x^2)$ structure as pattern intensity (section 3.2.1) is applied.

$$GD(s) = \sum_{x,y \in T} \frac{A_h}{A_h + (I_{diffH}(x, y))^2} + \sum_{x,y \in T} \frac{A_v}{A_v + (I_{diffV}(x, y))^2} \quad (3.8)$$

$$I_{diffH}(x, y) = \frac{dI_1}{dx} - s \frac{dI_2}{dx}, I_{diffV}(x, y) = \frac{dI_1}{dy} - s \frac{dI_2}{dy}$$

A_h and A_v are constants which work well if they are set to the variance of the respective reference image [30].

3.2.4 Sum of Local Normalized Correlation

This measure has been introduced in [24] as an extension of the Normalized Cross Correlation measure. If the intensities in the one image are distorted in terms of slight global changes, the constraint of a linear relationship between the intensities might not hold anymore. A solution is to calculate the correlation values of many small image parts and to take the average of them.

$$SLNC = \frac{1}{|Q|} \sum_{p \in Q} NCC(I_1, I_2, P(p)) \quad (3.9)$$

Q is a set of points in the image, and $P(p)$ defines a neighborhood around a specific point p , in which the correlation is calculated. A square of size 5 by 5 to 12 by 12 pixels makes probably sense. The set Q may for instance be defined as a grid of pixels, so that the squares are adjacent. Another option is to take more pixels into account, so that the squares are overlapping. On the extremum every pixel is considered, and the correlation around its neighborhood calculated. This is computationally costly, but an efficient implementation with recursive filters is possible.

A problem arises if some of the neighborhood areas are constant, as the NCC value is undefined then. For the fixed image it is fine to omit the values for this specific regions, as they are the same during the whole registration process. If the moving image has constant regions, however, the measure might get unstable with this approach, as certain region windows would sometimes be considered, sometimes not. In [24] a small amount of Gaussian noise is added to the DRR to solve this problem.

In my opinion, though, it is sufficient to omit those image regions, even if their number is not constant. To support this, let's consider a workaround that increases one arbitrary image pixel in the respective constant image region. Let x be the set of pixels considered in the moving image, y the pixels in the fixed image, k is the position where the respective x image value is increased by 1. The normalized correlation for this region can then be developed:

$$\begin{aligned} & \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2} \sqrt{\sum (y_i - \bar{y})^2}} = \\ & \frac{y_k - \bar{y}}{\sqrt{\sum (y_i - \bar{y})^2}} = \\ & \frac{y_k - \bar{y}}{\sqrt{n-1} \sigma_x} \end{aligned}$$

The expectation of this correlation is zero, as $E(y_k - \bar{y}) = 0$. Therefore it is justified to omit those undefined correlation windows.

Another improvement can be made by not just taking the average of the region correlations, but weighting them with the respective variance of the region in one of the images [24]. This makes sure that for instance the correlation of windows containing bony structures are better emphasized than low contrast windows containing e.g. tissue. The measure is then called *Variance-Weighted Sum of Local Normalized Correlation*. Furthermore this measure has the effect that structures that are only present in one image, can be efficiently ignored. By using weighting with the variance of the first image, for instance, the high variance at certain structures in the second image (which are considered to be disturbing) are being ignored.

Therefore this measure is especially apted for applications, where objects are present in one of the images that should be ignored in the registration.

3.3 Histogram Based Measures

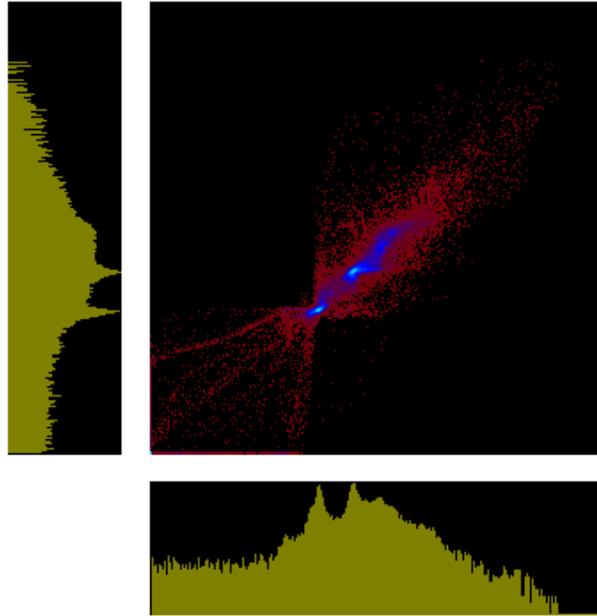


Figure 3.5: Joint and individual histograms for a DRR and a X-Ray image

Those measures overlap with what is often termed *Information Theoretic Measures*. Here, successful registration means to maximize the amount of shared information in the two images, or two minimize the amount of information present in a difference image. In order to automate this idea, some measure of information is needed as prerequisite. The most commonly used one is the Shannon entropy

$$H = - \sum_a p(a) \cdot \log p(a) \quad (3.10)$$

, where $p(a)$ denotes a discrete probability distribution. Another entropy concept apted for registration has been introduced in [46], the Cumulative Residual Entropy. If the amount of information in an image is to be assessed, the image pixel intensity is treated as random variable. $p(a)$ is then the probability that any pixel in the image has the intensity a . It is being summed up over all possible intensities, e.g. $[0 \dots 255]$ for 8 bit images. Therefore the distribution can be composed by creating a histogram (counting the occurrences of every possible intensity value) and dividing by the number of pixels in the image. Similar to that, a joint probability distribution is created by counting the occurrences of every pair of intensities in two images (at the same pixel position, respectively), resulting in a two-dimensional histogram. Figure 3.5 shows a joint histogram and the respective two individual histograms. Once those distributions are computed, every further processing is

done on them, without considering any of the original image data.

To mention is, that the measures from section 3.1 can be rewritten in order to be computed based on histogram information, too. The Sum of Squared Differences and Normalized Cross Correlation can be computed as follows:

$$\begin{aligned}
 SSD &= \sum_{a,b} p(a,b)(a-b)^2 \\
 NCC &= \frac{\sum_{a,b} p(a,b)(a-m_a)(b-m_b)}{\sqrt{\sum_a p(a)(a-m_a)^2} \sqrt{\sum_b p(b)(b-m_b)^2}} \\
 m_a &= \sum_a a p(a), \quad m_b = \sum_b b p(b)
 \end{aligned}$$

However, this is not the original idea of those measures, and the respective calculation is more costly. It is only important to bear in mind that the information base is in fact the same, namely image intensity information.

3.3.1 Entropy of the difference image

In this case the entropy of a difference image $I_{diff} = I_1 - sI_2$ like in 3.2.1 is examined [30]. If the two images are matching perfectly, the difference image should be empty (i.e. have constant intensity values), which results in an entropy of zero.

3.3.2 Mutual Information

For measuring the amount of combined information in two images, the joint entropy is used:

$$H(I_1, I_2) = \sum_{a,b} p(a,b) \cdot \log p(a,b)$$

it is being summed up over the intensity range of both images. The intensity probabilities $p(a,b)$, also called probability distribution function (PDF) can be visualized as a two-dimensional joint histogram. Every value of $p(a,b)$ tells the occurrence of the intensity a in the first and intensity b in the second image at the same positions, respectively. If I_1 and I_2 are totally unrelated, the joint entropy will be the sum of the entropies of the individual images. The more similar the images are, the lower the joint entropy is (if the images are the same, it equals the entropy of the images, $H(I_1, I_2) = H(I_1) = H(I_2)$). Mutual Information now combines the entropy calculations of the individual images and the combination:

$$MI = H(I_1) + H(I_2) - H(I_1, I_2) = \sum_{a,b} p(a,b) \cdot \log \frac{p(a,b)}{p_1(a)p_2(b)} \quad (3.11)$$

The range of this measure is now $[0, H_{max}]$, where H_{max} is the maximum entropy that is possible. It can be achieved by assuming equal distribution of all intensity values in the image, i.e. a constant histogram:

$$H_{max} = - \sum_a \frac{1}{n} \cdot \log \frac{1}{n} = \log n$$

n is the number of histogram entries, e.g. 256. It is desirable to normalize the mutual information measure to have a value between 0 and 1. This can be done easily [13]:

$$MI' = \frac{2MI}{H(I_1) + H(I_2)} = 2 - \frac{2H(I_1, I_2)}{H(I_1) + H(I_2)} \quad (3.12)$$

An alternative normalization scheme is to use the ratio between the individual entropies and the joint entropy. This has been shown to be more invariant to changes in the overlapping area of the images [40, 39], but nevertheless is only a rewriting of the first normalized equation:

$$MI'' = \frac{H(I_1) + H(I_2)}{H(I_1, I_2)} = \frac{1}{MI' - 2} \quad (3.13)$$

Some important features of normalized mutual information (equation 3.12):

- The result is zero if one or both of the images are constant.
- The result is one if both images are identical or the pixel values in one image are only scaled and shifted with respect to the pixel values in the other image. However, this is only valid if no rounding errors occur and no intensities get lost on the bounds. This implicates that the entropies of both images are still the same.
- Varying the size of the histograms and the joint probability distribution has an important affect on the MI value. I calculated normalized MI for two 256x256 16 bit images, which consist to one part of a constant ratio of common random noise, and to the other part of a certain percentage (which is displayed on the axis) of independent random noise. This calculation is done for all histogram sizes from 4 bit to 12 bit. The results can be seen in figure 3.6. It seems to be desirable to adapt the histogram size to different image resolution and registration accuracy. If we deal with coarse images, using high histogram resolution could end up in wrong local maxima, as pairs of exact intensities in the images are not relevant then. On the other hand, if high-resolution images are used and the registration estimation is already very close to the optimum, full histogram resolution results in a very sharp rise to a value of one.

A very important property of Mutual Information is, that it assumes no functional dependency between the images at all, only statistical dependence between the intensities. Due to this fact this measure is very popular in registration problems where different modalities are involved. A very comprehensive work about Mutual Information and its information-theoretic background can be found in [45]. The proposed stochastic maximization procedure and experimental results are also summarized in [44]. Mutual Information has become the most popular similarity measure for inter-modality registration problems, a nice survey is available by Plum, Maintz and Viergever [31].

One drawback of Mutual Information that is criticized sometimes, is that it does not consider any spatial information. In recent years some ideas have come up that would compute Mutual Information not just on intensity values, but on vectors incorporating intensity and one to three spatial parameters, for instance the pixel location in the images. The consequence is that the entropies of multi-dimensional probability distributions have to be computed. Those distributions, and especially the respective joint probability distribution,

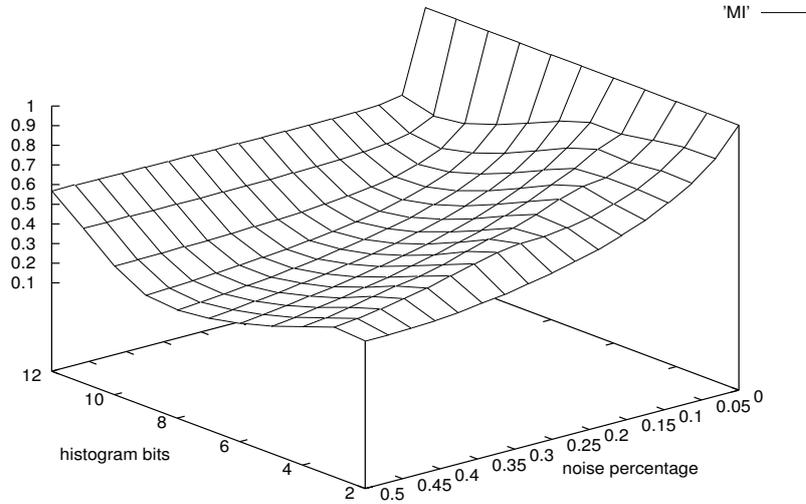


Figure 3.6: Normalized MI on images with different noise levels and histogram sizes

usually cannot be saved in an array anymore. Thus the problem is generalized to estimating the necessary entropies (overview in [2]), discarding any explicit form of the underlying probability distributions. For instance, some immediate storage of the “probability vectors” (incorporating discrete intensity information) in lists can be done, and the entropy calculation is carried out with some minimal spanning tree algorithm on those vectors [14]. Other strategies to estimate multi-dimensional entropies can be found in [27]. Theoretically, those multi-dimensional MI approaches are very promising, but are up to now doomed to fail due to the enormous computation effort.

3.3.3 Correlation Ratio

This is an alternative approach of an information theory based similarity measure, introduced by Roche et al. [18]. It can be seen as a measure of how well one image explains the other. Therefore the image intensities are treated as random variables and a functional dependency between the images is assumed. A very important characteristic of correlation ratio is, that it is not symmetric. One has to decide in advance which image should be the model, and therefore is used as a base for the estimation of the second image. The basic equation is:

$$\eta(I_2|I_1) = \frac{\text{Var}(E(I_2|I_1))}{\text{Var}(I_2)} \quad (3.14)$$

In this case the variables I_1 and I_2 stand for random variables of the image intensities, respectively. Assuming that I_2 is totally independent of I_1 , the expectation $E(I_2|I_1) = E(I_2)$ is constant and thus its variance is zero. On the other hand, assuming full functional dependence, every value of I_2 can be predicted given knowledge of I_1 and therefore $E(I_2|I_1) = I_2$,

resulting in a similarity measure value of 1. Computing the actual value is done with

$$\begin{aligned}\eta(I_2|I_1) &= 1 - \frac{E_a [Var(I_2|I_1 = a)]}{Var(I_2)} = 1 - \frac{1}{\sigma^2} \sum_a \sigma_a^2 p_1(a) \\ \sigma^2 &= \left(\sum_b b^2 p_2(b) \right) - \left(\sum_b b p_2(b) \right)^2 \\ \sigma_a^2 &= \left(\frac{1}{p_1(a)} \sum_b b^2 p(a, b) \right) - \left(\frac{1}{p_1(a)} \sum_b b p(a, b) \right)^2\end{aligned}$$

Correlation Ratio originates as an extension of Correlation Coefficient (termed Normalized Cross Correlation in my work, section 3.1.2), now assessing functional dependence instead of just linear one. When using this measure with images of the same modality, the behavior of Correlation Ratio and Correlation Coefficient is indeed very similar, as I was able to confirm in corresponding similarity measure plots (chapter 7).

In contrast to Mutual Information, the Correlation Ratio is not based on entropies, but on the variance of a image random variable that is conditioned with the random variable of another image. The main advantage of this is that proximity information in the intensity space is considered, whereby the entropy and joint entropy calculations only assess pairs of intensity ranges. The authors claim [18] that this primarily results in better robustness at lower image resolutions.

3.3.4 Probability Distribution Estimation

The probability distributions used for MI and CR can be approximated using only a specific number of random sample pixels from the images. This is especially beneficial for registration applications, as a DRR computation can be accelerated notably by just computing specific pixels in the image.

The *Parzen Window* method approximates a probability distribution using a fixed number of samples. The assumption is that each sample point a_i contributes to the overall distribution with a Gaussian density distribution centered at the sample value [43]:

$$p(a) \approx \frac{1}{n} \sum_{i=1}^n G(a - a_i) \quad (3.15)$$

, n being the number of available samples and $G(x)$ an apted Gaussian distribution centered at the origin. Using a two-dimensional Gaussian kernel allows to use the same scheme for the joint probability distribution. Viola and Wells developed equations using this estimation for calculating image entropies, Mutual Information and its derivatives efficiently [44, 45].

Sparse Sampling is a similar approach, assembling the necessary histograms by inserting the intensity values into the histogram, however using only some random samples and a big histogram bin size. Zöllei et al [48] build the histograms from 100-150 samples using 32 intensity bins.

Any similarity measure based on these estimations will be noisy due to the use of random samples. This can be taken care of by using stochastic optimization algorithms like stochastic gradient descent, overcoming at the same time problems with local optima (see chapter 4).

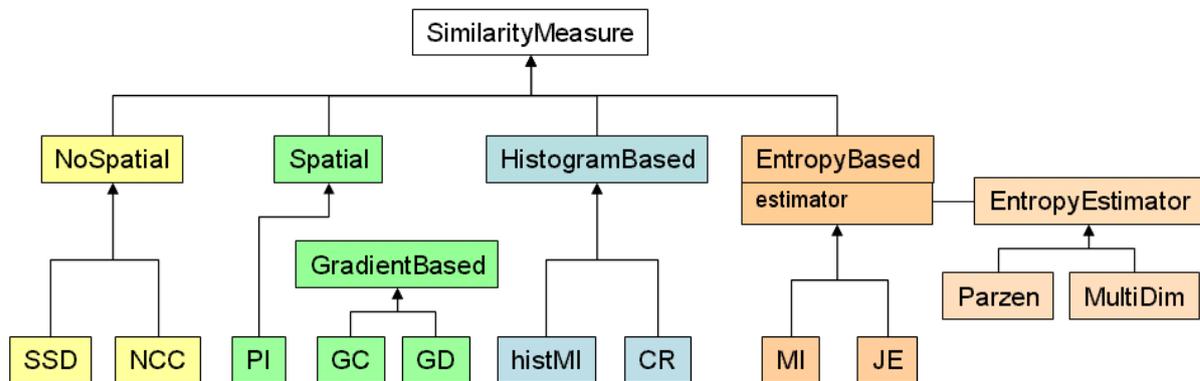


Figure 3.7: Class hierarchy proposal for similarity measures

3.4 Suggested Class Hierarchy

For use in our specific application, I implemented all introduced similarity measures in one single class (see section 6.3), in order to enable rapid computation of multiple measures at the same time. Besides, there were plans to integrate the measures into a library currently under development by Siemens Corporate Research, the Fusion Toolkit (FTK). As being part of a library, the measures should be as generic as possible, so that for instance the data type being used, or even the dimensionality, may be chosen by the user. I developed a class hierarchy that meets those requirements (figure 3.7). The top class should provide the interface to set the images to be compared and some general options. The various subclasses specify details for each kind of measures.

By now, the image class used in the FTK library is templated both with the data type and the number of dimensions. This allows to implement all measures only once, whatever dimensionality the images have (2- 3- or 4-dimensional). The iterator concept is used extensively, therefore allowing the measures to scan the image data without any case distinctions. For the subcategory of spatial measures, some customized neighborhood iterators are necessary. For Pattern Intensity a circular (or spherical) one, and for the gradient based measures one who can compose gradient information in arbitrary dimensions. The subclass for histogram based measures constructs individual and joint histograms, needed by Mutual Information and Correlation Ratio. The additional subclass for entropy based measures provides a scheme of attaching any desired entropy estimator. This might be Sparse Histogramming, Parzen Windowing or even multi-dimensional approaches.

Furthermore, the FTK image class will provide a mechanism to attach transformation and interpolation class instances for transparently returning transformed pixels using the same access method. Therefore any kind of registration can be implemented easily using those classes.

4 Optimization Algorithms

In this chapter I will present some optimization algorithms that are apted to solve the problem of finding the best rigid transformation for aligning a projected image with a reference image. This optimization problem thus has six degrees of freedom. Beforehand, I provide an overview about non-linear optimization in general.

4.1 Overview

Every optimization algorithm's goal is to find the parameter vector \bar{x} that minimizes (or maximizes) the value of a cost function $F(\bar{x})$. Therefore the algorithm searches the parameter space with a specific scheme iteratively. It terminates once some abortion criteria has been satisfied, for instance if the change in the cost function value is below a limit (functional tolerance) or the search distances in the parameter space drop below a given value (parameter tolerance). Optimization schemes have always been developed with the aim of getting along with as few cost function evaluations as possible, therefore taking rare samples of the cost function value and using a nifty strategy to decide where to set the next try in the parameter space. This is especially crucial in our application, as the cost function is a similarity measure which assesses the difference between a DRR and the X-Ray image. Calculating this measure comprises generating a new DRR, and though is very computationally expensive. Besides, this similarity between two images is naturally highly non-linear, and we can not make use of any parameter constraints that ease the registration problem (unless we use the approximations introduced in section 5.2). Therefore the class of optimization problem that we have to solve is *unconstrained non-linear optimization*.

A main difference between optimization algorithms is if they make use of the cost function's derivatives. The gradient vector can be used as base for determining the next steps, as done in the *Gradient Descent* algorithm or *Conjugate Gradient* methods. There are applications where the cost function gradient can be calculated much faster than the function value itself, boosting the optimization time accordingly. Some registration approaches use derivative approximations, especially when Mutual Information is used as similarity measure [48, 44, 45, 49]. These methods are based on estimating the underlying image entropies by using just some samples of the image pixels. As our DRR generation is 3D-texture-based and instantly renders the whole image, I do not consider those approaches. For the Gradient Descent optimizer, I compute the gradient vector by fully evaluating the cost function at displaced positions in each degree of freedom, which limits the performance of this algorithm. Another important set of methods can be used if the cost function is a sum of squares of nonlinear functions

$$F(\bar{x}) = \frac{1}{2} \sum_{i=1}^m f_i(\bar{x})^2 = \frac{1}{2} \|f(x)\|_2^2 \quad (4.1)$$

This type of function often occurs if a model is to be fitted to data, where \bar{x} is a parameter vector for the model function and the individual $f_i(\bar{x})$ express the distance from a specific part of the model to a data element. Both the gradient vector and the Hessian matrix of 4.1 have a special structure, allowing adapted optimization algorithms like *Gauss-Newton* and *Levenberg-Marquardt* to be used [11]. Out of the similarity measures that we use, only Sum of Squared Differences (SSD) has this scheme (equation 3.1), at the same time being the most simple measure. Hence we do not consider using special algorithms for sums of squares cost functions. However there is existing work using those methods [19].

Especially in engineering problems, where the objective function sometimes can be tremendously difficult to compute, optimization schemes are used that approximate the cost function, the so-called *Surrogate Methods* [41]. This is usually done using a specific interpolation method to simulate the cost function at desired positions. A lot of effort can be spend there, as the more knowledge about the underlying problem is used, the better the approximated values will be. [5] uses a surrogate based method for registration of CT with C-Arm fluoroscopy images.

