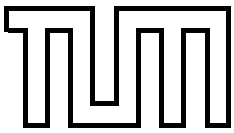
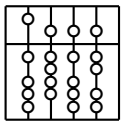


Technische Universität München
Fakultät für Informatik

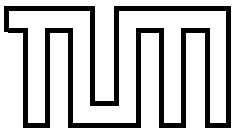


Diplomarbeit

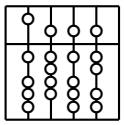
Mathematical Methods of Image Processing for Automated Navigation in Endoscopic Treatment of Aortic Aneurysms — Computer Aided Implantation of a Stent Graft

heARt: Heart Surgery Enhanced by Augmented Reality Techniques

Peter Keitler



Technische Universität München
Fakultät für Informatik



Diplomarbeit

Mathematical Methods of Image Processing for Automated Navigation in Endoscopic Treatment of Aortic Aneurysms — Computer Aided Implantation of a Stent Graft

heARt: Heart Surgery Enhanced by Augmented Reality Techniques

Peter Keitler

Aufgabensteller:	Prof. Gudrun Klinker, Ph.D. Dr. med. Robert Bauernschmitt
Betreuer:	Dipl.-Inf. Martin Bauer Dipl.-Inf. Hesam Najafi Eva Schirmbeck
Abgabedatum:	16. Dezember 2003

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

München, den 16. Dezember 2003 Peter Keitler

Abstract

In the last years, minimally invasive surgery techniques found their way into the operating theater. One very promising application is endovascular stent grafting for aortic aneurysms or dissections. At present, the three-dimensional CTA (Computed Tomography Angiography) model is only used for preoperative planning but is not available during operation. This situation is unsatisfactory for CTA provides by far more anatomical information than the two-dimensional intraoperative fluoroscopic images taken by a C-arm. Our goal is to close this gap by a *Computer Aided Navigation and Planning Tool*, in short CANP. CTA data is to be visualized intuitively for planning the correct position of the stent and its properties. During operation itself, CTA serves as basis for augmentation with intraoperative data acquired by the C-arm. By this means, we hope that radiation exposure for both, surgeon and patient, as well as usage of contrast agent, can be reduced. This thesis deals with mathematical challenges arising in this context. Before three-dimensional CTA and two-dimensional fluoroscopic images can be merged in one model, the two modalities have to be registered. For this purpose, the perspective geometry of the C-arm was determined by a calibration. Artificial X-ray images computed from CTA, so-called DRRs (Digitally Reconstructed Radiographs) are then to be overlaid with intraoperative radiographs. An efficient volume rendering technique based on 3D texture-mapping is proposed for this computationally expensive task. The pose of the CTA model is altered within a virtual C-arm setup by rigid transformations (rotations and translations). When DRR and radiograph coincide, the patient is registered to his CTA model and information can be exchanged between both modalities. Iterative best neighbor optimization techniques for an automatic registration procedure were evaluated. The similarity between DRR and radiograph is estimated by pattern intensity and gradient difference quality measures. This combination yields promising registration results in affordable time, though differences between both modalities, especially the influence of contrast agent and other instruments, have to be decreased in the future in order to achieve a really robust registration. Furthermore, different direct and indirect volume rendering techniques were evaluated with respect to their suitability for a fast as well as intuitive visualization of the augmented model.
Keywords: Stent, DRR, CTA, C-arm, 2D-3D rigid registration, calibration, three-dimensional visualization

Zusammenfassung

Im Laufe der letzten Jahre haben minimalinvasive Operationstechniken in den Operationssaal Einzug gehalten. Eine sehr viel versprechende Anwendung ist die endoskopische Stent-Implantation bei Aorta-Aneurysmen und -Dissektionen. Bis zum jetzigen Zeitpunkt wird das dreidimensionale CTA-Modell (Computertomographie, CT-Angiographie) nur zur präoperativen Planung, nicht jedoch während der eigentlichen Operation, benutzt. Diese Situation ist unbefriedigend, da CTA bei weitem mehr Einblicke in die Anatomie bietet, als zweidimensionale, intraoperative Röntgenaufnahmen eines C-Bogens. Unser Ziel ist es, diese Lücke mit Hilfe eines Planungs- und Navigationstools (Computer Aided Navigation and Planning Tool, CANP) zu schließen. Um die Position und die Eigenschaften des Stents zu planen, muss der CTA-Datensatz in einer intuitiven Art und Weise dargestellt werden. Während der eigentlichen Operation dient er als Basis, welche um intraoperative, mit Hilfe des C-Bogens gewonnene Informationen, erweitert wird. Dadurch hoffen wir, die Strahlen-Belastung für den Arzt und den Patienten, sowie den Einsatz von Kontrast-Mittel, verringern zu können. Diese Diplomarbeit befasst sich mit mathematischen Fragen, die sich in diesem Zusammenhang ergeben. Bevor dreidimensionale CTA- und zweidimensionale Fluoroskopie-Daten in einem Modell vereint werden können, müssen beide Bildgebungsverfahren miteinander registriert sein. Aus diesem Grund wurden die perspektivischen Eigenschaften des C-Bogens kalibriert. Künstliche Röntgenbilder, die aus den CTA-Daten berechnet werden (DRR, Digitally Reconstructed Radiograph), sollen nun mit intraoperativen Röntgenbildern überlagert werden. Eine effiziente Volumen-Visualisierungstechnik basierend auf 3D-Texturen wird für diese rechenintensive Aufgabe vorgeschlagen. Die Lage des CTA-Modells innerhalb eines virtuellen C-Bogens wird durch rigide Transformationen (Translationen und Rotationen) verändert. Sobald der "DRR" mit dem Röntgenbild übereinstimmt, ist der Patient mit seinem CTA-Modell registriert und Informationen können zwischen den zwei Modalitäten ausgetauscht werden. Zur automatischen Registrierung wurden iterative "best neighbor" Optimierungsverfahren evaluiert. Die Ähnlichkeit zwischen "DRR" und Röntgenbild wird mittels der Qualitätsmaße "pattern intensity" und "gradient difference" bestimmt. Mit dieser Kombination wurden viel-versprechende Ergebnisse bei akzeptabler Rechenzeit erreicht. Jedoch müssen die Unterschiede zwischen den beiden Modalitäten, im Besonderen der Einfluss von Kontrast-Mittel sowie Operationsinstrumenten, künftig verringert werden, um eine robuste Registrierung durchführen zu können. Darüber hinaus wurden versch. direkte und indirekte Volumen-Visualisierungstechniken unter den Gesichtspunkten einer schnellen und intuitiven Darstellung des augmentierten Modells untersucht.
Schlüsselwörter: Stent, DRR, CTA, C-Bogen, rigide 2D-3D Registrierung, Kalibrierung, dreidimensionale Darstellung

Preface

About this work This Diplomarbeit (diploma thesis) for the studies of Dipl.-Informatik (information science) originated in a cooperation between the Deutsches Herzzentrum München (DHM) and the Chair for Applied Software Engineering at the Technische Universität München (TUM) and has been given the mnemonic HEART as abbreviation for *Heart Surgery Enhanced by Augmented Reality Techniques*. It aims at the introduction of virtual- and augmented reality techniques to heart surgery. A three-dimensional computed tomography (CT) model is used as basis for augmentation with data acquired by some means or other. Our primary goal is to improve availability and awareness of information in order to improve or at least speed up the decision-making process in the operating theatre. For the sake of completeness, a brief overview of the sub-projects shall be given here.