For many optimization problems a danger exists of getting trapped in local optima of the cost function. One possible solution is to repeatedly start the optimization from different starting points in the parameter space, eventually choosing the best result as optimum. Another workaround comprises adding noise to the cost function value, either on purpose, or as byproduct of an accelerated cost function estimation [45]. This results in a stochastic optimization approach. Another widely used technique to overcome local optimas are Simulated Annealing methods, which sometimes take steps in bad directions with decreasing probability. That "bouncing around" in the parameter space reflects the movement of molecules when matter is in the transition from fluid to solid state. This method was not applied much to registration problems, though.

One problematic aspect with optimization for registration purposes is, that sometimes the correct alignment is found at a local optimum, and the global cost function optimum denotes some wrong registration. For instance, Mutual Information can reach a maximum if the structures in the two images do not overlap at all. In our application the starting pose is always close enough to the correct registration to avoid problems with broadly placed local optima. But on the other hand it is possible that the image similarity has different optimas scattered in a small area around the correct registration pose. This effect would be supported e.g. by interpolation artifacts, considering small changes in the DRR pose. Fortunately it turned out that both our DRR generation and some appropriate similarity measures work in a way that the resulting cost function is very smooth (see diagrams in section xx). Hence we do not have to address the local optima problems in our specific application.

Optimization comprises that a cost function is either to be *maximized* or *minimized*. For registration tasks, the cost function is the similarity of the two images to be registered. As most of the considered similarity measures increase in proportion to the image alignment, we want to maximize them. Therefore I will stick to the option of maximizing the cost function in the rest of this chapter. As far as similarity measures are used that reach a minimum at correct alignment (e.g. Sum of Squared Differences), I negate them in order to use them as cost function.

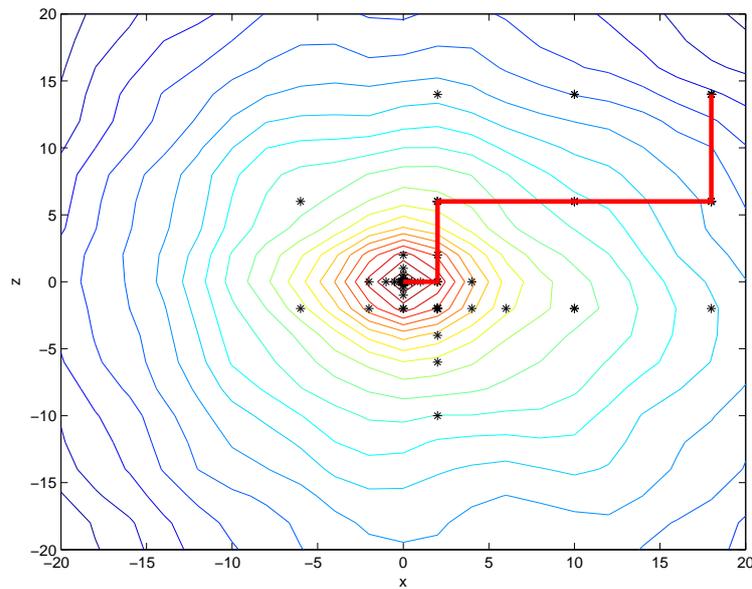


Figure 4.1: Best Neighbor Optimization run

4.2 Best Neighbor Search

This is one of the simplest optimization algorithms, often referred to as *Hill-Climbing*, too. In every iteration, the position in each degree of freedom is altered a specific step size in both directions and new values of the cost function at this positions are calculated. After having evaluated all $2n$ neighbors (with n being the number of dimensions, or degrees of freedom) the one that improves the cost function value most is chosen and set as the base position for the next iteration.

If none of the neighbors achieves a better value than the current, either a downscaling of the step size is executed, or the algorithm terminates, assuming to have found an optimal position. Applied to our registration problem where the parameter space has 6 dimensions (see section 5.1), the cost function value at 12 different locations is evaluated in every iteration. For successive iterations with a constant step size, one of the evaluations occurs at an already known position (the step "back") and can be omitted.

This algorithm basically compares the progress of the cost function in the different unit directions in the parameter space. Therefore it works the better, the more similar the cost function behaves in each direction. This requires that the parameterization of our rigid transformation is selected accordingly, see section 5.1.4 for details. The starting position for the next iteration is always a spot in the parameter space which has already been evaluated, and turned out to be the best known position. This *trust nothing that you don't see* attitude seems to be beneficial especially in our application, where the cost function is highly non-linear and sometimes wrinkly. Other methods that use a scheme to predict an unknown position in each iteration might perform worse. I used the Best Neighbor Search extensively for our registration, and it indeed performed very stable, despite being so simple (see the results in chapter 7). It is used in various other registration applications, e.g. [38, 37].

As the neighborhood in all degrees of freedom is evaluated in each iteration, one could

think about using steps combined of multiple DOFs that have the most significant ascent of the cost function. This would alter the algorithm towards the *Gradient Descent* method (section 4.3), but on the other hand have the property of setting steps into unknown locations in the parameter space. I did some sample optimization runs with the algorithm changed accordingly, which turned out to be slightly slower. There are researchers using this scheme successfully though, e.g. [30].

A similar algorithm often mentioned in literature is the *Downhill Simplex* method [32]. A simplex is a minimal geometric shape, consisting of $N + 1$ corners in N -dimensional space. A starting simplex is defined, the cost function is evaluated at the corners, and depending on the results the shape of the simplex is changed. This method is mostly known for its simple and elegant implementation, but is not supposed to be extraordinary efficient.

Integrated into the whole context of methods for solving unconstrained optimization problems, this kind of algorithm is referred to as *Pattern Search Algorithms*. These methods have in common, that they use a search pattern which is independent of the cost function in order to evaluate the cost function at specific points in the parameter space. According to the results, the position, orientation and size of that pattern is altered successively in each iteration. Torczon [42] provides a general discussion of Pattern Search algorithms, including a theory of the convergence behavior. Besides the Best Neighbor and Simplex methods, this category contains for instance the Hooke-Jeeves pattern search method, which has also been applied to rigid 2D-3D [5] and deformable 3D-3D [4] registration projects at my university.

4.3 Gradient Descent Method

A very common way to find a maximum is to step successively in the direction of the function gradient.

$$x_{new} \leftarrow \lambda \frac{df(x)}{dx}$$

The factor λ is a constant known as the learning rate. It is positive if we are seeking for a maximum, and negative otherwise. Unfortunately the success of the optimization depends very much on the right choice of λ . If it is too big, steps are taken too far and may skip the optimal position repeatedly. However, if it is set too small, the algorithm may get stuck before the optimum, as the step size depends both on λ and the absolute size of the function gradient.

There are some different approaches to use an adaptive learning rate. I used an additional factor, resulting in

$$x_{new} \leftarrow \lambda \frac{\ln\left(\left|\frac{df(x)}{dx}\right| + 1\right)}{\left|\frac{df(x)}{dx}\right|} \frac{df(x)}{dx}$$

. This limits the step size if the absolute size of the gradient is very high, which is often the case at the beginning of the optimization run. Which adaptive schemes and values are best for the optimization, depends on the specific registration problem and the cost function, and can only be determined empirically. A realistic approach would probably be to use a previously defined list of λ and adaptive parameters, defined for each similarity measure and even for each type of images to be registered. This algorithm is for instance used in [17].

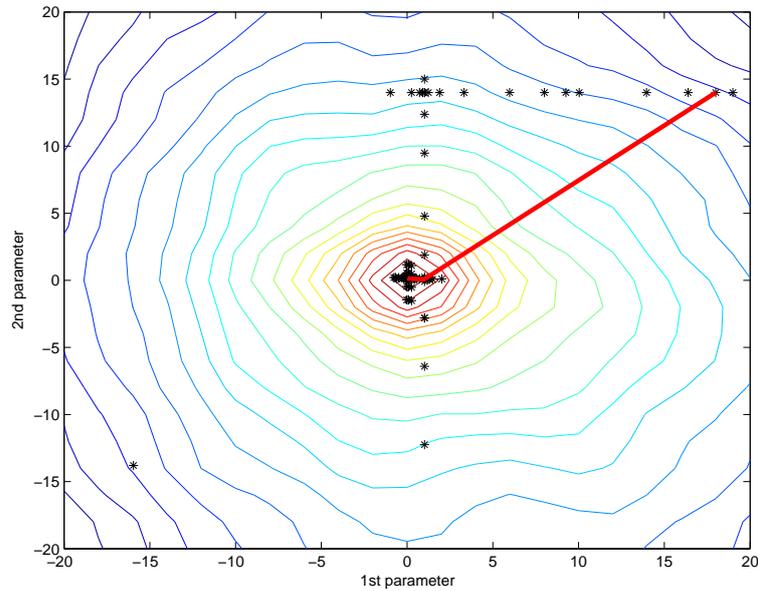


Figure 4.2: Powell-Brent Optimization run

An extension of this algorithm conducts line maximizations along the direction of the gradient, resulting in the *Steepest Descent* method. This does not rely on the empirical definition of a learning rate anymore. Every new iteration then starts from a maximum on the direction of the previous gradient, thus the component of the new gradient in that old direction will always be zero. This means that all iterative steps are perpendicular to each other, which is not very efficient. This problem led to the development of more sophisticated algorithms involving multiple old directions, the *Conjugate Gradient* methods. However, the effort necessary to implement one of those algorithms was in no relation to the expected performance, as we do not have efficient gradient calculation in our application.

4.4 Powell's Direction Set Method

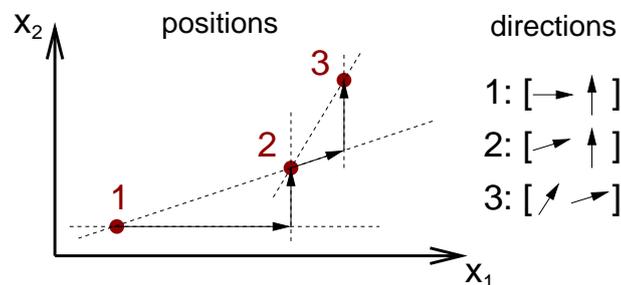


Figure 4.3: Powell direction set method in two dimensions

This algorithm starts at a given position in the parameter space, and minimizes the cost

function successively along certain directions. Therefore the problem is split up in two parts: Finding the best directions in n -dimensional space, and doing efficient line minimization on a new cost function with only one parameter. The latter problem can be solved very well with an algorithm called *Brent Line Minimization*. It uses both parabolic interpolation and golden section search, choosing dynamically in each step which method is more appropriate. It is described very detailed in [32], other methods for bracketing a minimum and root finding can be found in [26].

The other problem is now to find directions in the parameter space, so that a line minimization along one of them does not spoil the minimizations done along the former directions. Therefore, the different directions should be capable of being optimized independently. The crucial task is now to determine the best directions.

According to Powell's scheme, the first set of directions are the individual degrees of freedom themselves. After n line minimizations, the first direction is replaced with the vector that leads from the last to the current estimate, all other directions are shifted (see figure 4.3). This makes sure that the first line minimization of each new iteration is done along the way that the parameters improve most. From the old directions, the one where the biggest move happened on the previous line minimizations, gets "fired". It is most likely to be contained in the newly selected direction anyway, therefore the directions are kept as independent as possible.

The most obvious advantage of this method is that it does not need gradient information. This results in a highly reduced number of cost function evaluations, compared with gradient based methods. The only settings that need to be defined affect the line minimizer, like termination constraints. At this point it would probably make sense to use an adaptive strategy. The tolerance of the Brent line minimization could be set very high for the first Powell iteration (involving n line minimizations), and successively refined.

4.5 Multiresolution Strategy

Many optimization problems can be accelerated by starting with fast but coarse estimates of the cost function, refining it later and eventually using a fully precise yet slow evaluation. This especially applies to our registration problems, as the DRR image resolution significantly affects the performance. Often a hierarchy of sampled images at different resolutions is used for registration. The first optimization is carried out with the most rough images, terminating early due to respective abortion criteria. It is restarted at the resulting pose with the subsequently more precise images, this time with smaller tolerance values for abortion. The optimization is iteratively rerun until the finest resolution is in use.

It is important to note that a change in the image resolution significantly affects the similarity measure and accordingly the cost function. Therefore it is essential that the optimization is really restarted when proceeding to a higher level on the multiresolution pyramid. A very beneficial side-effect of using low-resolution images first is that the similarity measure usually is much smoother, and may even have less local optima compared to the full precision images. Therefore it is often not just desired to use a multiresolution strategy in order to speed up the registration, but a definite requirement to overcome problems with local optima.

It is not always necessary to implement multiresolution optimization by using different im-

age resolutions. In our case it is also possible to use less of the volume information in order to increase the speed of the DRR generation. Specifically, we use less 3D texture slices in order to obtain coarse DRRs (see chapter 6), but keep the full image resolution for not losing some of the important structural information in the X-Ray image. Similar approaches are to intentionally blur the images (see e.g. [38]) to avoid local optima. This is then termed a *multi-scale* strategy rather than a multi-resolution one.

5 Registration

In the last chapters I covered the three essential issues that need to be taken care of in order to build up a 2D-3D registration - namely DRR generation, similarity measures and optimization algorithms. However, there are some details that have to be considered in order to connect everything and make the system running, which I will present in the following.

5.1 Nature of the Transformation

We are looking for a rigid transformation in space, that describes the way from the patient's body to a world coordinate system. In our application, it is adequate to use the isocenter of the linear accelerator as center of the world coordinate system. This makes it most easy to describe the 3D placement of the linear accelerator with respect to a treatment location in the patient's body. The axis orientation is set according to medical standards, x pointing to the patient's left arm, y pointing down and z pointing towards the patient's head.

The rigid transformation that we consider, places the center of the CT volume somewhere in the world coordinate system. However, that is not the transformation that we want to modify with our registration algorithm directly, as the sensibility of transformation parameters would depend very much on the initial position. For instance, if the volume is misplaced with respect to the isocenter, a rotation around the x axis would at the same time result in a significant translation along the y and z axes. Therefore we use the following scheme for the registration:

$$T_{curr} = T_{center}^{-1} T_p T_{center} T_{world} \quad (5.1)$$

Basically we place the volume somewhere in the world coordinate system using T_{world} , and then apply another transformation T_p which sets it to the current estimate. However, we want any rotation in T_p to be centered around the volume center, which is counted in with T_{center} . T_{world} is the initial estimate for the registration, and is set manually by the user. Once the registration starts, only T_p is modified, starting with the identity transformation, $T_p = I_4$. There are now several ways to describe this rigid transformation in 3D.

5.1.1 Euler Angles

The most straight-forward representation is a six parameter vector $[t_x, t_y, t_z, \theta_x, \theta_y, \theta_z]^T$. The transformation consists of consecutive rotations around the three coordinate axes, given by the angles θ_x , θ_y and θ_z , and a successive translation along those axes, described by t_x , t_y and t_z . This parameterization is known as *Euler Angles*. It is minimal in the sense that the transformation is described with only six parameters, representing the six degrees of

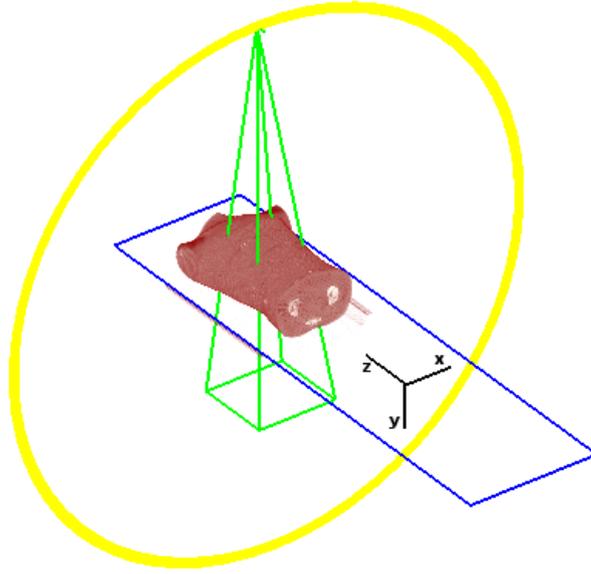


Figure 5.1: Scheme of the treatment room

freedom.

The conversion of this description to the common 4×4 matrix representation, most widely used in computer graphics, works as following:

$$\begin{aligned}
 T_p &= T(t_x, t_y, t_z) \cdot R_z(\theta_z) \cdot R_y(\theta_y) \cdot R_x(\theta_x) \\
 &= \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_z & -s_z & 0 & 0 \\ s_z & c_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_y & 0 & s_y & 0 \\ 0 & 1 & 0 & 0 \\ -s_y & 0 & c_y & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c_x & -s_x & 0 \\ 0 & s_x & c_x & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 &= \begin{pmatrix} c_y c_z & (s_x s_y c_z - c_x s_z) & (c_x s_y c_z + s_x s_z) & t_x \\ c_y s_z & (s_x s_y s_z + c_x c_z) & (c_x s_y s_z - s_x c_z) & t_y \\ -s_y & s_x c_y & c_x c_y & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5.2} \\
 & s_x = \sin(\theta_x), c_x = \cos(\theta_x), s_y = \sin(\theta_y), c_y = \cos(\theta_y), s_z = \sin(\theta_z), c_z = \cos(\theta_z)
 \end{aligned}$$

The books [7] and [10] provide detailed introductions to homogenous coordinates, which I assume to be known here. The only disadvantage of this parameterization is that it shows singularities at some specific angles. To show this, we calculate the derivatives of a rigid

transformation with respect to the rotational parameters:

$$\begin{aligned} \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} &= T_p \begin{pmatrix} x \\ y \\ z \end{pmatrix} \Rightarrow \\ \frac{\partial}{\partial \theta_x} \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} &= \begin{pmatrix} (c_x s_y c_z + s_x s_z)y + (-s_x s_y c_z + c_x s_z)z \\ (c_x s_y s_z - s_x c_z)y + (-s_x s_y s_z - c_x c_z)z \\ c_x c_y y - s_x c_y z \end{pmatrix} \end{aligned} \quad (5.3)$$

$$\frac{\partial}{\partial \theta_y} \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} -s_y c_z x + s_x c_y c_z y + c_x c_y c_z z \\ -s_y s_z x + s_x c_y s_z y + c_x c_y s_z z \\ -c_y x - s_x s_y y - c_x s_y z \end{pmatrix} \quad (5.4)$$

$$\frac{\partial}{\partial \theta_z} \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} -c_y s_z x - (s_x s_y s_z + c_x c_z)y - (c_x s_y s_z - s_x c_z)z \\ c_y c_z x + (s_x s_y c_z - c_x s_z)y + (c_x s_y c_z + s_x s_z)z \\ 0 \end{pmatrix} \quad (5.5)$$

If we now specify the angles θ_x and θ_y accordingly, some of the derivatives become identical:

$$\begin{aligned} \theta_x = \frac{\pi}{2}, \theta_y = -\frac{\pi}{2} \Rightarrow \\ \frac{\partial}{\partial \theta_x} \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \frac{\partial}{\partial \theta_z} \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} s_z y + c_z z \\ -c_z y + s_z z \\ 0 \end{pmatrix} \end{aligned} \quad (5.6)$$

Thus the three rotations are not independent of each other anymore - the angles θ_x and θ_z rotate around the same axis. However, in the architecture of our system we will never be using a patient-to-world transformation that includes those critical angles, as we are using a rigid transformation T_p relative to the starting estimate (equation 5.1).

5.1.2 Unit Quaternion

Representing 3D transformations with quaternions is very common in computer graphics and robotics. A rigid transformation can be described as a vector $[t_x, t_y, t_z, q_x, q_y, q_z, q_w]^T$, which combines three translational parameters and the four elements of a quaternion, denoting a rotation in space.

Quaternions are defined in the form $Q = iq_x + jq_y + kq_z + q_w = [(q_x, q_y, q_z), q_w]$, where i , j , and k have the following properties:

$$\begin{aligned} i^2 = -1, j^2 = -1, k^2 = -1, \\ ij = k, ji = -k, jk = i, kj = -i, ki = j, ik = -j \end{aligned} \quad (5.7)$$

All basic arithmetic operations can be derived from the properties in equation 5.7, the behavior is consistent with the one of complex numbers, as quaternions can be seen as an extension of them. Refer to [1] for a more detailed introduction to quaternions and their use in 3D computer applications. Here I will just present the very convenient way of describing a rotation in 3D with them.

For this to work, the quaternion has to be normalized, i.e. $q_x'^2 + q_y'^2 + q_z'^2 + q_w'^2 = 1$. This can be achieved easily by dividing by the size of the original quaternion

$$\begin{aligned} q_x' &= \frac{q_x}{|Q|}, q_y' = \frac{q_y}{|Q|}, q_z' = \frac{q_z}{|Q|}, q_w' = \frac{q_w}{|Q|} \\ |Q| &= \sqrt{q_x^2 + q_y^2 + q_z^2 + q_w^2} \end{aligned} \quad (5.8)$$

. If this is valid, the vector (q_x', q_y', q_z') represents the axis of rotation, and the angle θ is denoted indirectly by $\theta = 2 \cos^{-1}(q_w')$. The respective rotation matrix is

$$R(q_x, q_y, q_z, q_w) = \begin{pmatrix} 1 - 2q_y'^2 - 2q_z'^2 & 2q_x'q_y' - 2q_w'q_z' & 2q_x'q_z' + 2q_w'q_y' & 0 \\ 2q_x'q_y' + 2q_w'q_z' & 1 - 2q_x'^2 - 2q_z'^2 & 2q_y'q_z' - 2q_w'q_x' & 0 \\ 2q_x'q_z' - 2q_w'q_y' & 2q_y'q_z' + 2q_w'q_x' & 1 - 2q_x'^2 - 2q_y'^2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5.9)$$

. As this only valid for unit quaternions, the normalization in equation 5.8 should be carried out for every set of parameters implicitly. We can therefore use the original scaled quaternion $Q = [(q_x, q_y, q_z), q_w]$ as rotational parameters. However, this results in having seven parameters for the optimization - one more as degrees of freedom. In my opinion, this imposes an unnecessary burden for the optimization algorithm and is the reason for not using this parameterization.

5.1.3 Rotation Axis Representation

Another way of representing the rigid transformation, avoiding both singularities and over-parameterization, is to describe the orientation with a 3-vector (r_x, r_y, r_z) . This vector defines a rotation axis in space, similar to the quaternion usage above. However, in this case the size of the vector indicates the rotation angle [36]. The translational part of the transformation is denoted as a 3-vector (t_x, t_y, t_z) , as usual. This results in a transformation description that is both minimal in terms of the number of parameters, and has no singularities. A disadvantage is, however, that it is not very comprehensible for users, as the amount of rotation is encapsulated in all three values of the rotation vector.

5.1.4 Parameter Scaling

Decided to use the six-value representation including the three translational displacements and three rotation angles, there is still the issue how to use those parameters for the respective cost function. For any optimizer to work best, changing each of those parameters should have a comparable affect on the cost function value. This implies that altering any parameter value should change the similarity of the two images in the same order of magnitude. In the first place we need to introduce an appropriate mapping between the translational and rotational parameters. To derive this mapping, we use the corners of the bounding box of the volume used in the registration. We then claim that translating the box a specific amount should displace those corners equally than if we rotate the box around the center about the same amount. Making the simplification of using a cube with edge width d around the origin and considering only one corner point p , the mapping can be derived as follows:

$$\begin{aligned}
t_x &= \Delta p \\
\theta_x s &= \Delta p = \theta_x d \frac{\sqrt{3}}{2} \\
s &= d \frac{\sqrt{3}}{2}
\end{aligned}$$

t_x is a translation in millimeters, θ_x a rotation in radiant, Δp is the displacement of the point p and s is the mapping between translational and rotational values we are looking for. As we would like to have a mapping which is convenient to use, I propose to describe translations in millimeters and rotations in degrees. This corresponds to a scaling of $s = \frac{180}{\pi}$. Therefore the width of the respective bounding box satisfying the former constraints is $d = \frac{360}{\pi\sqrt{3}} \approx 66mm$.

This size is a bit smaller than the image regions that we are focusing for registration, but still lies in the same order of magnitude. Given this and some similarity measure plots that indicate comparable behavior for both altering the pose in millimeters and degrees (see Appendix) justify the use of this mapping.

5.2 Use of 2D Transformations

Searching a six-dimensional parameter space is very inefficient. If there is any way to reduce the number of parameters used in the optimization, we definitely should have a look at it. If we define our rigid transformation with respect to the DRR image, i.e. the camera points towards the y-axis and the image plane is perpendicular to it, two of the rotational parameters are out-of-plane, namely the rotation around the x- and z-axes. Altering those rotations significantly changes the content of the DRR image, as the respective projection is re-oriented in the CT volume. The remaining four parameters can now be considered as 2D image transformation operators. Rotation around the y-axis results in rotation around the image center, the x and z translations shift the image center accordingly, and the y translation scales the image content.

This can lead to various strategies. If the optimization is now performed on the six parameters defined like above, a new DRR has to be generated only if one of the two out-of-plane rotations is being altered. Otherwise, the desired pose change can be satisfied with a very cheap 2D image transformation, namely to translate, rotate or scale the existing DRR image. Using for instance the best neighbor optimizer, where the different parameters are altered separately, would result in reducing the number of necessary DRR generations to one third, as changes in only two out of six degrees of freedoms cause a new DRR! Another option would be to run the optimization using only the two out-of-plane rotations. For each iteration, one could perform an additional optimization on the remaining parameters in very little time.

Unfortunately, there are some problems with this approach. First of all, replacing the mentioned four parameters with 2D transformations is only mathematically correct for parallel projections. As we have to use a perspective camera model in order to achieve realistic radiograph reconstructions, using those 2D operations introduces some errors. Especially displacing the DRR center too much towards the edge would result in a significantly wrong

perspective. On the other hand is the perspectiveness of the X-Ray imaging device not really immense. With our calibration data of the Primus linear accelerator, we obtained a field of view (FOV) for the imaging device of just 12 degrees (see section 7.1). In my opinion, this justifies the approach of using the 2D transformations, making it nevertheless desirable to have some estimation on the resulting errors.

From the fact that the fast 2D operations are only approximations follows at the same time, that it is not easy to express the relationship between the respective parameters of the 2D and 3D movement correctly. The rigid transformation in 3D has to be approximated by a combination of out-of-plane rotations and a 2D transformation:

$$P_{world}^{image} T_{CT}^{world} = T_{2D}(x, y, s, \alpha) P_{world}^{image} T_{3D}(\beta, \gamma) \quad (5.10)$$

One approach by Sarrut and Clippe [34] uses optimization of a least squares term, denoting the distances of random points projected both by the 3D transformation and the desired combination of 3D rotations and 2D transformations:

$$\operatorname{argmin}(x, y, s, \alpha, \beta, \gamma) \sum_{x \in H} \left(P_{world}^{image} T_{CT}^{world} x - T_{2D}(x, y, s, \alpha) P_{world}^{image} T_{3D}(\beta, \gamma) x \right) \quad (5.11)$$

During optimization, they determine the respective T_{2D} and T_{3D} parameters for the current rigid pose T_{CT}^{world} in every iteration by minimizing equation 5.11 with a Powell optimizer. The similarity measure then compares the X-Ray image with a DRR precomputed for the angles β and γ and transformed accordingly. The disadvantage of this scheme is the least-squares optimization of the parameters that has to be done in every iteration of the whole registration process.

I suggest to directly optimize on the parameters x, y, s, α, β and γ . The result still can be backtransformed to the parameters of a single rigid transformation, using equation 5.11 or a similar scheme. Unfortunately we ran out of time to evaluate this approach.

Interpolation artifacts may be a problem as well, as we apply 2D transformations on existing DRR images. Translation in x and y can still be conducted in multiples of whole pixels, but this is not possible for scaling and rotation. Using the common similarity measures for comparing scaled or rotated images therefore involves some interpolation of the DRR, possibly affecting the stability of the measure.

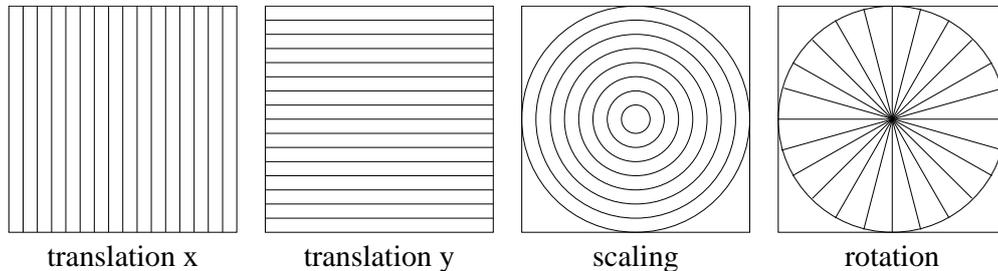


Figure 5.2: Structures for assessing alignment in specific 2D transformations

I developed a scheme of rapidly comparing two images with respect to 2D transformations on one of them, that I would like to introduce shortly. Basically it works by applying a similarity measure not directly on the image pixels, but on whole columns, rows, rings and rays defined on the images (figure 5.2). The sums of all pixels in each of those structure elements are used as intensity values for the similarity measure. According to the transformation, the corresponding structure elements or its neighbors are used in the measure. As there are not many intensities available at all, and they each consist of many averaged pixels, the measure should not be too sophisticated. I used the Sum of Squared Differences (SSD). The advantage of those structures is that two parameters at a time are independent from each other: the x-and y-translation, and the scaling/rotation, respectively. Therefore it is possible to find the optimum in this four-dimensional search space with few iterations. Even if this method is not used for supporting the registration, it can be of use for manual registration. A plot of the measure with respect to those 2D transformations, generated in negligible time, indicates directions in the parameter space where the alignment is improving.

5.3 Region of Interest

Depending on the imager for the reference image and other constraints, the need for defining a region of interest may arise. This means that not all image pixels are considered for computation of the similarity measure (and, if applicable for the DRR computation, too).

5.3.1 Image Borders

In my computer simulations using the skull CT data, there is usually a significant black border around the actual X-ray image or DRR of the skull. The question arises if it makes sense to omit the respective black image pixels in the similarity measure computation. Otherwise the registration results may be better than in a realistic environment, where the images usually have no border (as they focus on a certain spot within a CT data set & the patient). The following results show plots of the similarity between the usual skull DRR and reference images, while the DRR is being shifted along the X axis. Different schemes to omit black pixels are applied.

- Ignoring all pixels with a zero intensity in at least one of the images does not work very well. The size of the considered region heavily depends on the displacement between the two images. The larger the distance between the object in the two images, the smaller the region used in the computation. This makes it possible that an optimization bails out, as the similarity measure even increases with rising distance (topmost line in the diagram).
- A more evident approach is to use all pixels where the reference image provides information. This imposes that the ROI does not change once the reference image is defined.
- Going further, one could only omit those pixels where both image intensities are zero. Even more image information than in the latter scheme is used, therefore the similarity measure curve is more steep.

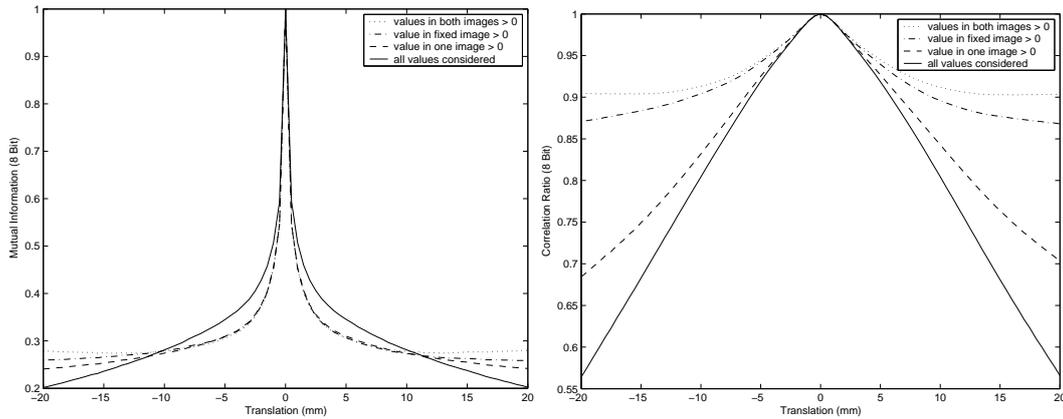


Figure 5.3: Similarity measure plot, with various schemes to omit black image regions applied

- Using the full image content, at last, makes the measure even more stable. This is very obvious, as all the differences between black and informative regions contribute to the joint histogram.

Another 2D-3D registration project (using CT and C-Arm fluoroscopy) currently done at my university [5] uses a threshold value in order to determine a binary mask on the DRR image, updated with each new iteration. This binary mask is expanded with a dilation operator of a selected size, e.g. 5 pixels, and defines the region of interest for the similarity measure computation. This seems to be a good compromise of the strategies listed above.

5.3.2 Collimator Settings

If the X-Ray images are taken as a byproduct of a patient treatment with a linear accelerator, a collimator may be in use. Depending on its settings, specific parts of the images acquired by the detector will be empty. Of course one should exclude those image regions from the similarity assessment. To support any collimator architecture, e.g. grid-based collimators, it is most universal to use a pixel-wise definition of the image regions, most easily implemented with a bit mask.

5.3.3 Content Based Mask

Another option is to use only parts of the images that seem to contain relevant information. This can on the one hand be done by manually selecting image features [37]. However, in our application we want to avoid too much interactivity by the user. On the other hand, a mask can be automatically created using for instance image gradient information, so that only image regions that actually contain a certain amount of information, are being considered [21]. A similar technique is already involved by using the *Variance-Weighted Sum of Local Normalized Correlation* measure, as it considers the variance of a rectangular region around

each pixel for determining the influence on the whole value.

For the patient positioning application, some kind of segmentation might be needed anyway. In order to reduce the influence of soft tissue structures on the registration, a simple threshold-based segmentation is usually able to shrink the volume content to only bony structures. This has at the same time the advantage that the DRR rendering is sped up.

6 Implementation

6.1 User Interface

Our software should provide a convenient user interface both for manual and automatic registration. We have a multiple-view architecture. The user loads a CT volume out of the patient database. This corresponds to the document, only one can be loaded at a time. On the right side of the screen is a dialog bar, where most of the settings for user interaction and registration can be adjusted. With some menu points, the other views can be loaded, which are summarized below.

- *Global View*: A three-dimensional visualization of the treatment room arrangement. The volume is being displayed (VRT technique, transfer function adjustable), with the volume center initially located on the isocenter of the linear accelerator. The patient couch and one or two cones symbolizing the source-detector path for the X-Ray acquisition are visualized, too. The sources, i.e. the cone tips are placed on a circle around the isocenter, reflecting the possible gantry setting ($0 - 360^\circ$) of the linear accelerator. The user can use the mouse to either navigate through this view (*Move Scene* mode), or to change the 3D pose of the volume (*Move Data* mode), resulting in a different pose for the DRR generation.
- *DRR View*: Displays the DRR in an OpenGL window. The rendering result is stored in a 8 bit texture, which in turn is displayed on this view. Therefore the size of this view does not have to correspond with the actual DRR size, as the texture is being stretched automatically for the display. For stereo registration, the two DRRs are displayed side by side, or alternatingly on the same place, if 3D shutter glasses are used. The user may adjust the window, level and transparency settings for the DRR computation with the mouse by clicking and dragging in this view.
- *Reference View*: Contains the fixed image(s). The user can either just create a view with a copy of the currently displayed DRR for synthetic experiments, or load images from a file. Like in the DRR view, window and level adjustment is done with the mouse. If images are to be loaded from a file, the user can choose between files in DICOM format or raw image files without header information. In the latter case, the usual resolution of 1024x1024 is assumed. Loading raw image data is especially useful if the X-Ray images are modified e.g. with Matlab, what we had to do for the image rectification (see section 7.1).
- *Registration View*: This view displays the values of all activated image similarity measures, together with a diagram of their trend. If Mutual Information or Correlation Ratio is in use, the two histograms and the joint probability distribution is displayed,

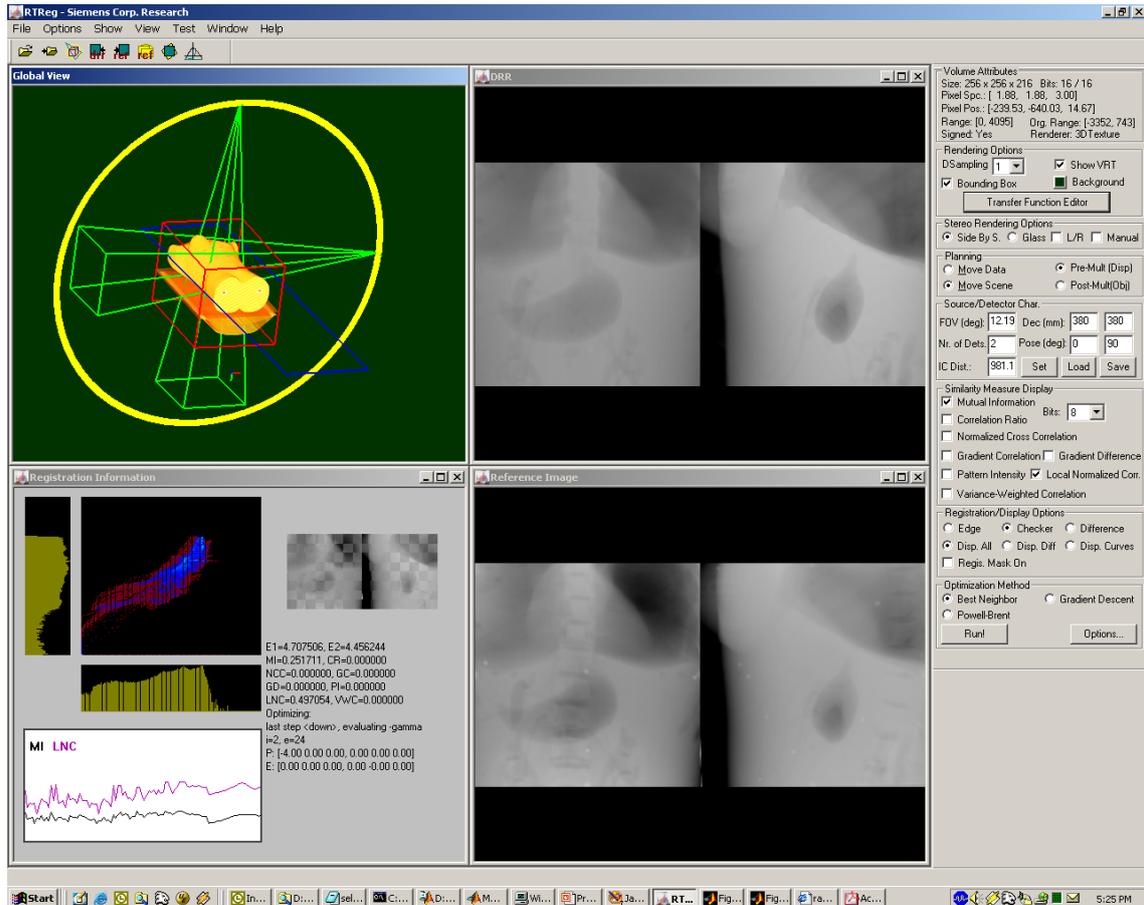


Figure 6.1: Screenshot of the RTReg Application

too. Furthermore a scaled difference image is shown. All this information is very helpful as assistance for manual registration.

6.2 Our DRR algorithm

As the focus of our application is both on interactive and automatic registration, our DRR algorithm should in the first place be able to generate full-size DRRs for visual display in realtime. Therefore we did not consider sparse sampling approaches as used in [49]. Other possible means to create more efficient DRRs would be Shear-Warp Factorization (which is an approximation and therefore produces some artifacts, that make it less apted for precise registration), and using a Transgraph data structure, as done in [24]. Refer to section 2.2 for further details on those techniques.

We compute the DRRs with a hardware-accelerated algorithm, based on [24]. The volume slices are stored in the texture memory of a NVidia GeForce FX graphics card, and combined using its Texture3D extension. So far the algorithm can handle volume sizes up to 256 x 256 x

256 voxels. The voxels are stored with 16 bit precision. For the accumulation, a window and level setting is applied on the texture slices. The "drawing" of the transparent Texture3D slices still takes place with 16 bit values, the outcome then is saved with a precision of 8 bit. Furthermore, the transparency of the individual slices to be combined may be adjusted.

The actual rendering result is not directly drawn into the OpenGL context, but into another 8 bit texture. This has two advantages. First, the resulting image can be displayed very easily, OpenGL taking care automatically of scaling the texture onto the specific window. Future work could use this feature to draw the texture with any translation, scaling and rotation, interpolating the image pixels in literally no time, as this is done by the GPU. Therefore speedups with 2D transformations as described in section 5.2 could be implemented very easily. Secondly, the image can be retrieved rapidly out of the texture buffer for further processing (in our case similarity measure computation).

With this algorithm we are able to produce high quality DRRs that look very similar to regular X-Ray or portal images. Using a relatively small window size (about 900) and transparency (0.01) we achieve the best results. See e.g. figure A.3 in the Appendix.

6.3 The Class SimilarityMeasures

6.3.1 Overview

For the Radiation Therapy application it is very preferable to calculate multiple measures at the same time, in order to allow extensive validation & combination of different measures. As several calculations are needed for more than one measure at the same time, I implemented one class that is able to calculate all measures discussed in chapter 3 at the same time. Different measures may be enabled/disabled by setting correspondent flags. This makes sure that the computation is as fast as possible, and that the image information is scanned as few times as possible. Following an overview of the sequence, assuming all measures are enabled.

- First pass through the image data:
 - Intensities are summed up for the mean and variance used in NCC
 - Absolute and squared intensity differences are summed up for SAD, SSD
 - For Pattern Intensity a difference image is created.
 - For Mutual Information and Correlation Ratio the joint histogram and the image histograms are build.
 - For Gradient Correlation and Gradient Difference the four required gradient images are calculated.
- Second pass through the images, if Normalized Cross Correlation is enabled. Given the mean values, the variances and the correlation are calculated.
- For Pattern Intensity another pass through the difference images is executed.
- Out of the histograms and the joint histogram the MI and CR values are computed.

- For Gradient Correlation, two further scans through the gradient images are needed (similar to NCC). As Gradient Difference works best with a σ parameter based on the gradient variance, GC is also calculated in this case.
- For Gradient Difference, the sum over the gradient difference image is computed.

I implemented the main calculation method according to this workflow. The class is templated with the image data type, therefore the user can choose between 8, 16 or 32 bit images, signed or unsigned. Furthermore for each image a minimum value and the number of bits used is stated, enabling different kinds of images to be measured. The class internally subtracts the minimum value and shifts one of the images accordingly, so that equivalent values are compared.

6.3.2 Algorithmic Aspects

Entropy calculation The calculation of the image entropies and the joint entropies is similar, basically it is a sum over intensities:

```
for (int i = 0; i < size; i++) {
    double p = (double)hist[i] / size;
    if (p > 0.0) entropy -= p * log(p);
}
```

If certain intensities do not occur, we can ignore the addend, because of $\lim_{p \rightarrow 0} p \cdot \log p = 0$

Speedup of Pattern Intensity Pattern Intensity runs through every pixel in the difference image and sums up with a $\sigma^2/(\sigma^2 + diff^2)$ scheme over all differences within a circular neighborhood. Already for a radius $r = 3$ (see section 3.2.1) the constraint $(x-v)^2 + (y-w)^2 < r^2$ results in 28 considered neighbors per pixel. If we look at this closer, we can see that every pair-wise pixel difference is considered twice, each time one of the pixels being the center of the neighborhood kernel. This allows us to omit half of the calculations. While scanning the image from top to bottom and from left to right, at each pixel position I consider only the differences to the neighbored pixels to the right and to the bottom, as illustrated by figure 6.2.