Two theses deal with robot surgery. Marco Feuerstein describes a system for planning the optimal placement of robot arms in the patient's body — also called port placement — with regard to issues like optimization of the arm's elbow-room without affecting other organs [19]. A system for tracking robot arms intra-operatively was developed by Jörg Traub [71]. The position and orientation of the arms is visualized in real-time within the patient's CT model. This compensates for the rather restricted perspective of the camera attached to one of the robot's arms and helps the surgeon in navigation.

This theses and a fourth one are concerned with endovascular stent grafting as treatment of aortic aneurysm or dissection. A *Computer Aided Navigation and Planning* tool (CANP) was developed in this context. It provides functionality for preoperative planning of stent placement in a computed tomography angiography (CTA) model as well as intraoperative support for catheter navigation. The two works are closely related, as will be seen in chapter 1. Martin Groher describes the architectural aspects, whereas this work concentrates on important mathematical issues, especially on the automatic registration of the three-dimensional CTA model with two-dimensional intraoperative radiographs. Registration is a necessary prerequisite for combining information from different image sources in one visualization. Furthermore, three-dimensional visualization techniques are investigated for a suitable representation of the acquired data.

Acknowledgements I am indebted to Prof. Gudrun Klinker (TUM) and PD Dr. Robert Bauernschmitt (DHM) for making this thesis possible at all. Furthermore, I would like to thank my supervisors Eva Schirmbeck (DHM), Martin Bauer and Hesam Najafi (both TUM) for their help and advice concerning medical as well as theoretical issues. Thanks also to my colleagues Martin Groher, Jörg Traub and Marco Feuerstein for motivating me whenever it was necessary, and also for the fun we had. Special thanks go to Martin for the really great teamwork. Moreover, I am deeply grateful to Bernhard Wende for proofreading this thesis.

I am much obliged to my parents Josef and Ingeborg Keitler for making my studies actually possible. And last but not least, I want to thank Agnes for enduring these weeks full of privations and also for reminding me of what is most essential.

Contents

List of Figures

1 Introduction

In the last few years, cardiac and vascular surgery more and more often make use of keyhole surgery techniques and endoscopic methods as shown in figure 1.1. Here, only small cuts are made on the patient's skin where special instruments are injected, guided to the region of interest and handled — sometimes remotely — by the surgeon. Treatments range from catheter injection to injection of robot or telemanipulator arms. Its benefits are obvious: First, the mortality rate as well as the morbidity rate is much lower when endovascular treatment is performed, compared to open surgery. Moreover, the patient's convalescence is very short, which is not only a benefit in terms of money. In almost all endovascular operations the heart-lung-machine is not used and it is even possible to treat high-risk patients with endovascular or endoluminal techniques [5, 4, 51, 27, 24]. Similar advantages also apply to keyhole methods in cardiac surgery [19, 71]. On the other hand, there is one problem which must be solved when open surgery should be avoided: How can the surgeon actually see where to operate, discern her instruments or spot the region of interest? In order to solve these difficulties, imaging techniques will be needed which make it easier or principally possible to treat patients with keyhole surgery.

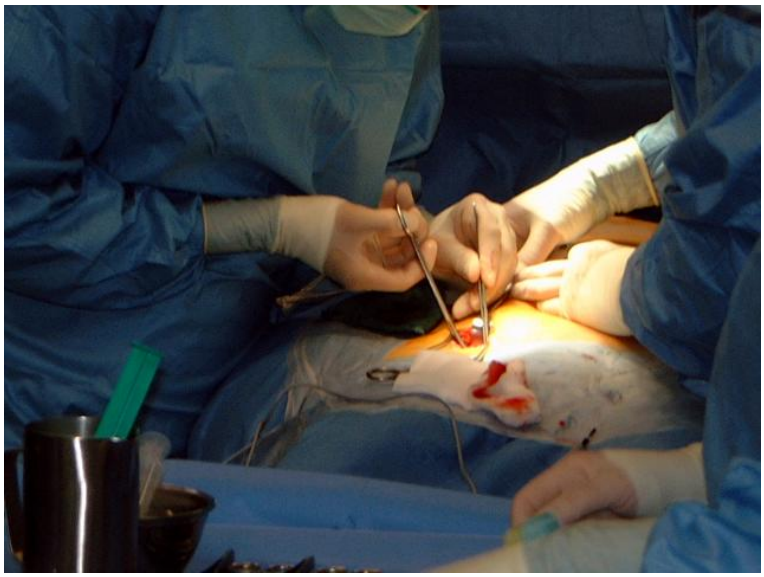


Figure 1.1: Endoscopic treatment: The patient's femoral artery is being prepared for insertion of a catheter

1.1 Imaging Techniques for Endoscopic Surgery

Basically, imaging techniques for endoscopic surgery are divided into two purviews, a pre-operative one, where surgeons plan an operation and an intra-operative one, where treatment itself is carried out. Mostly, the pre-operative planning is done one day before the actual surgery. There is no time pressure — the physician can take an extended look at all image material of the patient, compare, select or even process the data. Imaging techniques for intra-operative purposes have, on the other hand, different requirements. As time plays an important role while treating a patient, the faster the surgeon can get and view images, the better. Moreover, navigation in image data during operation should not require much interaction, and thus, enhanced visualization techniques will become more and more essential.

Pre-operative planning is mostly aided by computed tomography (CT) or magnetic resonance (MR) images which allow surgeons to view the region of interest in 3D or browse through axial, frontal and sagittal slices made of the body's interior (see figure 1.2). Axial

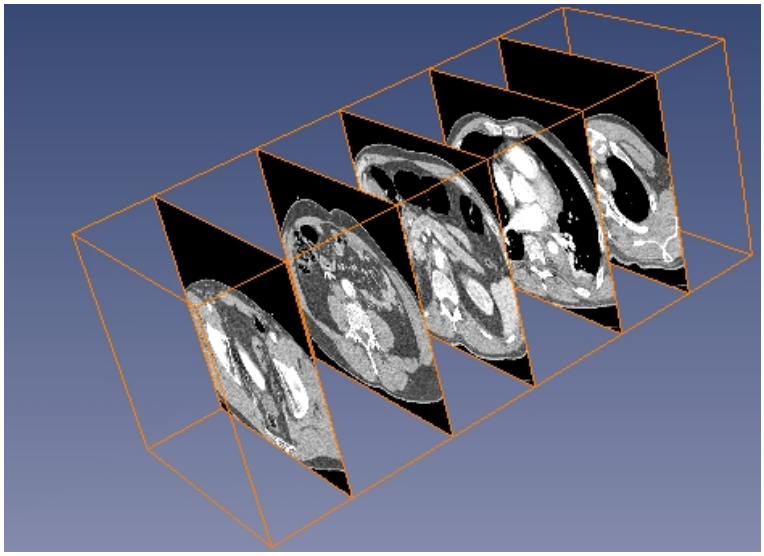


Figure 1.2: Axial CT slices showing the thorax enhanced by contrast agent to visualize vessels

slices are recorded by the CT or MR scanner itself whereas sagittal and coronal slices are reconstructed from the axial raw data. A three-dimensional representation of the tomographic data can also be displayed in order to present the data in a much more intuitive form. As the scanning procedure takes quite a long time and also requires huge and immobile technical equipment, it can only be performed before but not during the surgery. However, CT and MR images provide a vast range of information. Besides bones, they differentiate tissue types and organs and can, when extended by angiography[1], even display vessels and thrombus [6]. There are two major disadvantages of MR — first, MR cannot display calcium which is essential for e.g. vessel investigation, and second, MR images can be distorted by ferro-magnetism which emerges when using metallic grafts [6, 70]. Thus, CT is the dominant