This basically results in the following algorithm for calculating Pattern Intensity:

```
pi = 0; sum = 0;
for (int y = 0; y < height; y++) {
    for (int x = 0; x < width; x++) {
        for (int yy = y; yy <= y + radius; yy++) {
            for (int xx = x - radius; xx <= x + radius; xx++) {
                if ((xx >= 0) && (xx < width) && (yy < height)
                    && ((yy > y) || ((yy == y) && (xx > x)))
                    && ((x-xx)*(x-xx)+(y-yy)*(y-yy) <= radius*radius)) {
                    double d = diff[y, x] - diff[yy, xx];
```

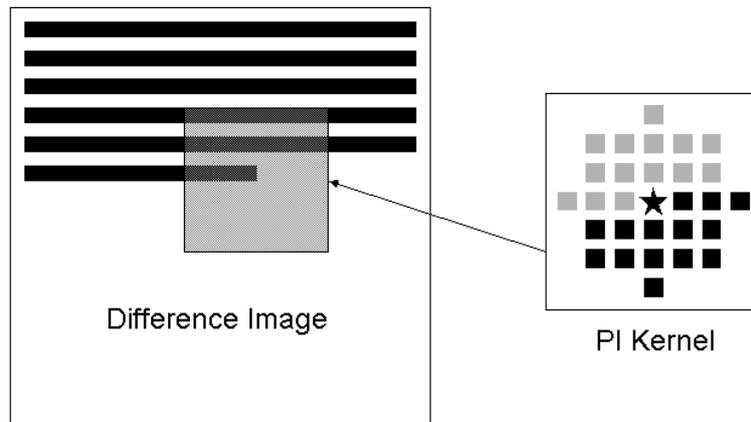


Figure 6.2: Kernel for computation of neighborhood differences for Pattern Intensity

```

        pi += sigma / (sigma + d*d);
        sum++;
    }
}
}
}
}
pi /= sum;

```

Stereo Case Our application supports registration of biplanar X-Ray images with known geometry between the images (The linear accelerator just acquires portal images from different gantry angles). The most elegant way to compute the similarity measures on now each two DRR and X-Ray images, respectively, is to treat them as one image. Therefore we append the images horizontally, which is the way we display them in our application anyway (see screenshot, figure 6.1). In that case the gradient- and neighborhood-based measures have to omit some columns in the middle of the combined image, where the two individual X-Ray images are connected. For instance, the 3x3 Sobel filter matrix applied exactly on that border would produce gradient information that we surely do not want to use in our registration. It is the same issue with the Pattern Intensity kernel and the window for the Local Normalized Correlation Measures. Therefore I inserted a `SetStereo(bool flag)` method which sets a flag indicating if we are using stereo images. In that case, some additional source code lines prevent the use of the respective middle columns.

Region of Interest In order to have the maximum flexibility for the definition of a region of interest (ROI) in the images to be registered, I implemented a binary mask denoting every pixel as included or not included in the similarity measure computation. Again, some adaptations had to be made for all measures that use spatial information.

By the time that the user defines a mask, a binary eroded mask is generated for the gradient based measures. A pixel is included in that mask if the pixel and its 8 neighbors are included

in the original mask, making sure that the 3x3 Sobel matrix for the gradient image generation is always defined.

A similar mask is computed for the Local Normalized Correlation measures. Here, a pixel is included if at least a specific number of pixels from the respective LNC window is included in the mask. To claim that the whole window should be in the mask is too rigorous, in my opinion. There are possible applications where the Region of Interest would exclude many small structures like calibration markers, placed close to each other. In that case a much larger part of the images would be neglected in the measure computation, as the respective LNC windows would not fit in between those structures. Using the adapted mask for LNC computation, the number of used pixels in each window can vary. Therefore I weigh each correlation value with the number of considered pixels in the respective window.

For Pattern Intensity the adaption is easier. There is no need to create a special mask to make sure that the circular PI kernel is always defined. As every addend of the PI formula (equation 3.7) is equivalent, we just have to check if the respective two pixels to be compared are included in the mask, and keep track of the total number of addends in order to eventually divide by it.

6.4 Speed Issues

To know what execution time is feasible for our registration problem, I measured the computation time of the DRR generation at different volume resolutions and downsampling rates. The CT data set used is a Rando® head phantom (see figure 7.4 in the next chapter). The machine was a 2.4GHz Pentium IV, with a NVidia GeForce FX 5600 (256MB) graphics card.

	1x	2x	4x	8x
256x256x287	340ms	200ms	130ms	95ms
128x128x143	130ms	95ms	78ms	68ms
128x128x71	123 ms	91 ms	75ms	67ms

The overhead caused by the calculation of various similarity measures is listed in the following table.

measure	MI4	MI8	CR4	CR8	NCC	GC	GD	PI
time	2.5ms	10.3ms	2.5ms	8.6ms	2.5ms	9.4ms	13ms	101ms

The additional time for displaying the histograms, joint histogram and difference bitmap is about 6.5ms per frame.

7 Validation

7.1 Camera Calibration

As mentioned earlier, the calibration of the X-Ray imaging device has to be known precisely in order to perform a reasonable registration. Therefore the intrinsic camera parameters, reflecting the image source-detector arrangement have to be computed. If they can not be retrieved solely based on technical specifications available for the device, some *camera calibration* is needed. The result of this procedure is the knowledge of the correct mapping of any 3D points in space onto the 2D image plane of the imager, or in other words the pixel coordinates in the resulting image. For X-Ray devices it is always a perspective projection, which can be expressed in homogenous coordinates with a 3x4 matrix:

$$\begin{pmatrix} \lambda u \\ \lambda v \\ \lambda \end{pmatrix} = \begin{pmatrix} k_u & 0 & u_0 & 0 \\ 0 & k_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (7.1)$$

The book of Faugeras [7] provides a very comprehensive covering of modeling and calibrating projective cameras.

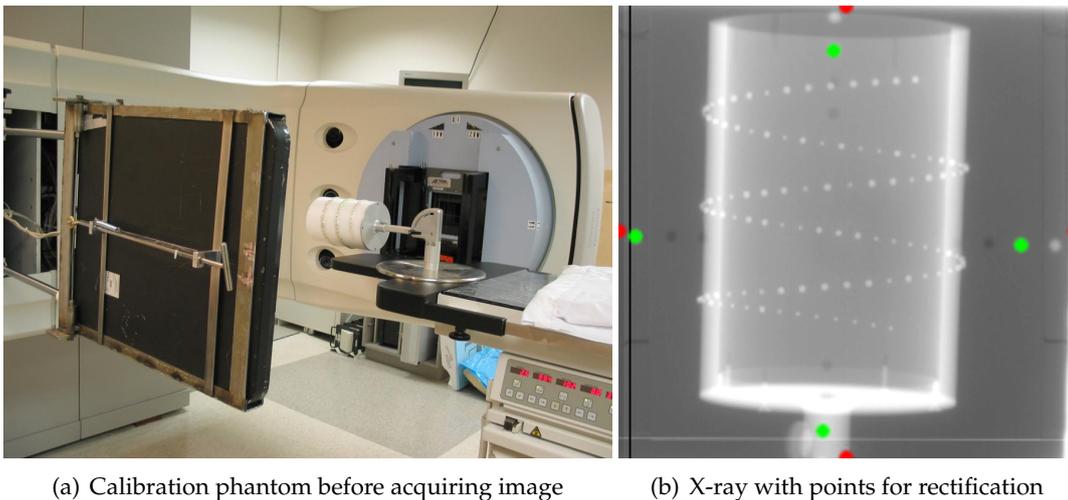


Figure 7.1: Calibration phantom used for camera calibration

We used a Siemens calibration phantom with many radioopaque metal markers, visible both on CT scans and X-Ray images, figure 7.1(a). We obtained a CT scan of it, and a set

of 3D marker positions that had been segmented automatically. We then acquired portal images of the phantom from various gantry angles. We used the Siemens Primus linear accelerator in use in UPenn Hospital. At this specific device, the original X-Ray detector has been removed and replaced by a digital solid detector manufactured by another company. This detector has been mounted by welding it on the accelerator frame, resulting in a not very stable construction (visible in figure 7.1(a)). Therefore we had to correct for the tilt of the detector, which is supposed to be different at different gantry angles. In order to achieve this, we used an acrylic glass tray with a marker pattern and inserted it right under the X-Ray source, so that the pattern appeared on the acquired images. After taking all the needed images with our phantom, we corrected each of the images, so that the outer markers of the tray were positioned exactly in the middle of the respective edges of the images, using a homographic projection according to [29]. Figure 7.1(b) shows one of the images taken with the linear accelerator. The image is rectified so that the green points, originating from the acrylic glass tray, become the edge centers (red points). We conduct this rectification with every acquired image, and therefore calculated the perspective projection with respect to the **rectified** images. Using many 2D and 3D marker positions from the calibration phantom, we calculated the 3×4 projection matrix P_{world}^{image} which maps any point from the isocenter-based world coordinate system to the respective pixel position on the rectified X-Ray images.

7.2 Obtaining Ground Truth

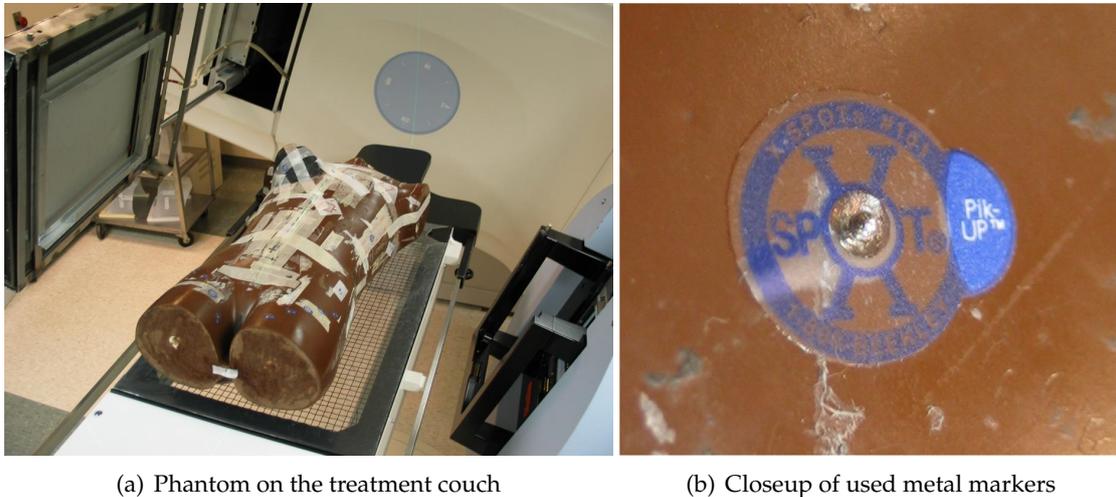


Figure 7.2: Rando phantom and marker used for ground truth computation

When assessing the quality of a registration algorithm, it is very essential to have some data where the correct alignment is known within a certain precision. This is usually achieved by attaching fiducial markers to the imaged objects and registering those markers to the corresponding data sets. Often tools like a stereotactic frame are used, too. Of course it is most desirable to have clinical data of real patients with ground truth information. This is very hard to achieve, though. Moreover, obtaining the correct alignment with a

higher precision often comes with higher invasiveness, too.

We were able to compute ground truth information using our UPenn phantom data. We attached many small metal ball markers (figure 7.2) to the surface of the phantom. These markers can be seen both in the CT and X-Ray images taken of the phantom.

Noting \vec{x}_i the pixel position of marker i in the image, and \vec{y}_i the position of marker i in the CT coordinate system, we can state

$$\forall i : \quad \vec{x}_i = P_{world}^{image} T_{CT}^{world} \vec{y}_i \quad (7.2)$$

Now we are looking for the rigid transformation T_{CT}^{world} that satisfies this equation. We parameterize it in the same way than in our application, using the Euler angles representation. We then want to minimize the RMS distance between the CT marker positions projected using equation 7.2, and the respective positions in the X-Ray image.

$$\sum_{i=1}^n |\vec{x}_i - P_{world}^{image} T(t_x, t_y, t_z, \theta_x, \theta_y, \theta_z) \vec{y}_i|^2 \quad (7.3)$$

We achieved this minimization using the Levenberg-Marquardt Optimizer in Matlab. We used $n = 4$ points for the ground truth estimation. The final RMS error after optimizer convergence was [ask Ali], predicting an overall TRE error of the Ground Truth pose of about x mm.

7.3 Expressing Alignment Errors

In order to assess the registration error while having ground truth information, it makes sense to use an error transformation T_{error} [49]. This is the transformation that has to be applied in order to take the final registration pose to the ground truth pose:

$$T_{GT} = T_{error} \cdot T_{result} \Rightarrow T_{error} = T_{GT} \cdot T_{result}^{-1} \quad (7.4)$$

If this transformation is expressed in Euler angles $(e_x, e_y, e_z, e_\alpha, e_\beta, e_\gamma)$, the displacement error in the different degrees of freedom is denoted as millimeters and degrees, respectively. This is most useful when we need a distinction between translational and rotational degrees of freedom. One single error value can be computed from this representation e.g. as RMS sum, $\sqrt{e_x^2 + e_y^2 + e_z^2 + e_\alpha^2 + e_\beta^2 + e_\gamma^2}$. However this is not correct in terms of a physical equation, as it should not be summed up over millimeters and degrees at the same time.

The other common ways to express registration errors are *Target Registration Error* and *Fiducial Registration Error* [13]. The former one just tells the (euclidian) distance in space between an important point in the fixed image, and the corresponding point in the registered image. This point usually is of medical relevance, like the center of the target area for a radiation treatment, or the center of an important object in the patient's body. The latter expresses the distance between multiple points in the images. But these points are chosen now rather because of their locatability than any clinical relevance. They may just be the positions of a set of markers implanted in the patient's body, or attached to a phantom. They may also be positions of marker points in a stereotactic frame encaging the imaged object. Then the error

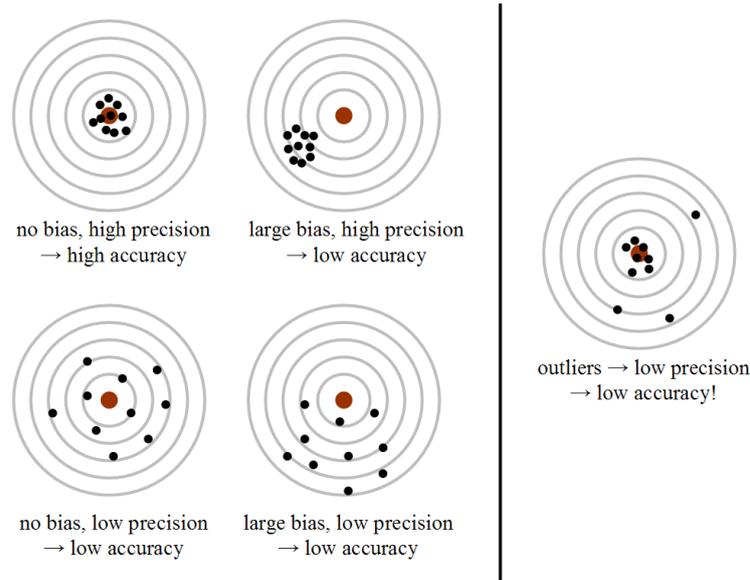


Figure 7.3: Affect of bias, precision and outliers on the accuracy of an algorithm

is usually expressed as RMS value of the distances of the individual points.

$$TRE = T(p) - q$$

$$FRE = \sum_{i=1}^N |T(p_i) - q_i|^2$$

T is the transformation found by the registration, p the significant point in the moving image, q the point in the fixed image. For FRE, p_i and q_i are respective sets of points. If the images even lack any markers, then FRE can be computed using some bounding points surrounding the used 3D dataset. In our case we can use the 8 corners of the bounding box of the CT volume.

As I still want to distinguish between the different parameters of the transformation, I will stick to using the RMS value of the individual components of the error transformation T_{error} , as explained above.

Another important property of the registration result is the certainty of resulting at the same pose, if started from different initial estimates. This, often termed as *precision* of the registration, can be expressed well with the standard deviation σ of the error from many registration runs, launched from different initial estimates. In combination with the absolute error (i.e. bias), it can tell us what kind of accuracy our algorithm delivers. Figure 7.3 illustrates it in two dimensions. No matter what combination of bias and precision may be the case, it can be furthermore ruined by outliers (right drawing). Therefore it might make sense to mention outliers separately, successively discarding them for statistics.

7.4 Experiments

As a lot of evaluations of mainly the performance of the different similarity measures will be done in the next sections, table 7.1 shows a list of the abbreviations used for easier reference of those measures.

MI _x	Mutual Information with histogram bit size x
CR _x	Correlation Ratio with histogram bit size x
NCC	Normalized Cross Correlation (also known as Correlation Coefficient)
GC	Gradient Correlation
GD	Gradient Difference
PI	Pattern Intensity
LNC	Sum of Local Normalized Correlation (sometimes denoted SLNC)
VWC	Variance-Weighted Sum of Local Normalized Correlation

Table 7.1: Abbreviations used for the similarity measures

The subsequent experiments all follow the same scheme. For any pair of CT and (real or reconstructed) X-Ray image, an initial estimate is defined. This is either the ground truth pose, if available, or some manually found pose with good alignment. Then many registrations are executed from starting poses randomly displaced around the initial estimate. For every random pose to start from, I run a separate optimization with each similarity measure. This both enables me to study the quality of the resulting alignment (bias) for each measure, and the certainty (precision) of finding this pose with many different tries, including the number of isolated wrong results (outliers). With a runtime of one to four minutes, it takes some time to acquire a reasonable amount of information. Usually I launched new experiments on Friday afternoon on my work computer, and stopped them on Monday morning.

7.4.1 Computer Simulations with Head Phantom (#1)

In this kind of experiment the image pairs to be registered are created by the computer. This provides basic, but nevertheless important information about the behavior of the registration algorithm. The resulting accuracy may be seen as an upper limit to registration with clinical data. This is because this synthetic problems are usually easier to solve than if using real data. In my case I simulate the X-Ray image with a DRR image from our CT dataset.

I am using a Rando® head phantom for this set of experiments. I load the volume with resolution 256x256x143. The X-Ray image is simulated with a copy of the DRR at this pose (see figure 7.4). The volume is now randomly displaced up to 20 mm or degrees in each DOF, and the Best Neighbor optimizer is run to find the correct alignment again (initial stepsize 8 mm/degrees). For each displacement the optimization is run 8 times repeatedly with all different similarity measures. Knowing the exact pose of the reference image (as it is actually a DRR) the success of the registration can be assessed with a ground truth value.

Diagram 7.5(a) shows the RMS values of each the translational and rotational components of the error transformation, 7.5(b) the errors in each degree of freedom for two specific measures, Mutual Information and Local Normalized Correlation. In the latter case the errors

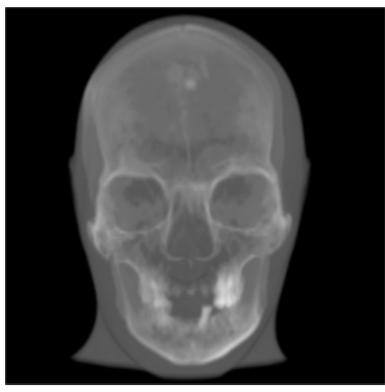
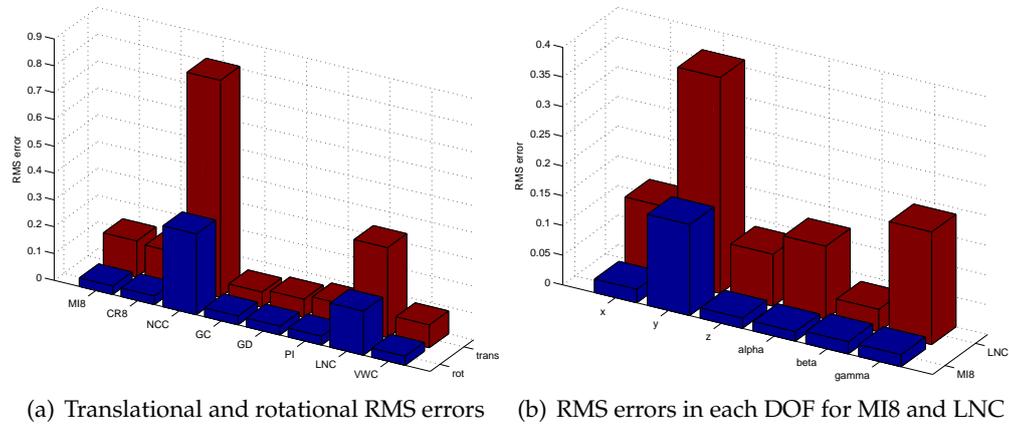
	Setting	Value
	data set	Rando Head Phantom
	used volume size	256 x 256 x 143
	voxel spacing	1.88mm x 1.88mm x 3.0mm
	start pose	unchanged in isocenter
	DRR window/level/transp.	4096 / 2048 / 0.1
	Field of View	15°
	displacement	up to $\pm 20mm / \pm 20^\circ$
	optimizer	Best Neighbor
	initial stepsize	8mm/8°
max. downscaling	0.001	

Figure 7.4: DRR image and settings of experiment #1

are averaged with a RMS term as well. The errors are in general very small, as expected. The largest component is in the y direction, which is basically the zoom of the DRR. The different behavior for specific pose parameters will become more clear in the successive evaluations, especially experiment #3. For this results, the errors are in general too small to make assumptions about the dependency between resulting pose, degrees of freedom and similarity measures. The only conclusion that I would like to point out for this experiment is about the number of outliers. Mutual Information and Correlation Ratio always find the right pose, whereas NCC and the neighborhood-based measures have some wrong results. I define a resulting pose as outlier, if the RMS value of the error transformation is larger than two. Therefore the convergence range of the first measures is higher. This can be traced back to the fact that the gradients in the used image are small and located very close to each other. As especially the gradient-based measures are assessing the alignment of those gradient structures, they become instable once the respective edges are too far apart. These measures should perform better when more coarse structures are present, so that the intensity gradients are more spread over the whole image. I did some quick evaluations with the DRR window/level setting altered to 4096/0, resulting in an image with brighter, more smooth structures. In that case all measures performed more or less equivalently.