[1]Computed Tomography Angiography, CTA

technique to do pre-operative planning. Angiography itself or digital subtraction angiography, is a method to visualize the lumen (the vessel's interior) of arteries and veins. Contrast agent, a liquid which absorbs X-rays, is injected in the vessel to be visualized, followed by the shooting of several X-ray images. Thus, a film-sequence can be generated where the lumen of the contrast-enhanced vessel (e.g. the aorta) can be captured [23]. Angiography is used in both parts, the pre- and intra-operative one. During operation, angiography can be done with a C-Arm, which is a portable X-ray device and shoots images in short intervals (see figure 1.3).[2] The X-Ray images can be viewed instantly by the operating physician but only provide a two-dimensional impression of the patient's body without showing any thrombus (see figure 1.4). As a matter of fact, many other techniques for visualization in medicine are

Figure 1.3: C-arm which is used for intra-operative X-ray imaging

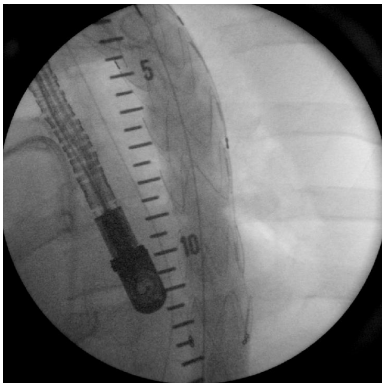


Figure 1.4: C-arm angiogram showing a part of the (contrast-enhanced) aorta where a stent is located (wires)

used, like endoscopic (stereo-)cameras or Transoesophageal Echocardiography (TEE), but as they are of little use for the purpose of stent implantation, they are not discussed in detail here.

1.2 Aortic Stenting — Description and Practice

According to [1] an *aortic aneurysm* is a sac formed by expansion of the aorta which is in direct communication with the blood in the aorta's lumen. The normal diameter of the thoracic aorta is less than four centimeters for the ascending, and less than three centimeters for the descending portions. A diameter exceeding five centimeters is usually considered an aneurysm. A diameter of the aorta greater than 1.5 times the normal diameter also constitutes an aneurysm. *True aneurysms* contain all three vascular layers[3] and are usually caused by degenerative diseases such as atherosclerosis and cystic medial necrosis, though the etiology may also be congenital, inflammatory, infectious or traumatic, whereas the *false*

[2] The reason for taking a picture sequence instead of just a single picture is the unpredictability of the contrast agent's location at a certain point of time. Hence, for a well-contrasted image, a sequence called angiogram is needed.

[3] intima, media and adventitia

aneurysms — usually caused by trauma or infection — are composed of only an outer adventitial layer. Corresponding to their exact location in the aorta, aneurysms are further categorized as either thoracic[4], thoracoabdominal[5] or abdominal[6] aortic aneurysms. Another closely related indication is an *aortic dissection*. In [1] it is described as a splitting of the media of the aortic wall by blood, which may be caused by a tear in the aortic intima, and hence, blood passing from the lumen into the wall, causing the intima to be torn from the wall by a variable distance. A dissection may also occur by spontaneous bleeding of the vasa vasorum causing intramural hematoma without rupture of the intima and consequently no connection between the lumen and the intramural dissection. The *Stanford* classification recognizes *Type A* (involvement of ascending aorta alone or involvement of ascending and descending aorta) and *Type B* (involvement of descending aorta alone) dissections. Please refer to [1] for an exhaustive description of the afore mentioned diseases. Normally, aneurysms and dissections do not come along with any disturbances for the concerned person and are therefore very often diagnosed with by accident. However, aneurysms and dissections denote a high risk of abruption of a thrombus or even rupture of the aorta and therefore should be treated as soon as possible.

The conventional form of treatment allows for insertion of a bypass into the open thorax and therefore bears high risks for the patient. Therefore, a new endovascular technique called stent grafting evolved and became more and more popular in the last few years. A so-called stent is pushed through the aorta with a catheter and unfolded at the problem region. There, it stabilizes the intima of the aorta. Interestingly, mortality rates of the first approach are significantly higher (about 50%) than those of the second (less than 10%)[10, 53, 56, 57].

1.2.1 Pre-Operative Planning

Mostly, aneurysms are not recognized by the patient unless they reach quite large dimensions. A diameter of about six centimeters e.g., would cause rupture in the abdomen[60]. Thus, the most common way to diagnose aneurysms is a routine check via CTA. Here, the problem is categorized and the data is prepared for the operating surgeon to be processed. As a matter of fact, there is a strict separation between surgeons and radiologists in german hospitals unlike in other countries. Radiologists merely edit and pre-process the raw data of the CT scanner, which mostly results in a colored 3D reconstruction of the aorta and the spine for orientation reasons (see figure1.5). They do not perform any surgical treatments on the patient. This is done by the operating physician who just receives the edited CT data the day before surgery, detects the location in the patient's aorta affected by an aneurysm or a dissection and determines an appropriate stent, which is mainly characterized by its length and its diameter. Sometimes the stent graft must have a special shape, namely, if the aneurysm covers more branches of the aorta like the two iliac arteries as shown in figure 1.6. In planning, as well as during implantation, special attention has to be paid on not to cover branching-off vessels, like the carotid artery or the coronary arteries. Additionally, the CTA data provides important knowledge about the aortic anatomy which is necessary for catheter navigation and also for detection of potential problems in operation, like kinking of the aorta or the femoral artery, which also has to be traversed by the catheter. Unfortunately,

[4]Thoracic Aortic Aneurysm (TAA), aneurysm located in the thorax or chest
[5]Thoracoabdominal aortic aneurysm (TAAA), aneurysm located in both, thoracic and abdominal regions
[6]Abdominal Aortic Aneurysm (AAA), aneurysm located below the chest in the abdomen or belly

all this data is not accessed during the operation. Hence, the physician has to keep in mind where the aneurysm is located, where kinking would cause the injected catheter to stop or where branching-off vessels are connected to the aorta.

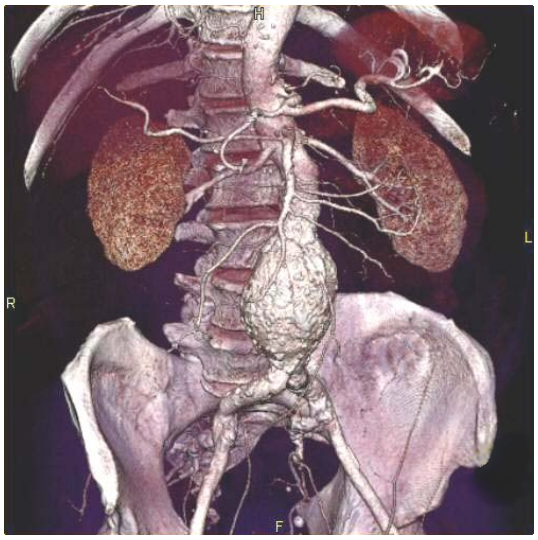


Figure 1.5: A 3D reconstruction of an abdominal aortic aneurysm

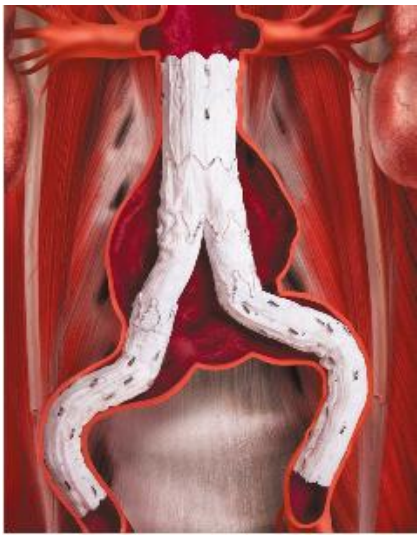
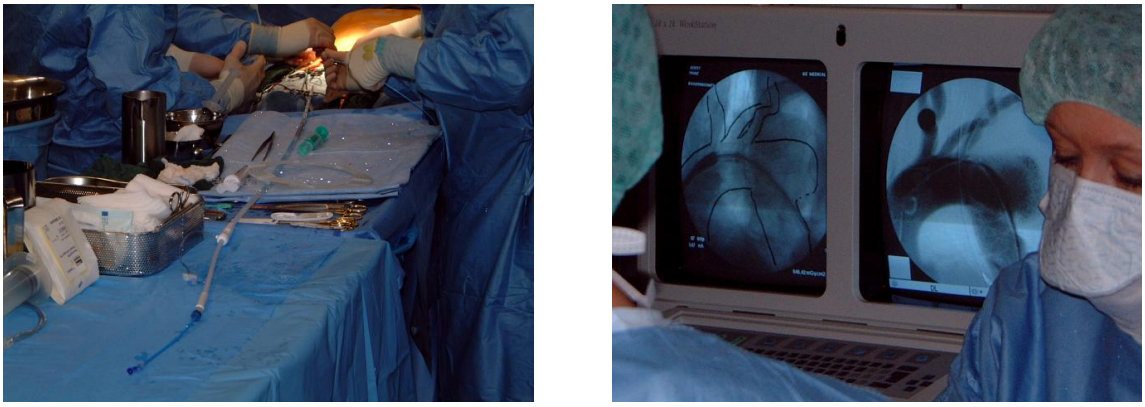


Figure 1.6: Stent graft for an AAA which also covers the two iliac arteries

1.2.2 Stent Grafting

When the patient has been anaesthetized and prepared for surgery, a pigtail catheter is inserted into the patient's shoulder and advanced to the region of interest. It will be used for injection of contrast agent into the aorta in order to visualize their lumen in X-ray images. Then, the femoral artery, which is well palpable since it is located directly below the skin, is detected and dissected with a scalpel. By using a guide wire with a flexible tip, a catheter is inserted on which the stent graft (see figure 1.7) is attached (*Seldinger technique*, [1]). It is advanced through the aorta until the surgeon gets the feeling of increased pressure, which means that the catheter got stuck in a kinking, or until the problem region is reached. In order to assure the current location of the stent graft, an angiogram is taken by the C-arm, enhanced by contrast agent as described in section 1.1. When treating TAAs, the way of the catheter is quite long: it must be pushed all the way through femoral, iliac, (infra)renal and thoracic aorta to finally reach the de- or ascending aorta near to the heart. This holds a lot of perils, and often, many contrasted X-ray images must be taken in order to avoid complications. Another technique sometimes used is to take a contrasted image and capture the aorta's (and also other branching vessels') contours by merely drawing them on the monitor as shown in figure 1.7. Then, subsequent images may be interpreted even if they are not enhanced by contrast agent. Naturally, after drawing the contours, the C-arm must not be moved any more. When the surgeon is certain to have the stent in the right place, it is expanded and the catheter is extracted. In some cases, an inflatable balloon is injected and guided to the stent in order to fix it to the aorta wall by using air pressure (balloon-expandable stent grafts).



(a) (b)

Figure 1.7: Endovascular stent grafting, (a) a new stent graft is attached to the catheter, (b) C-arm console, the left screen shows an X-ray image where the contours of the aorta and its branches are plotted.

1.3 Computer Aided Navigation and Planning Tool

Endovascular stent grafting as treatment of aortic aneurysm is a quite new technique, however, is believed to be very promising, and more and more hospitals tend to treat aneurysms via endovascular, instead of conventional open surgery. Due to the early stage of development it is not amazing that the usage of stents is not very experienced but has a rather improvisational character. In particular, there is no technical equipment to meet the special needs of stent implantation besides the standard equipment like CTA and C-arm[7]. E.g. the contours of the aorta have to be charted with a marker on the screen of the C-arm system so that they may be used for the interpretation of subsequent non-contrasted radiographs. Furthermore, the three-dimensional CTA model is only used for pre-operative planning but is currently not available during operation, except in the surgeon's memory. This lack of information is compensated by excessive usage of the C-arm. Whenever the surgeon is unsure about the current position of the stent catheter, a new angiogram is taken. This procedure may be necessary more than 20 times, depending mainly on the anatomy of the patient and the course of operation. The hazards for surgeon and patient, coming from extensive usage of X-ray angiography, do not have to be explained.

This situation is unsatisfactory for CTA models could provide the anatomical information that is missing in two-dimensional intra-operative fluoroscopic images taken by a C-arm. The C-arm, however, is necessary for providing information about the current position of the catheter. Therefore, we propose a *Computer Aided Navigation and Planning tool* (CANP) which is able to provide all available information in one modality, before as well as during operation. It is supposed to guide the physician through the pre-operative planning procedure by providing methods to virtually place and display a stent at the intended position in an existing CTA model. The information can be stored persistently and will serve

[7] At the DHM a GENERAL ELECTRIC MEDICAL SYSTEMS, SERIES 9800 C-arm is in use

as a reference for the surgeon to control the implantation progress during operation. During treatment of the patient, CANP will be able to process the latest C-arm image and CTA data such that the stent's momentary location is kept up-to-date in the CTA model. Thus, surgeons are reported permanently about the current position of the catheter, the planned stent position, the distance between the planned and the current position as well as about kinkings or other anomalies that may be of interest for the surgeon.