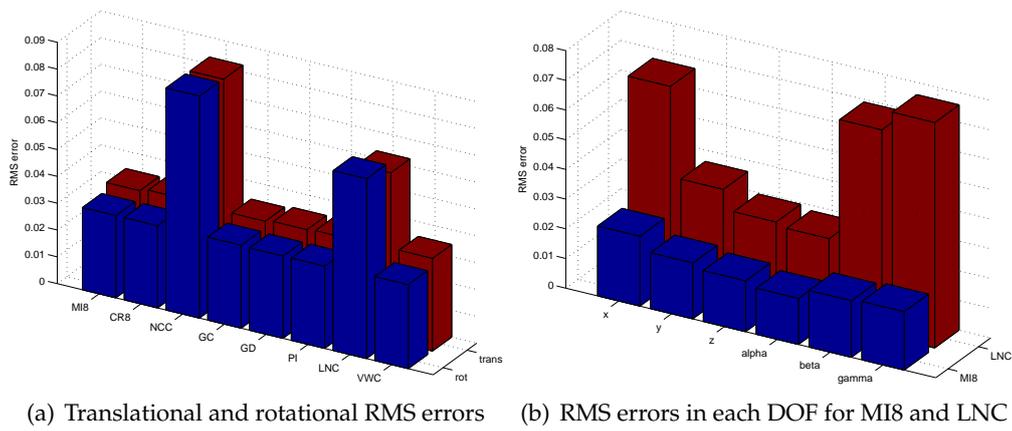
Now the registration is performed with an additional second image. It is perpendicular to the first one (gantry angle 90°), and the geometry between the two images is known, due to the known device dimensions of the linear accelerator. Expectedly, the stability increases enormously (figure 7.6). Registration errors are always below 0.1mm / 0.1°. The number of outliers decreased as well, only GC and VWC have one wrong result.

An additional test was done with a single image again, but using the Powell-Brent optimizer. The resulting accuracy was comparable to the previous run with the Best Neighbor optimizer, however the number of outliers for the neighborhood-based measures increased. Furthermore the absolute values of those outliers is higher, sometimes bailing out and reaching totally absurd values. This is probably due to the behavior of the Brent line minimizer, which estimates the next position on the current line based on a couple of previous evaluations, and is sensitive to disturbances in the parameter space.



Measure	MI8	CR8	NCC	GC	GD	PI	LNC	VWC
Outliers / 29 runs	0	0	5	5	3	4	5	4

Figure 7.5: RMS errors and outliers for experiment #1, mono image



Measure	MI8	CR8	NCC	GC	GD	PI	LNC	VWC
Outliers / 29 runs	0	0	0	1	0	0	0	1

Figure 7.6: RMS errors and outliers for experiment #1, stereo image

7.4.2 Experiments with not corresponding phantom data (#2)

The following experiments are done with a Rando body phantom, that we were allowed to use in collaboration with UPENN Hospital, Philadelphia. Before our second visit there (Aug. 3rd 2003), we had only a CT data set of the phantom without any markers and some X-Ray images of the same phantom. However, the latter images were done with something attached to the phantom that should simulate a female breast (visible in figure 7.2(a)). Note that the lobe of the lung in the upper left part of the X-Ray image is brighter than in the DRR (figure A.3). I used this fact as a chance to find out how well the registration works for partly different image content. For this combination of X-Ray and CT data, we do not have ground truth values. Therefore the focus on this set of optimizations is to assess the stability of the different measures, i.e. how regular the same registration pose is achieved from different starting estimates. Furthermore visual verification is done in order to make sure that corresponding structures in the images are correctly registered.

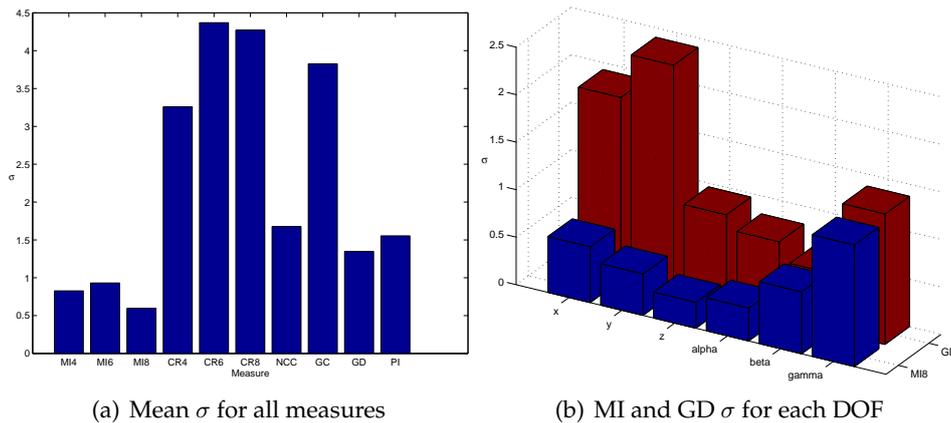


Figure 7.7: Experiment #2 with Stereo image, 0° and 90°

Figure 7.7 shows the results for registration from 33 randomly distributed initial estimates, conducted for each measure respectively. Mutual Information with 8 Bit histogram size seems to be the most stable measure, followed by Gradient Difference. However, they disagree on a final pose in the parameter space. Overlaying the pictures shows that the GD results are better aligned. Normalized Cross Correlation yields better results than expected, as the constraint of a linear dependency between those images is definitely not appropriate. However, looking closer at the results we can see that this measure "snaps" to one of two preferred orientations (see Appendix, figure A.4(c), two clusters are visible in the diagrams). The gradient-based measures finish around the same location (figure A.4(b), A.4(d)), though with different σ values. Unfortunately, the Local Normalized Correlation measures were not yet implemented at the time of this experiment. According to their properties (see section 3.2.4) they are supposed to give relatively good results when applied to not corresponding image data (especially the variance weighted method).

7.4.3 Experiment with corresponding phantom data (#3 - #6)

This is a set of experiments with ground-truth based data. X-Ray images and CT data were obtained with both markers and the one breast attached to the phantom.

Experiment #3, Mono

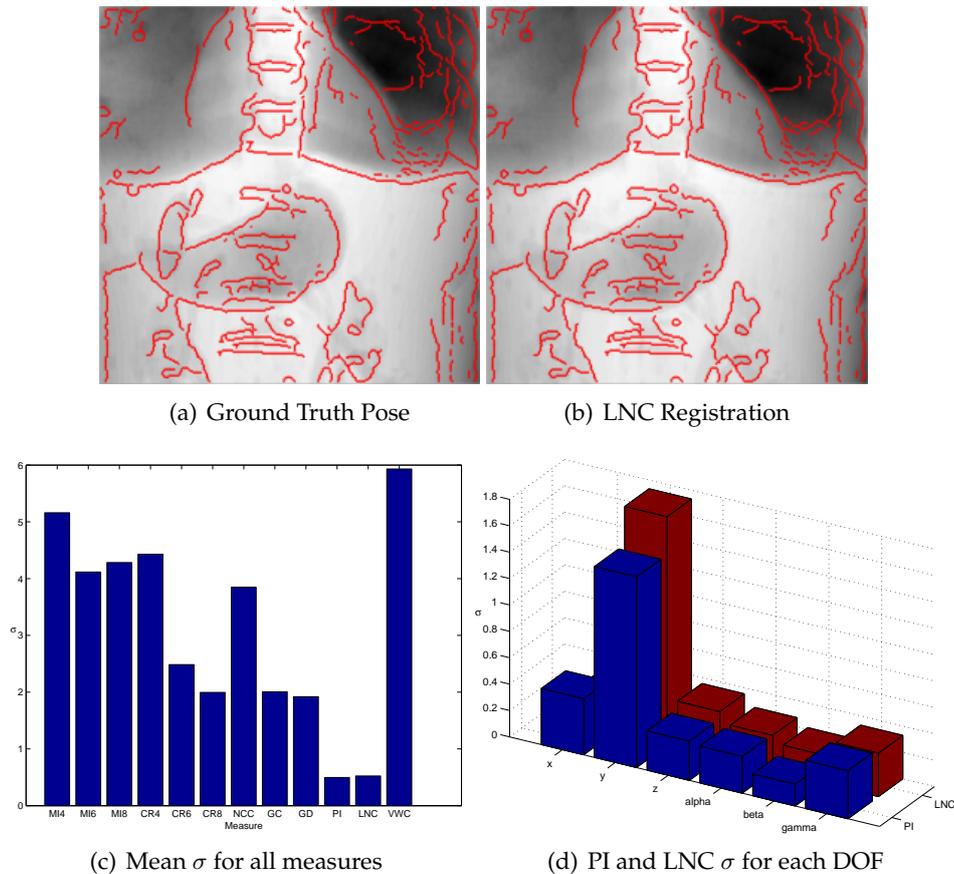
The registration was run starting from 33 different randomly displaced points around a preliminary ground truth pose, and with all different measures each. The mean distances from the resulting poses to the ground truth pose was e.g. for the LNC measure $dx = -2.6mm$, $dy = 12.1mm$, $dz = -2.8mm$, $d\alpha = -0.6^\circ$, $d\beta = -0.1^\circ$, $d\gamma = -0.1^\circ$. Therefore it is not applicable to express the registration result as distance to the ground truth, as this ground truth information does not seem to be precise enough. This is supported by the images showing edges of the X-Ray overlaid on the DRR (figure 7.8). The structures in the registration result (b) are better aligned than on the preliminary ground truth pose (a). Consequently, a good indicator of the stability of the registration is again the standard deviation σ of the displacement error with respect to the different starting positions. Diagram 7.8(c) shows the σ values with respect to the different measures (averaged over all degrees of freedom), diagram 7.8(d) displays σ for each degree of freedom in the results achieved with PI and LNC.

Pattern Intensity (PI) and Sum of Local Normalized Correlation (LNC) are the most stable measures, leading far ahead. Triggered from any starting pose within a $10mm/10^\circ$ hypercube in the parameter space, they find their way to a final registration pose with a mean standard deviation of less than $0.5mm/0.5^\circ$ in all DOFs except y . Important to mention is, that all measures using spatial information (GC, GD, PI, LNC and VWC), agree very precisely on the same best registration pose. Some of them have more outliers than others, though (see figures A.6(b) and A.6(d)). Mutual Information, Correlation ratio and Normalized Cross Correlation again disagree on a registration pose. The clouds of the resulting pose are unfortunately centered around different points, both for the translational and the rotational degrees of freedom (figures A.6(a) and A.6(c)).

Independent of the similarity measures, one can see the difference in the sensibility of the transformation parameters. Generally speaking, precision in out-of plane directions is always lower than in-plane ones. As for this experiment the detector angle is zero degrees, and we are therefore looking parallel to the y axis, moving along this direction results in zooming of the DRR. As the field of view of the source-detector arrangement is only 12 degrees, moving the volume some millimeters in y direction affects the DRR only slightly. This explains the high deviation in this direction, compared to all others. Concerning the rotational parameters, it can be seen that the registration is a bit more stable in the β -rotation than α and γ . Again this is obvious, as β , i.e. rotation around the y axis, is an in-plane transformation, unlike the others.

Experiment #4, Stereo 0° - 12°

Exactly the same kind of experiment is now conducted as biplanar registration with two images at 0 and 12° gantry angles.

Figure 7.8: Rando Phantom experiments with single image, 0°

Normalized Cross Correlation and Variance-Weighted Sum of Local Normalized Correlation do not converge most of the time. The rest of the neighborhood information-based measures behaves very well again (GC, GD, PI and LNC), Pattern Intensity is leading. In general, the precision increases slightly due to the use of a second, 12 degrees rotated image. Especially the y direction is now registered better, as it is not a pure out-of-plane translation in the 12° image.

Experiment #5, Stereo 0° - 90°

Using the same settings again, I did 33 runs with stereo images, detector positions 0° and 90° now. The standard deviation of the initial pose lies within 5 and 7 mm / $^\circ$ for each DOF. Average computation time was about three minutes. Obviously two perpendicular images are the richest information base available for 2D-3D registration, as they, if used in conjunction, contain some three-dimensional information. Thus biplanar registration with two perpendicular images is sometimes referred to as 2 $\frac{1}{2}$ D-3D registration as well.

Accordingly, the results are very accurate. Even for the measures that turned out to perform poorly in the previous experiments, the standard deviation is in average below $0.8\text{mm}/^\circ$.

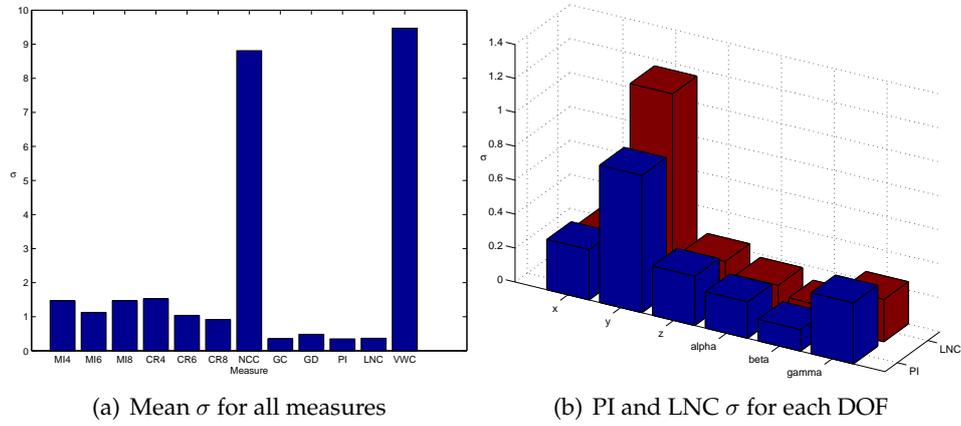


Figure 7.9: Rando Phantom experiments with stereo images, 0° and 12°

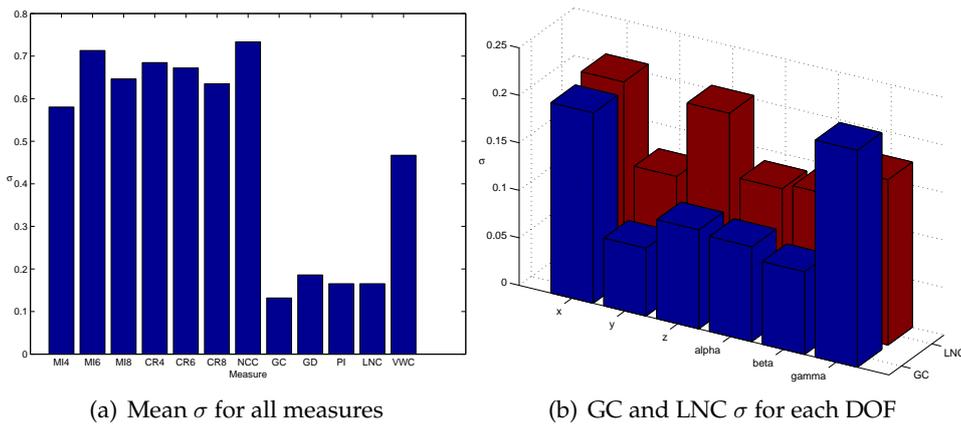


Figure 7.10: Rando Phantom experiments with stereo images, 0° and 90°

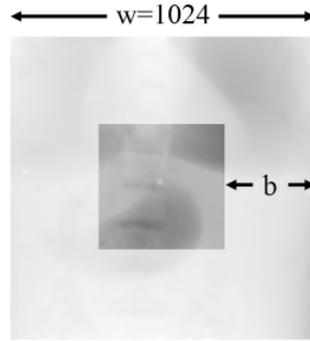


Figure 7.11: Region of interest applied to X-Ray image

The best measures in this case, Gradient Correlation and Local Normalized Correlation, achieved σ values of always less than $0.25mm/^\circ$. This basically indicates that one final registration pose is exactly retrieved each time, no matter where the initial estimation was. This pose, expressed as an error transformation relative to the ground truth estimate, has the following value:

$$x = -2.4mm, y = -2.0mm, z = -2.7mm, \alpha = -0.8^\circ, \beta = -0.3^\circ, \gamma = 0.1^\circ$$

This is within the error range of our ground truth estimation, and therefore can be considered as correct registration. The Variance-Weighted Local Normalized Correlation (VWC) measure produced only one outlier in this set of registrations (visible in figure A.8). If we remove it, the mean σ value becomes 0.12, surpassing all other measures.

Experiment #6, Region of Interest

Next, I want to assess the effect of using only parts of the images for the registration. Therefore I systematically defined three regions of interest and used each of them for a set of optimizations, just like in the previous sections. The ROIs are defined as squares centered on the original X-Ray image files, which have a resolution of 1024×1024 (figure 7.11). I used three sizes, omitting a border of $b = 100, 200$ and 300 pixels on each side of the square. The fraction of the number of pixels used for registration is then 65%, 37% and 17%, according to

$$\frac{(w - 2b)^2}{w^2}$$

Figure 7.12 shows the same σ diagrams as usual, though this time the results are denoted for once the full image content, and each of the ROIs with 100, 200 and 300 pixels border. It is clearly visible that the stability only decreases rapidly with use of ROIs for the measures that did not register stable anyway (especially NCC). The measure least sensitive to shrinking the used image information is Local Normalized Correlation, therefore the detailed σ values are drawn in figure 7.12(b). The change in the σ values for different degrees of freedom does not seem to follow any scheme. The assumption that I conclude for the use of ROIs is, that the results will always be highly dependent on the underlying image structures.

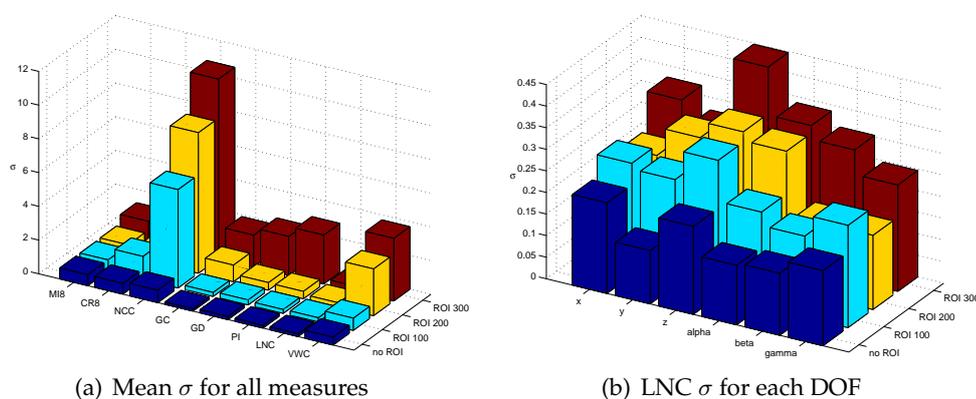


Figure 7.12: Results for biplanar registration with different region of interests

7.5 Parameter Space Evaluation

To find out more about the behavior and stability of the registration, one can compute and display similarity measure values while altering the transformation parameters. The best way to get a feeling about the behavior in the parameter space in general, is to use a good visualization in order to display the measure values. Unfortunately one can not visualize six degrees of freedom at the same time in an intuitive way. Therefore I created many plots of all similarity measures applied on various images, while changing two degrees of freedom. I would like to present some of the resulting diagrams, as they tell a lot about the properties of the registration problem for specific data & measures. All colored two-dimensional plots can be found in the Appendix.

The computations described below are done on the Rando Phantom data set with a single X-Ray image at 0° gantry angle, which is the same configuration as in experiment #3, section 7.4.3. The plots in figure A.1 show all similarity measures with the in-plane translations x and z altered $\pm 20mm$ each. Changing the DRR in these degrees of freedom usually results in the most smooth and independent change of the respective measure. This is therefore the best way to point out the characteristic behavior of the different similarity measures. Mutual Information is reasonably smooth and has a significant maximum in the center, whereas the experiments showed that this does not correspond with the best alignment. Correlation Ratio and Normalized Cross Correlation both have the same characteristics - a very smooth and monotonously curved surface, with NCC having a slightly different bending. I observed a similar shape with any kind of images and pose parameters to be modified, which can be explained with the common nature of these two measures (see section 3.3.3). However, the drawback they have in common is that the maximum is not defined very distinctive, as the gradient is very small in a wide area around it. Gradient Correlation, Gradient Difference and Pattern Intensity all produce a similar plot with a very well-defined maximum, yet Gradient Correlation having some wrinkles for large displacements. The Local Normalized Correlation measure generates an even sharper peak at the correct

registration pose. As I already confirmed with experiments, Variance-Weighted Local Normalized Correlation bails out for larger displacements, in this case rising again in the negative x axis.

In order to show the dependency between specific pose parameters, I included plots of other DOF pairs altered, using the LNC measure. They are presented in figure A.2.

Altering the two out-of-plane rotations α and γ simultaneously results in a similar behavior than altering x and z (see LNC plot in figure A.1). Obviously those parameters are very independent of each other as well.

In the next plot β and y are altered together. For the used single X-Ray image with 0° gantry angle, the y axis is parallel to the viewing direction and thus determines the zoom into the image. In order to affect the similarity measure significantly, the change in the y direction has to be much larger than for all other pose parameters (see also experiment #3, section 7.4.3). I set it to $\pm 100mm$ for this plot, while β is altered $\pm 20^\circ$. The maximum is a sharp ridge, exactly parallel to the y axis. This shows first that the β angle has a very precise optimum, which is no surprise for the in-plane rotation. Secondly, β and y are very independent of each other as well.

The x and γ parameters have a very high dependency, as the maximum falls off on a ridge diagonal to the axes. If a majority of the structures contained in the CT data set are in front or behind the γ axis, a rotation around it has the effect of moving these structures along the x axis, which explains this dependency.

A similar behavior can be observed for the z and α parameters in the last plot. The ridge is again diagonal to the axes, being at the same time a bit wrinkly, though. Probably this is because the spine structures are moved vertically by the y and β parameters, causing some repeated better alignment at wrong poses.