We expect that CANP will help to reduce dramatically X-ray exposure as well as the amount of contrast agent, the patient's circulation is injected with. In order to achieve the goal of merging information from two different image modalities, it is necessary to know - in abstract terms - how modality A corresponds with modality B. I.e., it is necessary to know, where modality A is located with respect to modality B or vise versa. This is commonly referred to as *image registration*. In our case, the pre-operative three-dimensional CT-model has to be registered with the current two-dimensional X-ray image from the C-arm. The architectural aspects of the CANP system as a whole are described by Groher [25]. This thesis will concentrate on important graphical and mathematical issues arising in the context of the *registration problem*. Thereby, the remarks shall not stringently be focused on stent grafting, but also give insights in other medical fields of application.

In the next chapter, the computational aspects related to medical image registration in common and specific to 2D-3D registration to be implemented in CANP, shall be outlined. In this context, also an overview of existing research projects with similar intentions shall be given. The different aspects of the problem at hand will be much clearer then. Therefore, the overview of the whole document shall be postponed until then.

2 Requirements and Goals

In the last chapter, we already got an impression of what shall be achieved by the introduction of CANP in the context of stent grafting. The main goals, viewed from the medical point of view, shall be shortly summarized here.

- A planning tool provides an intuitive way for the visualization of, and the navigation within the 3D patient model acquired by CTA.
- Thereby, possibilities are to be provided in order to locate the aneurysm or dissection and determine where the stent is finally placed.
- A navigation tool is supposed to be able to visualize, besides the 3D model itself, the planned location of the stent also during operation.
- Furthermore, the model shall be augmented by information about the current location of the stent, acquired by fusion of intra-operative X-ray images with the 3D model. This provides the surgeon with a full overview of all available information. Especially, the current location with respect to the planned location of the stent is always represented.
- This improvement in information quality shall lead to reduced usage of the C-arm, hence, less X-ray exposure for patient and staff. This will also result in a decreased amount of contrast agent needed for visualization of vessels.
- The image fusion or registration procedure must not delay the progress of operation. Therefore, the time needed to register one X-ray image with the existing model, should not exceed one or two minutes.
- The accuracy of information should meet the medical requirements. In the context of endovascular stent grafting, this means that the predicted position, especially in axial direction, does not deviate more than several millimeters from the actual position. The other directions are not that important, for the stent is always assumed to be located within the aorta.
- The merged model is to be presented in a manner that supports cognition of, and navigation in the data as good as possible. To achieve this, appropriate visualization techniques are needed.

Before our own approach to this 2D-3D registration problem shall be proposed in section 2.2, we want to take a look on related publications and similar research projects in section 2.1. Section 2.3 will outline the structure of the document.

2.1 Related Work

This section is split up in two halves. 2.1.1 gives a general overview of image registration in the medical context. After that, we will focus on approaches related to ours in 2.1.2.

2.1.1 Image Registration in Medical Applications

Image registration is necessary whenever information from different modalities shall be gathered in one resulting model. There are a lot of different imaging devices used for examination of the human body. Besides the common and well-known CT and X-ray diagnostics, also other techniques as e.g. SPECT[1], PET[2] or MRT (all provide 3D information) come into operation for certain purposes. But also other factors have impact on a suitable approach towards a specific registration problem. For example, different solutions are necessary to register soft-tissue structures, as compared to bony structures. An exhaustive introduction to image registration in the context of medical applications is given by Maintz [50]. His "Survey of Medical Image Registration" describes an intuitive classification scheme based on nine criteria and lists important works belonging to the respective category. It shall be presented in order to narrow down our own registration problem.

Dimensionality Image information is provided either in two-dimensional or three-dimensional form. This yields the possible combinations *2D-2D*, *3D-3D* and *2D-3D registration*. Furthermore, the dimensionality is increased, when *time series* have to be considered. They are acquired, e.g. in order to monitor tumor growth or bone growth in children and further complicate the problem.

Nature of Registration Basis A very important issue is the kind of information that is used for registration. Three different types of *registration bases* may be distinguished, extrinsic, intrinsic and non-image based. Registration according to an *extrinsic registration basis* relies on additional information from fiducials which are either implanted or attached to the patient's skin and are visible in the image modalities involved. With this information, a transformation can be calculated directly, without the need of complex optimization algorithms. However, it is restricted to be a rigid transformation, since tissue deformation, visible in the patient data itself, is completely disregarded. Drawbacks are the complications arising from usage of fiducials, maybe combined with additional perils for the patient in the case of their implantation.

Therefore, an *intrinsic registration basis* is often used. In this case, registration is either *segmentation based*, relying on pre-segmented image data (see chapter 5) or *voxel property based*, working directly on the unmodified voxel data (see chapter 4). A third possibility are salient points called *landmarks*, which have to be set by the user interactively. All those types of an intrinsic registration base are principally suited not only for finding rigid or affine transformations, but also for registration of *deformable models*, though a lot of user-defined points would be needed in the case of landmark based non-rigid registration. Deformable models

[1] Single Proton Emission Computed Tomography
[2] Proton Emission Tomography

are normally represented by nets, whereby the edges are modelled by localized functions like splines. Algorithms depending on deformable models, normally proceed iteratively, performing only small deformations at a time and are rather suited for finding local registrations, as e.g. for a certain organ. They normally need a good initialization in order to converge. The quality of segmentation based registration is always restricted by the quality of the segmentation step, which commonly cannot be fully automated (see chapter 5), but rather comes along with cumbersome manual interaction (see chapter 5). Voxel property based registration techniques can be further distinguished. The first method relies on the computation of center of gravity and its principal orientations (principal axis) from the image zeroth and first order moments, which are in turn derived from the pixel or voxel intensities. The principal points and the axis can then be aligned. This technique is not very accurate but broadly used either as pre-registration step or when registration quality is not a major issue. The second method relies on the full image content. Similarity measures (see chapter 6) are minimized iteratively, which causes high computational costs. These may be minimized by using a rough pre-registration, achieved through other means. The technique provides good registration quality and is therefore especially suited for optimization.

The last major category under the aspect of registration basis is *non-image based registration*. It fully relies on *calibrated coordinate systems*. These may be given by, e.g., by two adjacent CT or X-ray scans without moving patient and imaging device, or by tracking them. In a recent project belonging to this category, e.g. a surgical robot as well as the patient is tracked by an optical tracking system [71, 19]. By this means, the position of the robot's arms within the patient, lying in operating theater, can be displayed in the CT model. In this approach, the 3D model is registered to the patient via fiducials visible in the CT scan.

Nature of Transformation Four categories are distinguished under the aspect of *transformation type*. The most restrictive is *rigid transformation*, which only allows for translations in the x, y and z-axis direction and rotations around those axis, thus six degrees of freedom in the three-dimensional case. *Affine transformations* additionally include scaling (seven degrees of freedom) or even change of angles (twelve degrees of freedom) and *projective transformations* allow parallel lines to intersect (15 degrees of freedom). They do not have a physical basis in medical imaging except for 2D-3D registration. So far, the transformations have been linear. With a *curved transformation*, however, a line may even be mapped to a curve or a plane to a surface respectively, which is much more complex but provides more accuracy in some cases. Theses transformations are mainly considered in combination with deformable models.