Concluding for the study of figure A.2, I would like to point out that the behavior of similarity measures in the parameter space is always highly dependant on the used images as well. The effects that I mentioned above are likely to be found to a certain extend in most other registrations, as they are mainly caused by the general affect that the parameters of a rigid-body transformation have on a 2D projection image. On the other hand, the underlying image data is surely defining a lot of other dependencies & artifacts in the parameter space. Some of these can cause major trouble for trying to apply an automatic optimization. Two fellow students at my university were working on a project which involved 2D-3D registration of CT and fluoroscopy images [12, 20]. They encountered problems as the images consisted mainly out of spine structures. The vertebrae caused redundancy in a way that many optima in the parameter space existed, each denoting some alignment of vertebraes in the DRR and fluoroscopy image. Many other problems like this can be found, depending on the images that are to be registered.

8 Conclusion

8.1 Result

Similarity Measures The experiments in chapter 7 show that it is advisable to use various similarity measures for a 2D-3D registration problem. The information-theoretic measures Mutual Information and Correlation Ratio are resistant to local optima in large displacements, and thus have a very large convergence range (section 7.4.1). In addition Mutual Information seems to perform better when images with partly different content are used (section 7.4.2). The neighborhood-based measures are able to reach the correct alignment with the highest precision, once the edges in the respective images are sufficiently close to each other. Variance-Weighted Sum of Local Normalized Correlation (VWC) creates the sharpest peak for the optimal alignment of two images, being at the same time the most sensitive if the displacement is too large.

Resulting from those facts, I suggest that a hierarchical approach should be used not only in terms of e.g. image resolution & smoothing, but also for different similarity measures. For instance, a registration could start with Mutual Information, switch to some neighborhood-based measure after a defined time (GC, GD, PI or LNC), and eventually use the VWC measure. This could assure that both the largest convergence range and the highest registration accuracy is achieved.

Optimization Algorithms Given the quality of our DRR and X-Ray images, it turned out that there is no need to use stochastic optimization procedures in order to avoid wrong results. Both the Best Neighbor and Powell-Brent optimizer proved to work very reliably. In fact, the registration results of those optimizers were always quite similar, causing me to discard further evaluations in that direction. Only if the cost function is badly conditioned, the Powell-Brent method is slightly more sensible in terms of outliers. The Gradient Descent algorithm has the potential to register very rapidly, however does its behavior and especially the learning rate depend too much on the underlying images and on the used similarity measure. Its routinely use would therefore involve some interaction and manual verification of the correct behavior.

Application Prototype The main contribution of this work is definitely to present a working application prototype for intensity-based 2D-3D registration using CT and X-Ray data. The project was targeted towards the use in Radiation Therapy, but we are confident that it can be applied successfully to any similar registration problem. The RTReg program is a test platform, featuring the use of 8 different similarity measures and 3 optimization algorithms. The system can perform synthetic experiments or registration of real data. Either a single X-Ray image or two images with known geometry between the images can

be used. The latter case is often referred to as $2.5D - 3D$ registration, as the biplanar X-Ray images already contain some spatial information.

The registration accuracy turned out to be extraordinary. For validation, we acquired ground truth information for a phantom data set with metal markers. Our registration, using a specific selection of similarity measures, always ended up within the error range of the Ground Truth pose, except in the out-of-plane translation for mono images. Besides, the statistical properties of many experiments on the phantom data show that the registration started from arbitrary locations results at the same pose with submillimeter precision. This does of course not guarantee that the alignment of the physical structures of a patient within the CT data will have the same precision. It only predicates that the system is very stable, as it registers the same data very determined to the same pose. Almost not to mention are the synthetic experiments, which demonstrate perfect registration (also with sub-millimeter precision in each DOF) for all used data sets.

The registration is executed in 1-4 minutes, mostly depending on the volume resolution and the optimization parameters. There are already faster systems available, especially for applications in realtime like fluoroscopic guidance, but not with our accuracy. The most time-consuming process in our application is the DRR rendering with a 3D-Texture algorithm. This rendering however contributes a lot to the achieved registration quality, as it creates very realistic DRR images without artifacts that come with other faster rendering techniques. The whole system runs on a high-end consumer PC, which is very inexpensive compared to for instance a graphics workstation.

With our results, we demonstrated the feasibility of fully automatic 2D-3D registration procedures in order to solve the patient positioning problem in radiation therapy.

8.2 Problems/Discussion

First of all, our registration works very well with the kind of data that we obtained from the UPenn hospital. The X-Ray sensing device there was a solid detector, that was added after installation of the system. Linear accelerators, that obtain portal images by using integrated image intensifying sensors, produce images of a much poorer quality. On the other hand, it is foreseeable that solid X-Ray detectors will gain importance in the future. Right now they are often not considered for purchase due to their high price.

Even if our results are very promising, many things are yet to be evaluated, especially from a clinical point of view. We do not have information about the registration accuracy on specific objects of a real patient, like the brain, the spine or the abdomen. For each of those objects, a certain amount of non-rigidity has to be accounted for, as e.g. tissue or spine vertebrae can be shifted with respect to the CT and the portal images.

8.3 Future Work

The work on this project is very likely to be carried forward. Both detailed algorithmic aspects and general application issues can be further examined and improved:

- The main bottleneck in terms of execution speed is the DRR algorithm, but there is potential for further development. The amazing technological level of consumer graphics cards is likely to advance rapidly, making even things like 3D texture rendering more efficient. This involves the possibility to use both larger volume sizes and resolution of the DRR images for registration. It seems to be reasonable to work with DRR resolutions up to 1024x1024, generated using a full CT volume content consisting for instance of 512x512 slices - that is the actual size of X-Ray images generated by digital detectors. Processing of the full amount of information made available by the imaging modalities thus would be guaranteed.
- It would probably make sense to include some simple segmentation algorithm into the application. Depending on the imaged object, the used CT data could be reduced e.g. to bony structures which are more likely to stay rigid between the acquirement of the CT scan and the portal images. Using some strategy to define a region of interest, both stability and speed of the registration could be improved (see section 5.3).
- The parameters of the used optimization algorithm and their influence on the registration performance should be further examined. The Powell-Brent optimizer should be trimmed to use adaptive termination constraints for the line minimizer (section 4.4) in order to speed up the computation. The influence of different images on what the best learning rate for the Gradient Descent optimizer (section 4.3) is, has to be evaluated. Besides, the already implemented multiresolution strategy (4.5) should be used intensively in test runs.
Furthermore, among the manifold available non-linear optimization algorithms, there are other potential candidates that would work well in our application. This includes for instance *Surrogate* methods or the *Hooke-Jeeves Pattern Search* algorithm, which are successfully applied by [5].
- In my opinion it definitely makes sense to study and apply the approximation using 2D transformations (section 5.2) more. Most importantly, the error introduced by approximating perspective projections with a parallel one should be assessed. This would indicate the precision that could be achieved using this simplification, and up to which level on a multi-resolution strategy it could be used.
- No matter how many portal images from different positions are acquired by the linear accelerator, the inter-image geometry is always given precisely by the knowledge about the gantry rotation around the isocenter. Therefore it is possible to use more than two images at once for the registration. Supposedly it would increase the overall precision and stability to use more images. With a whole set of images, this can furthermore lead to reconstruction approaches like [28]. The radiation exposure of the patient is probably neglectible when the linear accelerator is in imaging mode, with respect to the radiation energies used for the treatment.
This fact could even be used to change the whole treatment procedure. The registration can be used not only for the initial patient positioning, but to successively correct for positioning errors during the treatment. The linear accelerator has to rotate the gantry with defined steps and angles anyway, in order to spare healthy tissue from being strained by the radiation. At each gantry repositioning, a newly acquired portal

image could be used in conjunction with the previous ones to adapt the patient position, in order to hit the target spot with the highest possible precision.

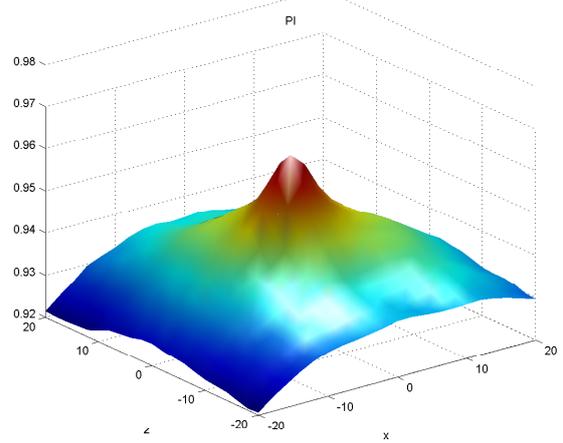
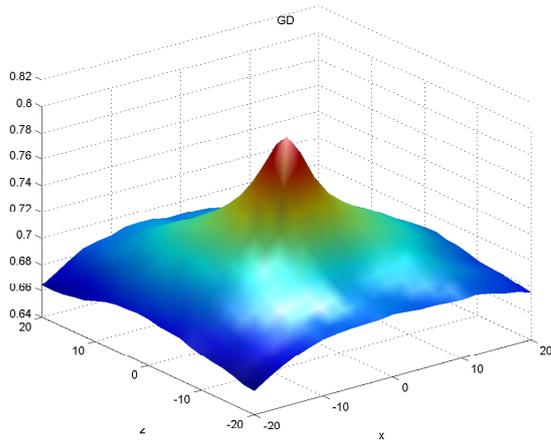
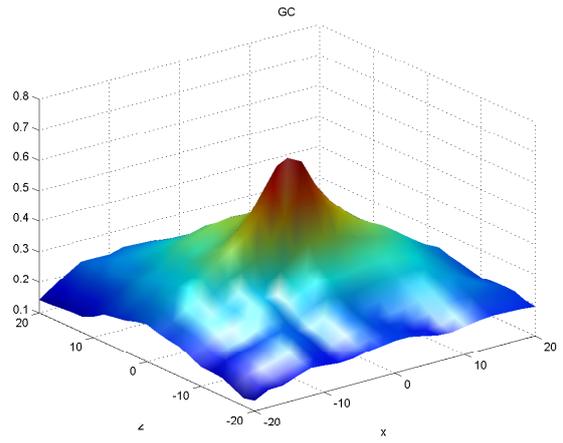
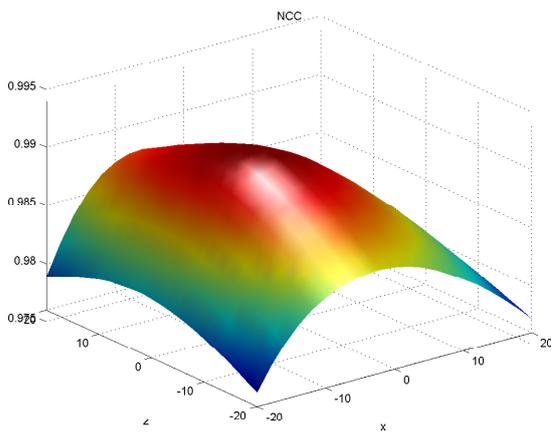
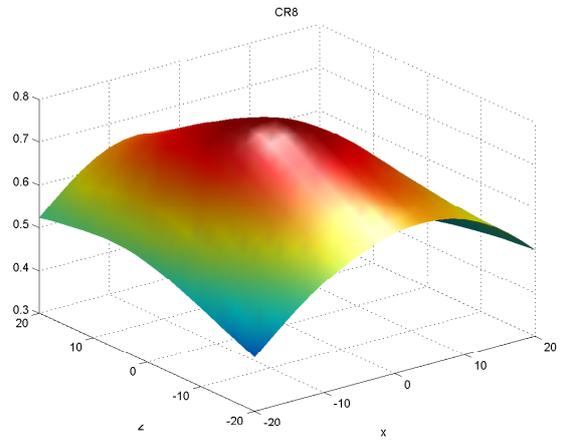
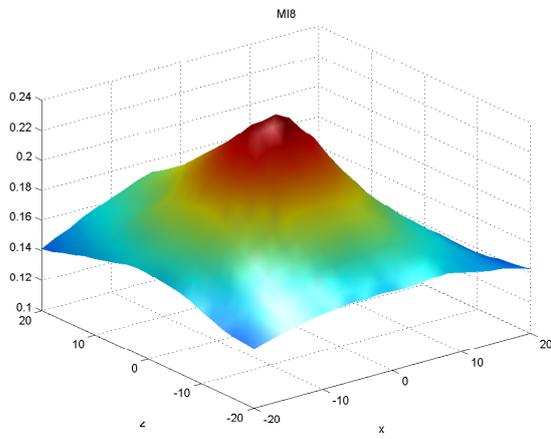
- Clinical trials are necessary to proof the claim that this application can be used for supporting or even replacing existing patient positioning methods. Ground Truth based registration has to be conducted on various target objects in order to make sure that the overall precision is sufficient for regular clinical use. Among other things, it has to be studied if the non-rigidity of the specific target area is always neglectible. This sometimes depends on the right kind of patient immobilization as well [3].

A Detailed Experimental Results

On the following pages is a set of interesting diagrams, images and experimental data.

The figures [A.1](#) (spans over two pages) and [A.2](#) contain two-dimensional plots of two pose parameters against a specific similarity measure. They are explained and interpreted in section [7.5](#).

The subsequent pages contain detailed information on the experiments #2 - #6. In [A.3](#) and [A.5](#) the used images and settings for a series of experiments are shown. Figures [A.4](#), [A.6](#), [A.7](#), [A.8](#), [A.9](#), [A.10](#) and [A.11](#) depict the distribution of the respective final poses in the parameter space and their standard deviations, they are referenced in the Experiments section [7.4](#). Those distributions are split up into the translational (x, y, z) and rotational (α, β, γ) degrees of freedom. Each two of those plots are drawn for MI, CR, NCC (left plots) and GC, GD, PI, LNC, VWC (right plots). The initial estimates are displayed on all diagrams as yellow points.



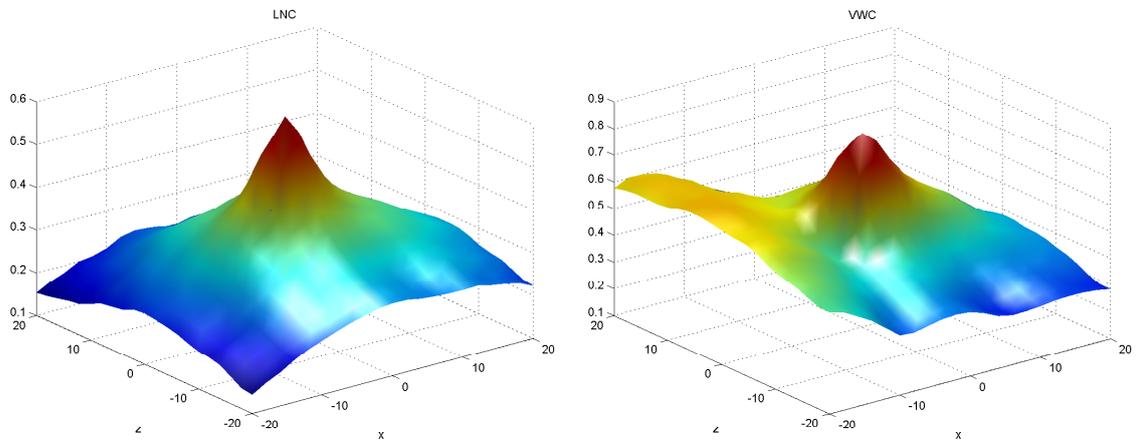


Figure A.1: Similarity Measure plot for altering the in-plane translations

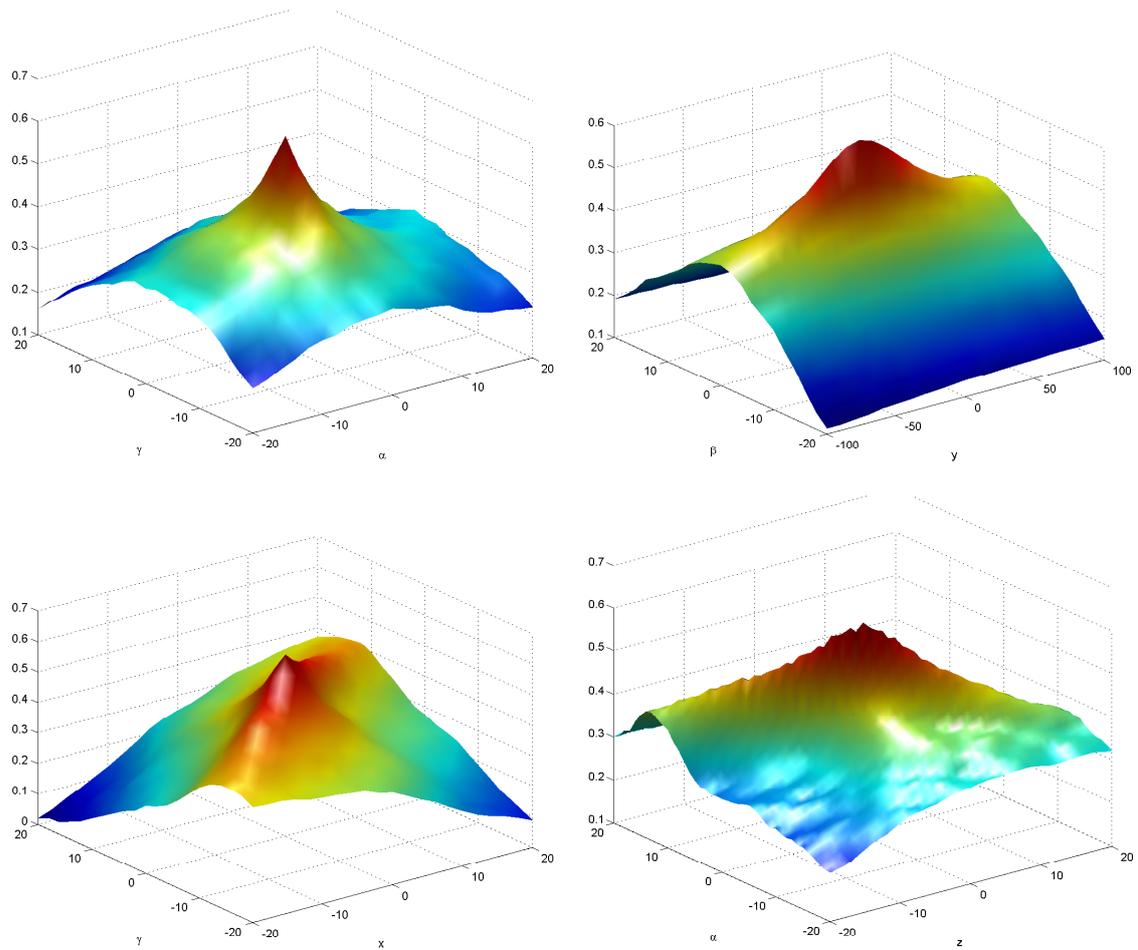
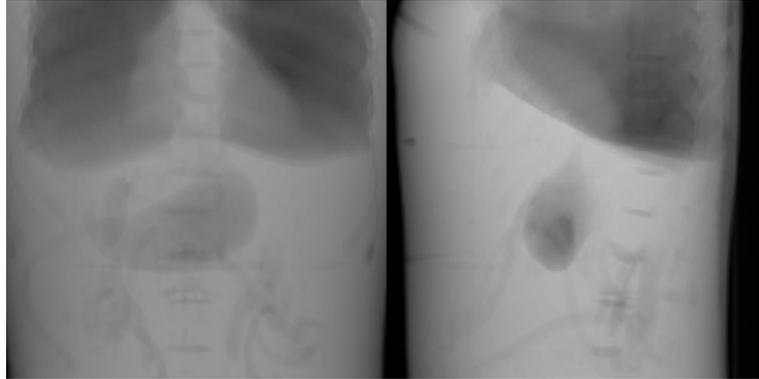
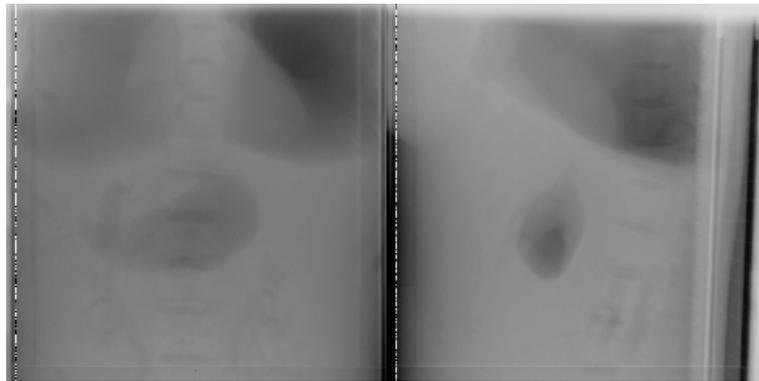


Figure A.2: Plots of the LNC measure with different pairs of DOFs altered



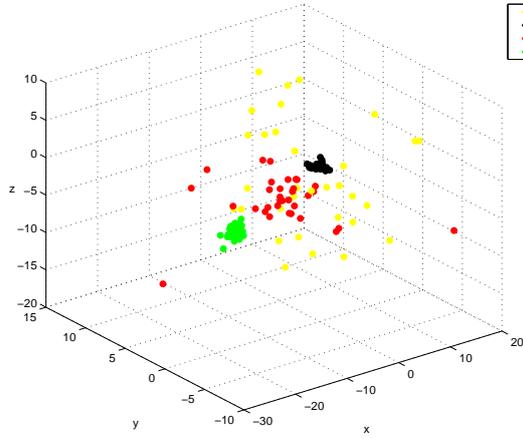
(a) DRR images, 0° and 90°



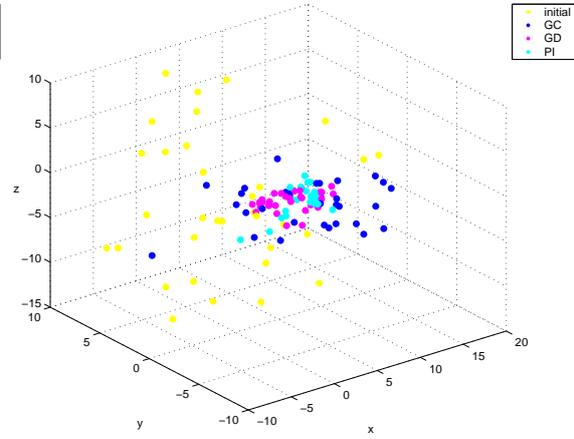
(b) X-Ray images, 0° and 90°

Setting	Value
data set	Rando Phantom #1
used volume size	256 x 256 x 185
voxel spacing	1.88mm x 1.88mm x 3.0mm
start pose	[2.45, 270, -578.5, 0, 0, 0]
DRR window/level/transp.	920 / 2400 / 0.01
Field of View	15°
Detector	Stereo, 0° and 90°
displacement	up to $\pm 10\text{mm} / \pm 10^\circ$
optimizer	Best Neighbor
initial stepsize	8mm/ 8°
max. downscaling	0.001