Domain of Transformation A transformation is called *global*, if it applies to the whole image or model and *local*, when it is calculated individually for different parts of the body, e.g. single bones or organs. In most cases, rigid or affine transformations are assumed, when a *global transformation* is to be found. Curved models are normally only used for calculation of *local transformations*.

Interaction Depending on the amount of *user interaction* which is required by the registration system, it is either called *interactive*, *semi-automatic* or *automatic*. In the first category, the

work is done by the user, supported by appropriate visual or numerical impressions provided by the system. Algorithms, which work automatically, after they have been initialized by the user, belong the second category. Some algorithms also need steering by the user, e.g. the rejection of an obviously false hypothesis, suggested by the algorithm. Automatic algorithms do not depend on human assistance at all.

Optimization Procedure Parameters of the transformation may either be calculated *directly*, e.g. in the case of corresponding point sets, specified by landmarks or fiducials, or searched for *iteratively*. The latter technique depends on the formulation of a registration paradigm as a function of the transformation parameters. The registration paradigm is normally represented by a *similarity measure* which gives information about the quality of a certain transformation. The optimal parameters are searched for iteratively by a change in parameter space, followed by an evaluation of the resulting similarity. This is computationally much more expensive than a direct calculation, and a broad range of algorithms has been proposed for solving this problem.

Modalities Involved Different image modalities occurring in the medical context have already been mentioned above. Registration is now either *mono-modal*, if two instances of the same modality shall be registered or *multi-modal*, if instances from different modalities are used. Furthermore, a certain instance can be registered to an already existing mathematical model. For example, an imaged femur could be registered to a general model of the femur, which is independent of the patient. This is called *modality to model*. The last category is called *patient to modality* and describes the registration of the patient itself with one of his models. As an example may serve the robot application mentioned above.

Subject When all of the images, used for the registration task, are acquired of one single patient, we refer to it as *intra-subject* registration, whereas instances from different patients are used in *inter-subject* registration. If one of the modalities involved, does not belong to a certain patient but represents an abstraction created from different persons, we speak about *atlas* registration.

Object This last aspect serves as a classification of the different parts of the human body. A solution of the registration problem strongly depends on the type of tissue, imaged by the involved modalities. Applications dealing with the human brain, have much higher requirements with respect to accuracy of registration than those dealing with the leg. Soft-tissue structures need a treatment different to bones, since they sometimes vary a lot between two different acquisitions, whereas bones remain in a quite fix position. Therefore, deformable models are used especially in this context.

Also a very good survey of automatic 3D surface registration techniques has been presented by Audette [3]. Based on an appropriate choice of the underlying transformation, methods for model representation and similarity criteria are evaluated. Then, different matching algorithms and global optimization schemes are discussed. Thereby, the focus

lays on iterative optimization of a global transformation based on different similarity measures. Although the problem at hand is not a 3D-3D but rather a 2D-3D registration problem, the ideas described here are of value for us, since they may easily be adapted.

Non-rigid registration methods for intra, and inter-subject registration using basis functions and splines is described by Rueckert [66]. The author describes different mechanical models like elastic deformation, fluid deformation or FEM[3]. The latter one is also used by Roth [63].

2.1.2 2D-3D Registration

One of the first publications dealing with intensity-based registration of a CT model with (planar film) radiographs was published by Lemieux [47]. He also introduced the term Digitally Reconstructed Radiograph (DRR). Previous work in this context had been restricted to usage of pre-segmented CT data, with the disadvantages described above (segmentation based intrinsic registration base). Gottesfeld Brown extended this approach [8]. She describes an iterative approach for finding the optimal transformation. Thereby, the correlation ratio between a two-dimensional DRR computed from the CT model and an original radiograph, is to be minimized. The parameters to be optimized, describe the pose (rigid transformation) of the CT model within the virtual X-ray modality used for computation of DRRs. In this context, the necessity for calibration of the X-ray image modality is outlined. Principal differences between X-ray fluoroscopy and CT are listed, since they may affect adversely the quality and stability of registration. Furthermore, it is described, how the CT intensity values have to be interpreted in order to calculate a DRR.

Similar applications have later been described by Guéziec in the context of guiding a surgical robot and also by Penney [26][58]. Both are based on the same approach. Two new similarity measures especially suitable in combination with a voxel property based registration basis, pattern intensity and gradient difference, have been introduced by Penney, along with their evaluation compared to already existing measures [58]. They have also been successfully tested by Imamura and Wein [31, 76]. Imamura describes a system that shall aid in endovascular stent grafting, hence, a system very similar to ours. Interestingly, he gets by on using only parallel instead of perspective projection for computation of DRRs, unlike all other publications. He justifies this simplification with the unimportance of information about image depth in the special case of stent implantation.

Generation of DRRs is a computationally expensive task, but lot of DRRs are needed in the iterative approach pursued here. Therefore, efficient volume rendering techniques are necessary and have been evaluated by Göcke and Weese in the context of 2D-3D image registration [21, 22, 75]. A completely different approach for a speed-up of this task has been described by Roth [63]. Training images, created pre-operatively, are used for interpolation of DRRs intra-operatively. Meanwhile, however, they have replaced this approach by hardware based volume rendering techniques and completely abandon training images [64]. A multi-resolution approach can help to reduce costs by using coarse images first. Along the edges of a pyramid, image resolution is increased from top to bottom during the iteration, until the full resolution has been reached. This also smoothes error behavior and therefore helps to avoid problems arising from local minima during optimization [33].

[3] Finite Element Method

2.2 Problem Statement

Now, it is about time to formulate an approach which meets the requirements listed in the beginning of this chapter. It may be useful to classify it according to the scheme introduced in 2.1.1. Two-dimensional radiographs shall be registered with three-dimensional CT data. Thus, we are confronted with a 2D-3D registration problem, as already stated before. It shall be solved using an intrinsic registration basis since neither fiducials nor tracking techniques shall be applied. This guarantees a compact system which is independent of additional hardware like a tracking system and manages without the sophisticated application of fiducials. Furthermore, it provides a good registration quality.

Although an approach using parallel projection for DRR computation has been proposed [31], we will stick to perspective projections, since they provide a better quality without increased computation times, if efficient volume rendering algorithms are used. The complexity of perspective needed for calculation of appropriate projections from three-dimensional CT data requires the calibration of the C-arm. As soon as this has been achieved, the remaining transformations are assumed to be rigid, since we could not take advantage of curved transformations. This becomes clear from the fact that we just want to extract the current position of the stent from an intra-operative radiograph. Therefore, we could not benefit from a model which perfectly aligns each rib separately, because only the global transformation is of interest for us. It is determined best with respect to all bony structures visible in the image. Furthermore, computation time is a major topic in intra-operative applications and it is wise to restrict the modelled degrees of freedom as far as possible. As a consequence of the intrinsic registration basis, we cannot rely on direct point correspondences but use image intensity values instead for registration. Therefore, we cannot compute a transformation but need an iterative approach for finding an approximation of its optimal value. A suitable algorithm has to be found in order to accomplish the optimization with as few iterations as possible, since each iteration increases the number of evaluations of the similarity measure, and each evaluation comes along with the expensive creation of an additional DRR. Due to the same reason, a pre-registration will be necessary. Each iterative approach would be inefficient without a good initial value for the iteration. Sampling the similarity measure over a broad range is not affordable in an intra-operative application. The initialization step may be performed automatically, e.g. by attaching fiducial markers to the patient's skin, or interactively. Since the latter is acquired quite quickly, if parameters like the inclination of the C-arm or the horizontal alignment of the patient are assumed to be known, we propose an interactive approach, for the sake of a compact system.