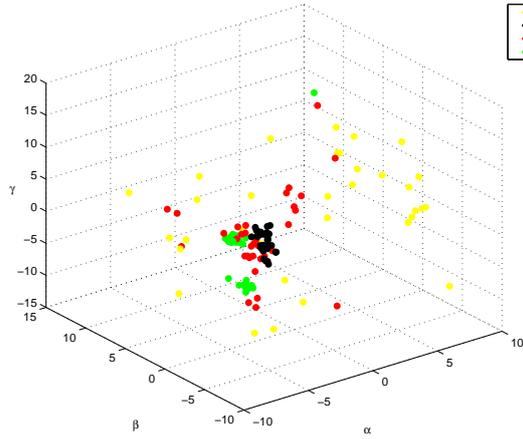
Figure A.3: Settings and used images for experiment #2



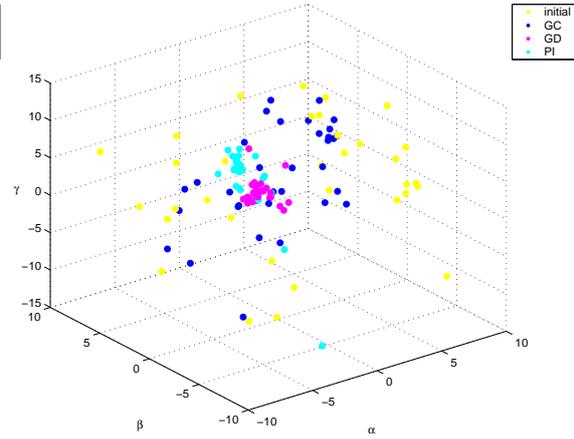
(a) Translational DOFs, first part of measures



(b) Translational DOFs, second part of measures



(c) Rotational DOFs, first part of measures

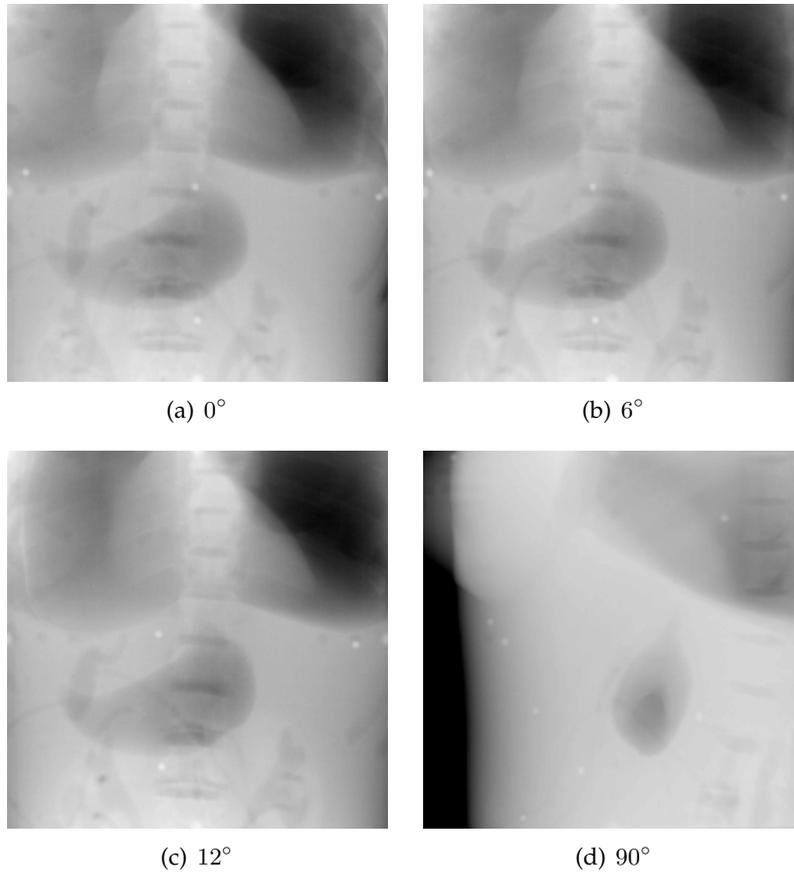


(d) Rotational DOFs, second part of measures

	x	y	z	α	β	γ	mean
MI4	1.1818	0.3878	0.3667	0.3145	0.9566	1.7523	0.8266
MI6	0.9262	0.5585	0.4961	0.3316	0.9259	2.3481	0.9310
MI8	0.5865	0.4350	0.2714	0.3451	0.6499	1.2786	0.5944
CR4	4.6364	3.3421	2.2124	1.3324	2.4035	5.6165	3.2572
CR6	5.9632	4.3600	3.4035	1.5486	5.0085	5.9328	4.3694
CR8	7.3566	2.8715	3.5817	1.3724	4.3677	6.0667	4.2694
NCC	1.3492	1.0023	0.9947	1.2395	1.0167	4.4558	1.6764
GC	4.9412	3.8834	3.0450	4.8561	1.9214	4.3058	3.8255
GD	1.9253	2.3997	0.9600	0.8101	0.6091	1.3721	1.3460
PI	2.1622	1.0269	1.2935	0.8311	1.0053	3.0029	1.5536

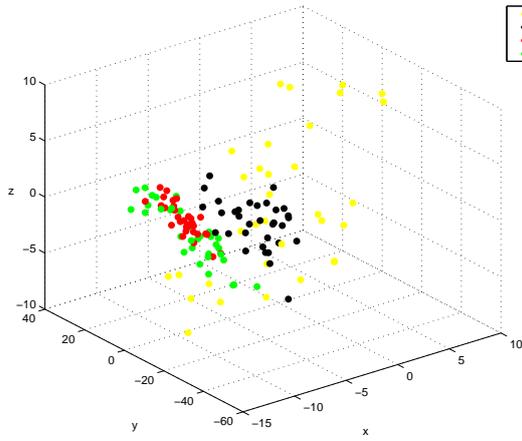
(e) Table of standard deviations

Figure A.4: Distribution in parameter space and σ values for experiment #2

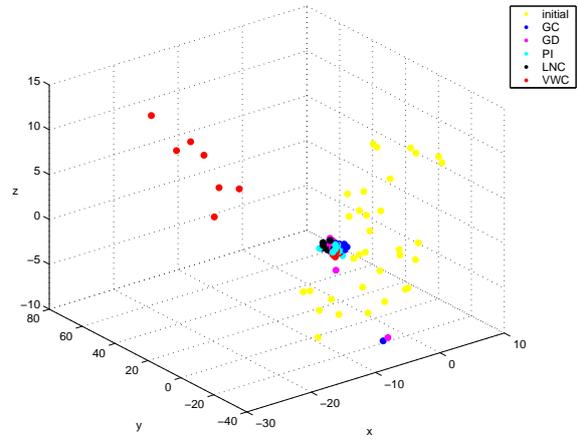


Setting	Value
data set	Rando Phantom #2
used volume size	256 x 256 x 216
voxel spacing	1.88 x 1.88 x 3 mm
start pose	ground truth
DRR window/level/transp.	900 / 2900 / 0.01
X-Ray window/level	124 / 130
Field of View	12°
displacement	up to $\pm 10mm / \pm 10^\circ$
optimizer	Best Neighbor
initial stepsize	8mm/8°
max. downscaling	0.001

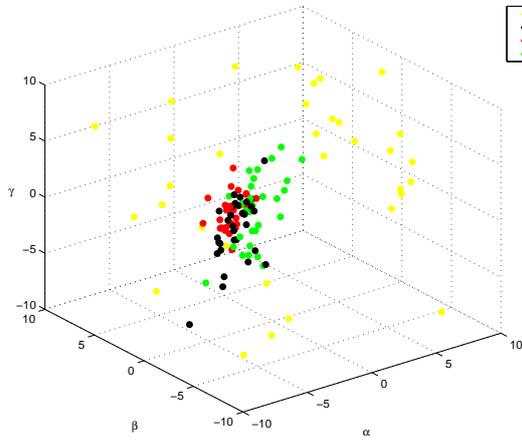
Figure A.5: Settings and used X-Ray images for experiments #3 - #6



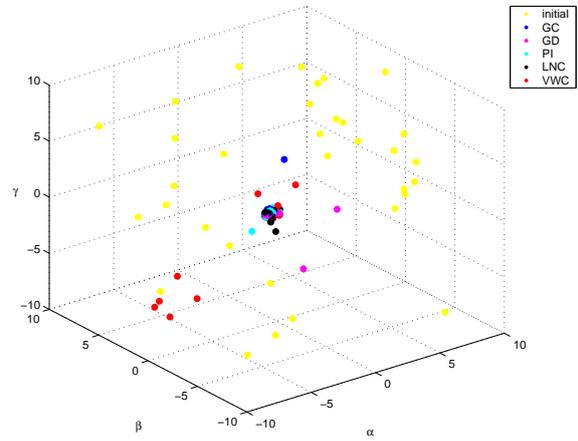
(a) Translational DOFs, first part of measures



(b) Translational DOFs, second part of measures



(c) Rotational DOFs, first part of measures

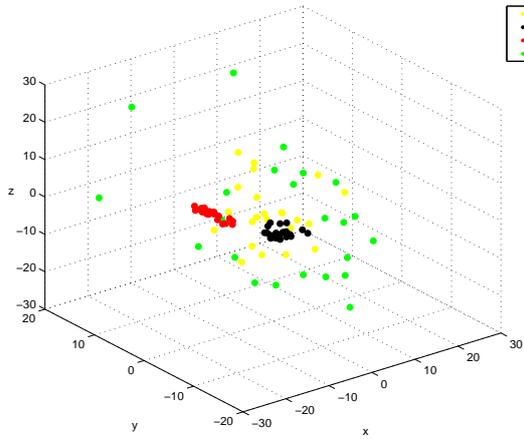


(d) Rotational DOFs, second part of measures

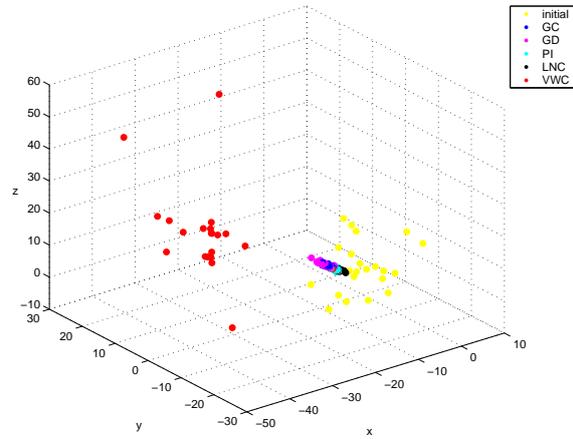
σ	x	y	z	α	β	γ	mean
MI4	3.2081	16.8850	3.3212	2.5930	1.1774	3.3992	5.0973
MI6	1.4548	15.9270	2.3985	1.7798	1.1183	1.8413	4.0866
MI8	1.6321	16.8986	2.1787	1.8231	1.2688	1.8517	4.2755
CR4	2.9569	13.6513	2.6940	2.7586	0.9166	3.4300	4.4012
CR6	0.7968	10.1979	1.0982	1.0618	0.6366	1.2080	2.4999
CR8	0.8263	7.6663	0.9281	0.9008	0.4719	1.1653	1.9931
NCC	1.6113	14.6071	1.8968	1.7175	0.8101	2.2125	3.8092
GC	0.9340	8.0687	0.8625	0.8997	0.8832	0.2174	1.9776
GD	0.4790	7.0843	1.1198	1.2229	0.3899	1.0682	1.8940
PI	0.4180	1.4848	0.2949	0.2769	0.1625	0.3610	0.4997
LNC	0.2313	1.7229	0.3535	0.2703	0.2119	0.3358	0.5210
VWC	5.7846	18.8061	4.5100	2.1550	1.2279	2.6524	5.8560

(e) Table of standard deviations

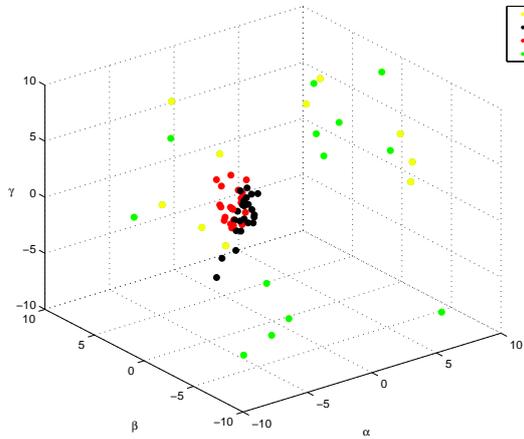
Figure A.6: Experiment #3 (single image)



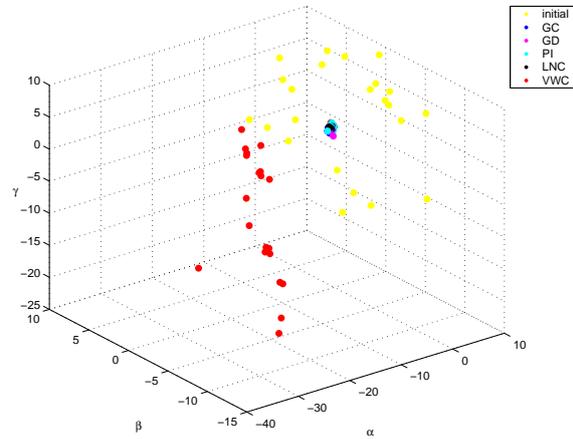
(a) Translational DOFs, first part of measures



(b) Translational DOFs, second part of measures



(c) Rotational DOFs, first part of measures

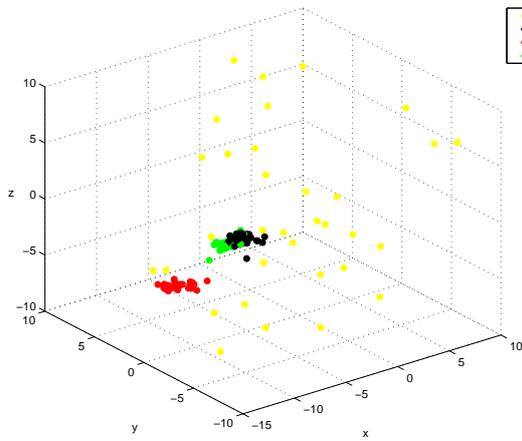


(d) Rotational DOFs, second part of measures

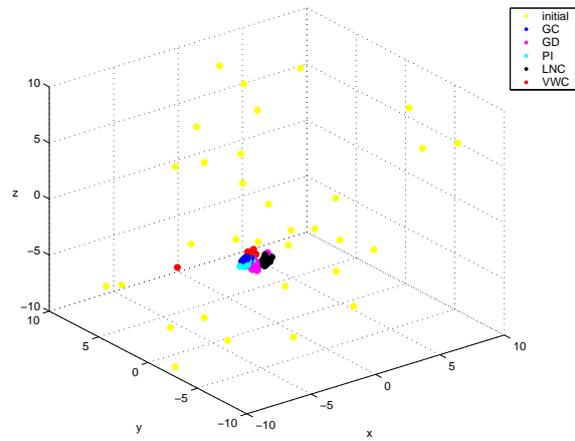
σ	x	y	z	α	β	γ	mean
MI4	1.2665	2.0836	1.5596	1.2872	0.8643	1.7476	1.4681
MI6	0.6034	1.6397	1.6364	1.2686	0.7371	0.8846	1.1283
MI8	0.9907	2.2460	1.9105	1.4809	1.0599	1.1500	1.4730
CR4	1.8030	1.7943	1.3432	1.3151	0.6804	2.2317	1.5280
CR6	0.7932	2.0943	0.7737	0.8670	0.5467	1.1579	1.0388
CR8	0.7141	1.7999	0.6352	0.7187	0.5335	1.0952	0.9161
NCC	14.3676	8.1203	12.0076	5.9504	5.6735	6.7413	8.8101
GC	0.3311	0.7485	0.3274	0.2200	0.1514	0.3333	0.3520
GD	0.4025	1.6057	0.2486	0.1680	0.1436	0.3311	0.4833
PI	0.2958	0.8097	0.2898	0.2142	0.1261	0.3585	0.3490
LNC	0.2531	1.1598	0.2469	0.1815	0.1222	0.2493	0.3688
VWC	11.7310	11.5754	13.7084	7.7952	4.6251	7.3952	9.4717

(e) Table of standard deviations

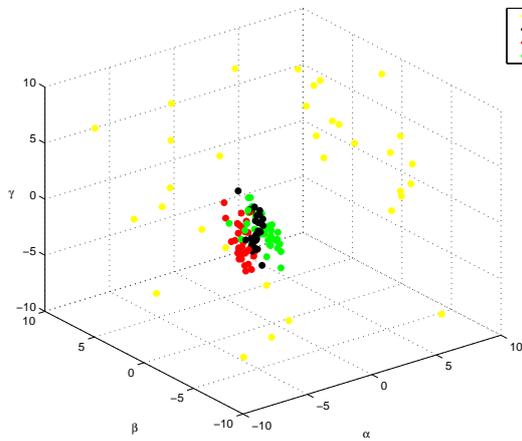
Figure A.7: Experiment #4 (stereo, 0° and 12°)



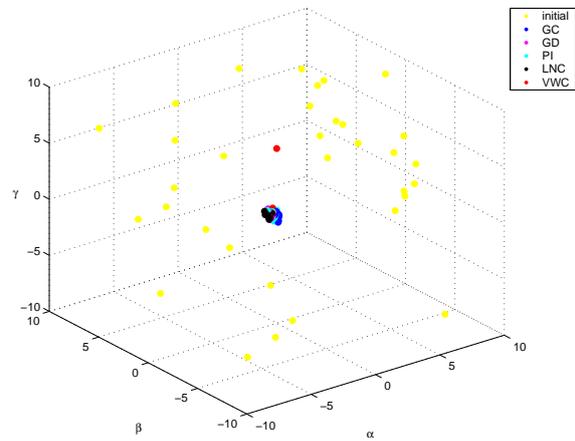
(a) Translational DOFs, first part of measures



(b) Translational DOFs, second part of measures



(c) Rotational DOFs, first part of measures

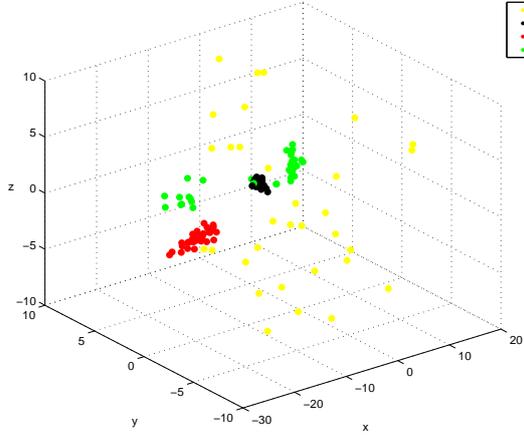


(d) Rotational DOFs, second part of measures

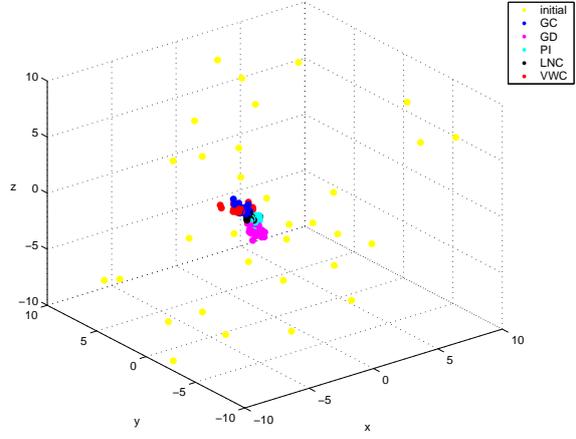
σ	x	y	z	α	β	γ	mean
MI4	0.9077	0.2650	0.3485	0.2874	0.6614	1.0126	0.5804
MI6	1.0263	0.5441	0.6981	0.5151	0.5519	0.9427	0.7130
MI8	0.9344	0.2575	0.5541	0.4388	0.6495	1.0437	0.6463
CR4	1.0038	0.2539	0.4972	0.3529	0.7863	1.2132	0.6846
CR6	1.2906	0.1543	0.3766	0.2589	0.5348	1.4179	0.6722
CR8	1.0958	0.1381	0.3527	0.2992	0.5731	1.3504	0.6349
NCC	0.8738	0.3195	0.5024	0.4196	1.2300	1.0546	0.7333
GC	0.2004	0.0713	0.1041	0.0999	0.0869	0.2282	0.1318
GD	0.2341	0.1814	0.2578	0.1325	0.1354	0.1743	0.1859
PI	0.2373	0.0998	0.1244	0.1213	0.1470	0.2619	0.1653
LNC	0.2097	0.1234	0.2037	0.1378	0.1434	0.1740	0.1653
VWC	0.9451	0.1187	0.1498	0.4369	0.4822	0.6696	0.4671