This categorization yields a system similar to those described above in 2.1.2. In the introduction, is has already been mentioned that especially architectural aspects of CANP will not be considered in this work. This includes an user interface for doing the pre-registration step interactively by changing the six rigid transformation parameters [25]. We will therefore assume that the automatic registration algorithm is started at a position near to the exact registration.

2.3 Structure of the Document

First, a calibration of the C-arm is required in order to determine the intrinsic parameters of the perspective projection. Chapter 3 will care about this issue. There, also an overview of iterative optimization schemes shall be given, since they are needed not only for the actual registration but also for calibration of the C-arm. Chapter 4 covers basic theory about volume rendering techniques that are principally applicable in medical imaging. Some of them are necessary for the computation of DRRs. A whole chapter is devoted to volume rendering techniques because they are not only indispensable for our intensity based registration approach but also play a major role in the visualization of information gathered by means of merging instances from two modalities. Without them, registration would be worthless for our application. Another important aspect of DRRs, the interpretation and adaption of grey-values in order to imitate a real radiograph, will be examined in chapter 5. Also other visualization aspects occurring in the context of X-ray based medical imaging will also be covered there, since they are closely related. They are needed for an intuitive representation of the merged model. So far, the foundations for image registration will have been laid and the actual registration may be performed. Our approach will be described in chapter 6. A validation of the techniques is also included there. Finally, chapter 7 will give an impression of what has been achieved and what still can be done in order to improve the quality and stability of CANP.

3 C-Arm Calibration

The C-arm is a portable X-ray imaging device often used in the operating theater, especially in the case of endoscopic treatment, where the surgeon cannot see inside the body by other means. Instead of X-ray film material, the C-arm captures a digital image, which is then displayed on its console, as can be seen in figure 1.7, although the principles are very similar. Radiation comes from an X-ray source, which is assumed to be a point source, passes the human body more or less attenuated and reaches the curved X-ray detector. Electrons are emitted, where X-ray photons hit the surface. Those electrons are bundled in the electron optics, steered by magnetic fields, and finally reach a phosphor layer in the output window, where they are converted to visible light. This image in then captured by a CCD[1] camera. The whole setup is drafted in figure 3.1. In section 3.1, a camera model is described, which

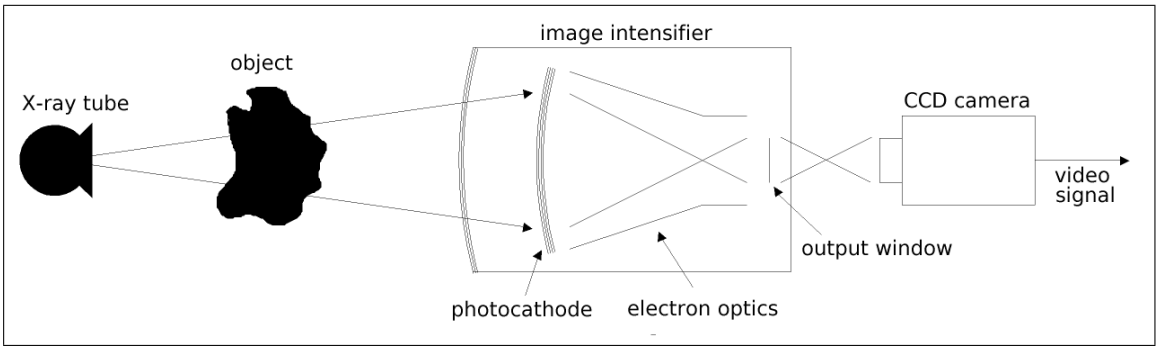


Figure 3.1: Principal setup of C-arm [25]

is principally able to emulate the properties of the C-arm camera. Since this model is a linear mapping, it cannot consider nonlinear image distortion. The reason for such effects and also their treatment will be covered in section 3.2. Subsequently, two algorithms shall be presented in section 3.3 for estimating the parameters of the derived model. Calibration error and parameter optimization are important issues for such algorithms and shall be deepened in the adjacent sections 3.4 and 3.5. Results from the latter issue are also relevant for the automatic registration described in chapter 6, where the parameters of a rigid transformation have to be estimated which minimizes differences between radiograph and DRR. Finally, our implementation of camera calibration shall be presented in section 3.6.

[1] Charged Coupled Device

3.1 A Mathematical Camera Model

Before the C-arm camera can be calibrated, we need to understand how cameras principally work. Therefore, a mathematical model shall be described which covers common cameras but which may also be applied to the C-arm geometry, as we will see. In principal, cameras are linear devices, hence describable by a geometry of points and lines.

The process of camera calibration includes the computation of *intrinsic* and *extrinsic* camera parameters. The former mainly consist of the *focal length*, but also some other constants belong to this category. The latter, however, describe the position and orientation — also called *pose* — of a camera in a reference coordinate system, e.g. the world coordinate system. The intrinsic parameters, once determined, normally remain the same, whereas the extrinsic parameters may change between the acquisition of different images.

Intrinsic and extrinsic parameters are combined in a 3×4 projection matrix, which describes the transformation of a given 3D point[2] X of a scene in reference coordinates to a 2D point x on the camera's image plane in image coordinates. Thereby, points in both coordinate systems are expressed by homogeneous coordinates.

$$x = PX \tag{3.1}$$

The projection matrix P has 11 degrees of freedom, up to overall scaling. Its derivation has been described e.g. by Zisserman and Faugeras and the main ideas shall be described in the following [30, 17]. Projections can be carried out in different ways. In particular, *orthographic projections* or *parallel projections* with a camera center lying on the plane at infinity and *perspective projections* with a camera center in finite space have to be distinguished. A camera center lying at infinity resembles the fact that parallel lines intersect in this point. As stated in the last chapter, the C-arm shall be calibrated using a perspective projection, since this approximates its physical properties best. Therefore, the orthographic case, which as a matter of fact is a simplification, will not be covered here.

The *extrinsic* camera parameters of a perspective camera are described by a rotation and a subsequent translation. They describe the transformation of any given point in reference coordinates into camera coordinates. The translation is represented by a homogeneous vector $t \in \mathbb{R}^4$, thus three degrees of freedom, and the rotation by a matrix $R \in \mathbb{R}^{4\times4}$. R can be computed either from three *Euler angles*, a *unit quaternion* or from one single rotation by a certain angle about an arbitrary axis [63], thus additional three degrees of freedom. Rotation matrices are always *orthonormal*, i.e. their inverse is the transposed matrix ($RR^T = I$) and their determinant is one ($det(R) = 1$).