(e) Table of standard deviations

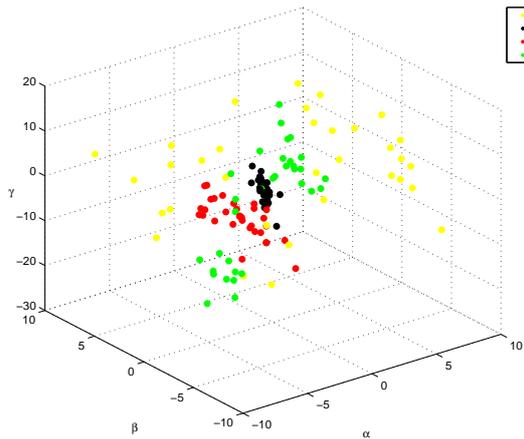
Figure A.8: Experiment #5 (stereo, 0° and 90°)



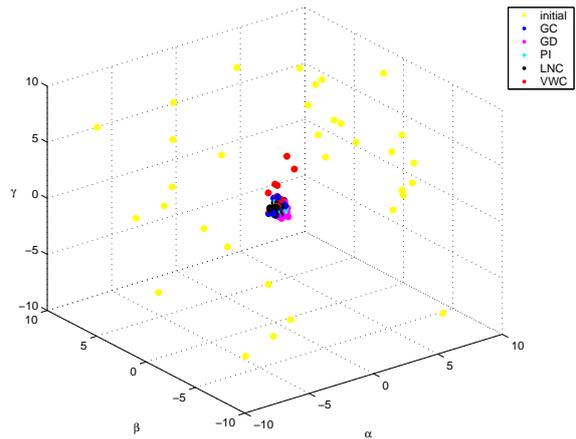
(a) Translational DOFs, first part of measures



(b) Translational DOFs, second part of measures



(c) Rotational DOFs, first part of measures

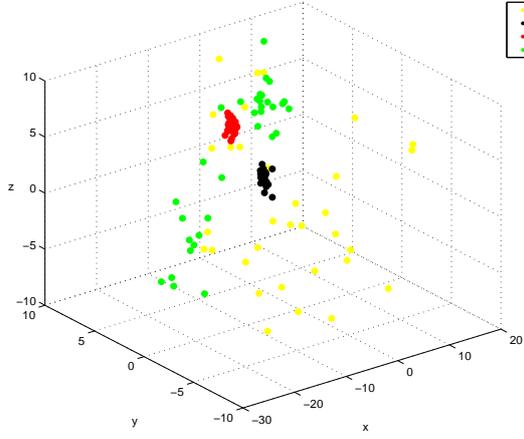


(d) Rotational DOFs, second part of measures

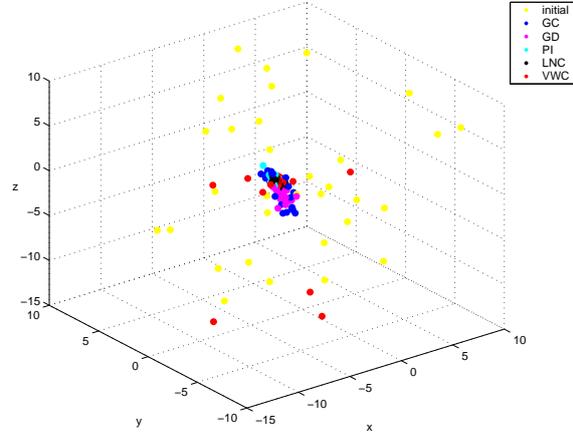
σ	x	y	z	α	β	γ	mean
MI8	0.8166	0.3089	0.2823	0.3944	0.7879	1.3530	0.6572
CR8	1.0594	0.9564	1.0708	1.1657	1.9190	2.0104	1.3636
NCC	12.6280	1.7623	1.3265	0.9307	3.2584	15.0435	5.8249
GC	0.2668	0.1513	0.3397	0.2482	0.1446	0.2898	0.2401
GD	0.3194	0.3101	0.3806	0.2197	0.1849	0.3347	0.2916
PI	0.2333	0.1863	0.3021	0.1822	0.2454	0.2400	0.2316
LNC	0.2597	0.2459	0.3150	0.2193	0.1876	0.2370	0.2441
VWC	0.7428	1.2658	0.7285	0.5023	0.3571	0.7455	0.7237

(e) Table of standard deviations

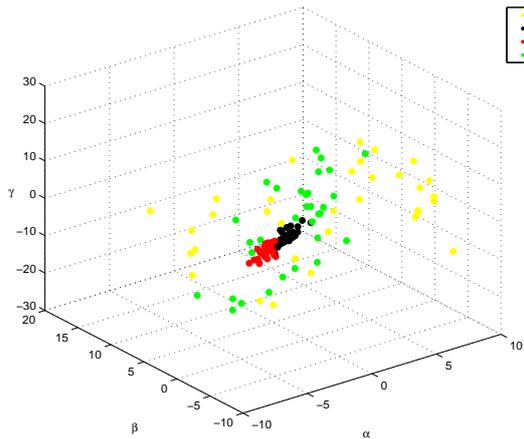
Figure A.9: Experiment #6 with ROI border 100



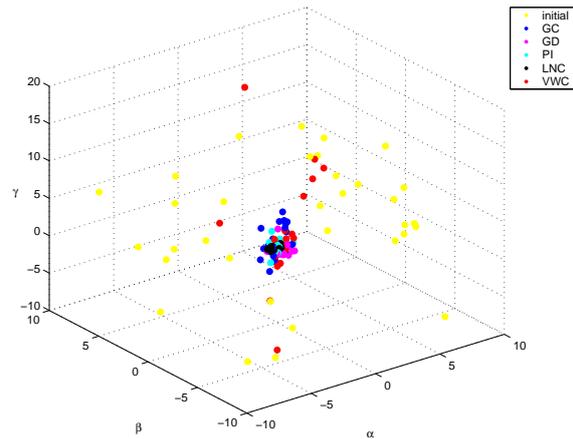
(a) Translational DOFs, first part of measures



(b) Translational DOFs, second part of measures



(c) Rotational DOFs, first part of measures

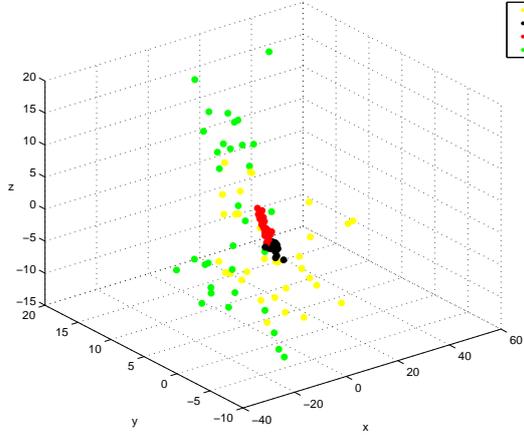


(d) Rotational DOFs, second part of measures

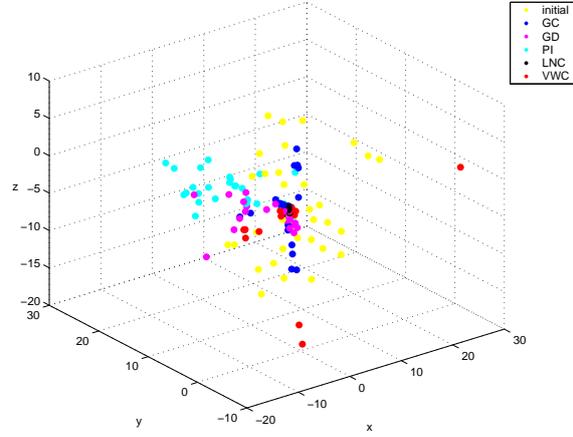
σ	x	y	z	α	β	γ	mean
MI8	0.7869	0.4004	0.7691	0.7835	0.4437	0.9381	0.6870
CR8	0.8052	0.2474	0.6632	0.5876	0.5833	0.8018	0.6148
NCC	14.4286	3.8001	3.6434	4.0531	6.8828	17.1891	8.3329
GC	0.6828	0.8791	1.6513	1.0597	0.9716	0.6828	0.9879
GD	0.4628	0.4214	0.6032	0.4303	0.3510	0.4314	0.4500
PI	0.3142	0.4654	0.6210	0.4442	0.3675	0.3707	0.4305
LNC	0.2355	0.3011	0.3391	0.3184	0.1713	0.1718	0.2562
VWC	2.6472	1.9559	3.4020	2.3704	2.6460	3.6048	2.7711

(e) Table of standard deviations

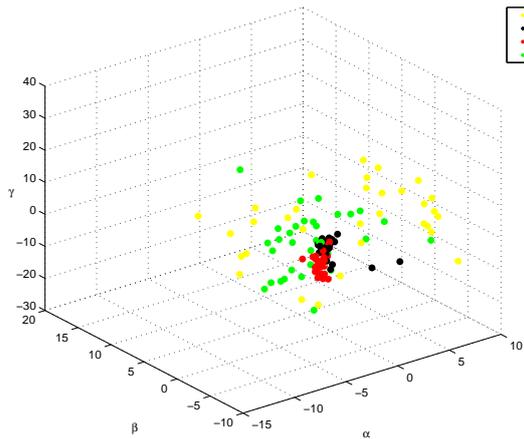
Figure A.10: Experiment #6 with ROI border 200



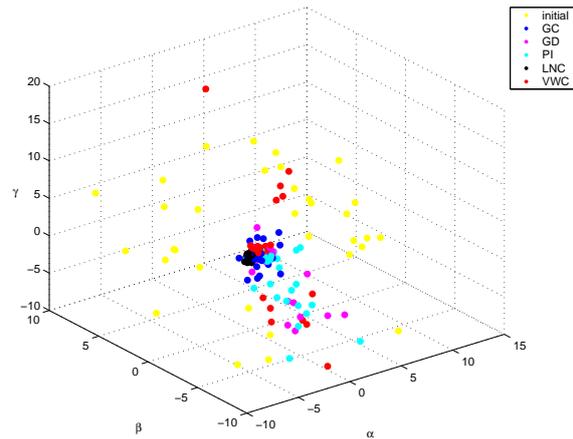
(a) Translational DOFs, first part of measures



(b) Translational DOFs, second part of measures



(c) Rotational DOFs, first part of measures



(d) Rotational DOFs, second part of measures

σ	x	y	z	α	β	γ	mean
MI8	1.6720	0.8514	0.6917	1.0967	1.1878	1.7287	1.2047
CR8	1.2564	0.8942	1.2873	1.1299	1.4755	1.2400	1.2139
NCC	17.2920	7.1832	7.7929	6.0454	8.4362	17.1089	10.6431
GC	2.1226	2.1876	2.6763	1.7287	1.5371	1.0921	1.8907
GD	2.5937	4.1031	1.6863	1.3283	1.9906	2.1853	2.3146
PI	2.6113	6.4527	1.8567	1.3358	2.3156	2.8147	2.8978
LNC	0.3224	0.2705	0.4470	0.3358	0.3036	0.2463	0.3209
VWC	6.3808	2.5137	3.3586	2.6080	3.1576	4.2305	3.7082

(e) Table of standard deviations

Figure A.11: Experiment #6 with ROI border 300

B Glossary

CT. Computed Tomography. A technique for producing 2D and 3D cross-sectional images of an object such as the human body from flat X-ray images

DOF. Degrees of Freedom

DRR. Digitally Reconstructed Radiograph. Artificial X-Ray image, obtained by using the voxel information of a CT data set.

DSA. Digital Substraction Angiography

FLE. Fiducial Localization Error. Deviation of the found positions of fiducial points from the real physical locations. Relevant for point- & feature-based registration algorithms. See also FRE.

FOV. Field of View. Angular parameter of a perspective view transformation.

FRE. Fiducial Registration Error. Measure of registration success, used with point-based registration systems. The difference between fiducial points in the two images is expressed, often as root-mean-square value.

Gold Standard. Any registration system whose accuracy is known to be high, is a gold standard registration system. Used for developing and validating new registration algorithms.

GPU. Graphics Processing Unit. The processor located on a graphics card, able to rapidly process 3D vector information for accelerated rendering.

Ground Truth. The known solution to a registration problem, in other words the true transformation, is called Ground Truth. It is very valuable to have experiments where Ground Truth information is given, however it is often very hard to find this information.

LINAC. Linear Accelerator. Device generating high-energy X-Ray radiation that can be focus on a specific target in order to perform radiation therapy.

Matching. To bring two modalities/images into spatial alignment in order to compare the modalities/images directly.

MIP. Maximum Intensity Projection, the equivalent to DRRs for MR images

Registration. Process of estimating the parameters for the matching of two modalities/images.

MRA. Magnetic Resonance Angiography

RMS. Root-Mean-Square, common scheme of taking the average of multiple measured values.

$$RMS = \sqrt{\sum_{i=1}^n |x_i|^2}$$

Stereotactic Frame. A metal ring that is usually attached to a patient's head and used for registration purposes.

TRE. Target Registration Error. Measure of registration success, containing the difference of a clinical relevant point in the images to be registered.

Bibliography

- [1] B. D. ALLEN, G. BISHOP, and G. WELCH, *Tracking: Beyond 15 Minutes of Thought*. SIGGRAPH Course 11, 2001.
- [2] J. BEIRLANT, E. DUDEWICZ, L. GYORFI, and E. VAN DER MEULEN, *Nonparametric entropy estimation: An overview*, International Journal of the Mathematical Statistics Sciences, 6 (1997), pp. 17–39.
- [3] G. C. BENTEL, *Patient Positioning and Immobilization in Radiation Oncology*, McGraw-Hill Professional Publishing, Oct. 1998.
- [4] V. BROY, *Voxel-basierte, elastische Volumenregistrierung von Patientendaten (in german)*, Master's thesis, Technische Universität München, Jan. 2003.
- [5] M. DÖTTER, *Fluoroskopiebasierte Patientenlageerkennung für die intraoperative Navigation in CT- und MR-Volumendaten zur Unterstützung chirurgischer Eingriffe in der Orthopädie*, PhD thesis, Technische Universität München, to appear, Mar. 2004.
- [6] K. ENGEL and T. ERTL, *Interactive High-Quality Volume Rendering with Flexible Consumer Graphics Hardware*, Eurographics, State of the Art Report (STAR), (2002).
- [7] O. FAUGERAS, *Three-Dimensional Computer Vision, A Geometric Viewpoint*, The MIT Press, 1993.
- [8] B. FEI, J. L. DUERK, D. T. BOLL, J. S. LEWIN, and D. L. WILSON, *Slice-to-Volume Registration and its Potential Application to Interventional MRI-Guided Radio-Frequency Thermal Ablation of Prostate Cancer*, IEEE Transactions on Medical Imaging, (2003).
- [9] M. FEUERSTEIN, *Design of a Planning Tool for Port Placement in Robotically Assisted Minimally Invasive Cardiovascular Surgery*, Master's thesis, Technische Universität München, to appear, Jan. 2004.
- [10] J. D. FOLEY, A. VAN DAM, S. K. FEINER, and J. F. HUGHES, *Computer Graphics, Principles and Practice*, Addison-Wesley, 1993.
- [11] P. E. GILL, W. MURRAY, and M. H. WRIGHT, *Practical Optimization*, Academic Press, Inc., 1981.
- [12] M. GROHER, *Development of a Planning and Navigation Tool for Endoscopic Treatment of Aortic Aneurysms - Computer Supported Implantation of a Stent Graft*, Master's thesis, Technische Universität München, to appear, Jan. 2004.
- [13] J. V. HAJNAL, D. L. HILL, and D. J. HAWKES, eds., *Medical Image Registration*, CRC Press, 2001.

- [14] A. HERO and O. MICHEL, *Robust entropy estimation strategies based on edge weighted random graphs*, 1998.
- [15] J. H. HIPWELL, G. P. PENNEY, T. C. COX, J. V. BYRNE, and D. J. HAWKES, *2D-3D Intensity Based Registration of DSA and MRA - A Comparison of Similarity Measures*, *Lecture Notes in Computer Science*, 2489 (2002), pp. 501–508.
- [16] P. HUBER, *Robust Statistics*, Wiley, 1981.
- [17] T. INOMATA, H. LIAO, M. IWAHARA, T. DOHI, and N. HATA, *Mutual Information-Based Markerless Registration for Surgical Image Overlay Navigation*.
- [18] INRIA, *Multimodal Image Registration by Maximization of the Correlation Ratio*, vol. 3378, 1998.
- [19] S. JONIC, P. THEVENAZ, and M. UNSER, *Multiresolution-based registration of a volume to a set of its projections*.
- [20] P. KEITLER, *Mathematical Methods of Image Processing for Automated Navigation in Endoscopic Treatment of Abdominal Aortic Aneurysm — Computer Aided Implantation of a Stent Graft*, Master's thesis, Technische Universität München, to appear, Jan. 2004.
- [21] D. KNAAN and L. JOSKOWICZ, *Effective intensity-based 2D/3D rigid registration between fluoroscopic X-ray and CT*.
- [22] P. LACROUTE and M. LEVOY, *Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation*, *Computer Graphics*, 28 (1994), pp. 451–458.
- [23] P. G. LACROUTE, *Fast volume rendering using a shear-warp factorization of the viewing transformation*, PhD thesis, Computer Systems Laboratory, Stanford University, Sept. 1995.
- [24] D. A. LAROSE, *Iterative X-Ray/CT registration using accelerated volume rendering*, PhD thesis, Carnegie Mellon University, May 2001.
- [25] J. B. A. MAINTZ and M. A. VIERGEVER, *An Overview of Medical Image Registration Methods*.
- [26] J. H. MATHEWS, *Numerical Methods for Computer Science, Engineering and Mathematics*, Prentice Hall, Inc., 1987.
- [27] E. MILLER, *A New Class of Entropy Estimators for Multi-Dimensional Densities*, 2003.
- [28] M. MITSCHKE, A. BANI-HASHEMI, F. GHELMANSARAI, A. KHAMENE, and J. POULIOT, *3D tomographic reconstruction from portal imaging for patient positioning*, 2003.
- [29] N. NAVAB and M. MITSCHKE, *Method and apparatus using a virtual detector for three-dimensional reconstruction from x-ray images*, May 2001.
- [30] G. P. PENNEY, J. WEESE, J. A. LITTLE, P. DESMEDT, D. L. HILL, and D. J. HAWKES, *A Comparison of Similarity Measures for Use in 2D-3D Medical Image Registration*, *Lecture Notes in Computer Science*, 1496 (2001).

-
- [31] J. P. W. PLUIM, J. B. A. MAINTZ, and M. A. VIERGEVER, *Mutual information based registration of medical images: a survey*, IEEE transactions on medical imaging, (2003).
- [32] W. PRESS, S. T. ABD W.T. VETTERLING, and B. FLANNERY, *Numerical Recipes in C, Second Edition*, CRC Press, Inc., 1992.
- [33] PROCEEDINGS MICCAI 1998, *A Novel Approach for the Registration of 2D Portal and 3D CT Images for Treatment Setup Verification in Radiotherapy*, 1998.
- [34] PROCEEDINGS MICCAI 2001, *Geometrical transformation approximation for 2D/3D intensity-based registration of portal images and CT scan*, 2001.
- [35] C. REZK-SALAMA, *Volume Rendering Techniques for General Purpose Graphics Hardware*, PhD thesis, Universität Erlangen-Nürnberg, Dec. 2001.
- [36] M. ROTH, *Intraoperative fluoroskopiebasierte Patientenlageerkennung zur präzisen Unterstützung chirurgischer Eingriffe*, PhD thesis, Technische Universität München, Mar. 2000.
- [37] D. B. RUSSAKOFF, T. ROHLFING, A. HO, D. H. KIM, J. R. A. JR., and C. R. M. JR, *Intensity-Based 2D-3D Spine Image Registration Incorporating One Fiducial Marker*.
- [38] D. B. RUSSAKOFF, T. ROHLFING, A. HO, D. H. KIM, R. SHAHIDI, J. R. ADLER, JR., and C. R. MAURER, JR., *Evaluation of Intensity-Based 2D-3D Spine Image Registration Using Clinical Gold-Standard Data*, in Biomedical Image Registration, Lecture Notes in Computer Science, June 2003.
- [39] C. STUDHOLME, *Measures of 3D Medical Image Alignment*, PhD thesis, University of London, Aug. 1997.
- [40] C. STUDHOLME, D. HILL, and D. HAWKES, *An overlap invariant entropy measure of 3D medical image alignment*, Pattern Recognition, 32 (1998).
- [41] V. TORCZON and M. W. TROSSET, *Using approximations to accelerate engineering design optimization*, 1998.
- [42] V. J. TORCZON, *On the Convergence of Pattern Search Algorithms*, 1993.
- [43] P. VIOLA, N. N. SCHRAUDOLPH, and T. J. SEJNOWSKI, *Empirical Entropy Manipulation for Real-World Problems*, in Advances in Neural Information Processing Systems, D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, eds., vol. 8, The MIT Press, 1996, pp. 851–857.
- [44] P. VIOLA and W. M. WELLS, *Alignment by Maximization of Mutual Information*, International Conference on Computer Vision, (1995).
- [45] P. A. VIOLA, *Alignment by Maximization of Mutual Information*, PhD thesis, Massachusetts Institute of Technology, June 1995.
- [46] F. WANG, B. C. VEMURI, M. RAO, and Y. CHEN, *Cumulative Residual Entropy, A New Measure of Information & its Application to Image Alignment*, International Conference on Computer Vision, (2003).

Bibliography

- [47] J. WEESE, R. GÖCKE, G. P. PENNEY, P. DESMEDT, T. M. BUZUG, and H. SCHUMANN, *Fast voxel-based 2D/3D registration algorithm using a volume rendering method based on the shear-warp factorization.*
- [48] L. ZOELLEI, E. GRIMSON, A. NORBASH, and W. WELLS, *2D-3D Rigid Registration of X-Ray Fluoroscopy and CT Images Using Mutual Information and Sparsely Sampled Histogram Estimators.*
- [49] L. ZOLLEI, *2D-3D Rigid-Body Registration of X-Ray Fluoroscopy and CT Images*, Master's thesis, Massachusetts Institute of Technology, Aug. 2001.