With the center of the camera C and 3D points X in reference coordinates, the transformation to the corresponding 3D points X_{cam} in camera coordinates can be set up.

$$X_{cam} = R(X - C) \tag{3.2}$$

The camera center C may also be expressed implicitly which yields the transformation

$$X_{cam} = RX + t, \tag{3.3}$$

[2] In the following, 3D points will be denoted by an upper-case letter in order to distinguish them from corresponding 2D points.

where $t = -RC$. After this transformation has been applied to the 3D points in world coordinates that shall be imaged by the camera, those are available in the camera coordinate system with the camera center as its origin and the camera's viewing direction as an axis, commonly the z-axis. The actual projection to the image plane now only depends on the camera's intrinsic parameters, as we will see. The most elementary form of a perspective

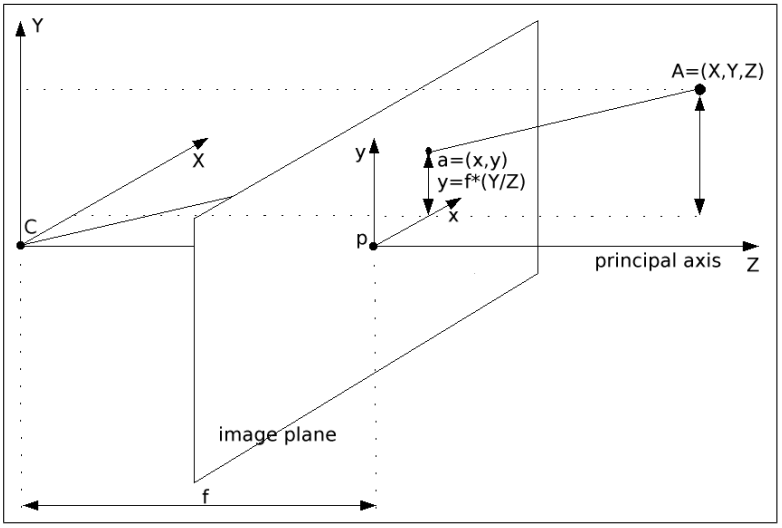


Figure 3.2: Pinhole camera geometry with camera center C, principal point p and focal length f. The coordinates of an arbitrary point a on the image plane can be derived directly from the corresponding 3D point A using the theorem of intersecting lines.

projection is a central projection. It models a mapping along the z-axis for 3D points $X_{cam} \in \mathbb{R}^3$, here in inhomogeneous camera coordinates, to 2D points $x \in \mathbb{R}^2$ as follows:

$$(x_{cam_1}, x_{cam_2}, x_{cam_3})^T \mapsto (x_{cam_1}/x_{cam_3}, x_{cam_2}/x_{cam_3})^T \tag{3.4}$$

The same may be expressed also in homogeneous coordinates with 3D points $X_{cam} \in \mathbb{R}^4$ and 2D points $x \in \mathbb{R}^3$ respectively.

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x_{cam_1} \\ x_{cam_2} \\ x_{cam_3} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_{cam_1} \\ x_{cam_2} \\ x_{cam_3} \\ 1 \end{pmatrix} \tag{3.5}$$

Since the resulting vector is in homogeneous coordinates now, the division by x_{cam_3} will be performed implicitly, as soon as the vector is finally mapped to inhomogeneous coordinates again. To understand this, we have to remember that a homogeneous vector $(v_1, v_2, v_3)^T \in \mathbb{R}^3$, representing a 2D point $(v_1/v_3, v_2/v_3)^T$, actually is a line in $\mathbb{R}^3$, defined by different scaling factors v_3. The corresponding 2D point is located on the plane $v_3 = 1$ in $\mathbb{R}^3$, exactly where those line intersects the plane. This illustrates, why, with homogeneous coordinates, the projection is performed implicitly. Actually, homogeneous coordinates define a perspective projection with the origin as its projection center. So far, no scaling or *zoom* has been considered. In fact, it has been defined to one, following from the definition of

homogeneous coordinates. Therefore, the *focal length* is included in the model which yields a projection P'

$$(x_{cam_1}, x_{cam_2}, x_{cam_3})^T \mapsto (f \cdot x_{cam_1}/x_{cam_3}, f \cdot x_{cam_2}/x_{cam_3})^T \tag{3.6}$$

in inhomogeneous coordinates. Homogeneously, it is expressed by the following matrix:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \underbrace{\begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{P'} \begin{pmatrix} x_{cam_1} \\ x_{cam_2} \\ x_{cam_3} \\ 1 \end{pmatrix} \tag{3.7}$$

This projection describes the geometry of a *pinhole camera* which is applicable also to other cameras, the lack of nonlinear distortion presumed. The origin of the image coordinate system is located in the center of the image, exactly where the z-axis of the camera coordinate system intersects the image plane. It is called the *principal point* in the camera's geometry. Normally, however, the origin is assumed to be in one of the corners of the image. This additional requirement is modelled by an offset, which accomplishes the translation of the principal point to the desired origin and is expressed by two additional constants p_x, p_y in our projection matrix. In the following, only the homogeneous notation will be used.

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \underbrace{\begin{pmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{P'} \begin{pmatrix} x_{cam_1} \\ x_{cam_2} \\ x_{cam_3} \\ 1 \end{pmatrix} = \begin{pmatrix} f x_{cam_1} + p_x x_{cam_3} \\ f x_{cam_2} + p_y x_{cam_3} \\ x_{cam_3} \end{pmatrix} \tag{3.8}$$

Until now, image coordinates have been defined by real numbers. These are not very useful for the representation of discrete pixel coordinates which are commonly used, e.g. by CCD cameras, for the representation of 2D images. Pixels are not necessarily quadratic. Thus, *pixel pitch* has to be represented separately for the x and y directions. Let m_x be the number of pixels in x-direction and m_y the number of pixels in y-direction, the projection matrix P' can be written as

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \underbrace{\begin{pmatrix} \alpha_x & 0 & c_0 & 0 \\ 0 & \alpha_y & c_1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{P'} \begin{pmatrix} x_{cam_1} \\ x_{cam_2} \\ x_{cam_3} \\ 1 \end{pmatrix}, \tag{3.9}$$

where $\alpha_x = f \cdot m_x$, $\alpha_y = f \cdot m_y$, $c_0 = p_x \cdot m_x$ and $c_1 = p_y \cdot m_y$. Certainly, the resulting coordinates have to be rounded to discrete values. P' may also be written as a 3×3 matrix K' and an additional fourth column. Thus,

$$P' = [K'|0] = K'[I|0], \tag{3.10}$$

with I the 3×3 identity matrix. The equation is extended by the transformation into the camera coordinate system $X \rightarrow X_{cam}$ described above and we obtain the 3×4 matrix P containing now all extrinsic and intrinsic parameters of the projection

$$P = K[R|t], \tag{3.11}$$

where $t = -RC$ and C the center of the camera in reference coordinates. K is a 3×3 matrix encapsulating four intrinsic parameters.

$$K = \begin{pmatrix} \alpha_x & 0 & c_0 \\ 0 & \alpha_y & c_1 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.12}$$

Together with six extrinsic parameters, as already stated above, this yields an overall number of ten parameters corresponding to ten degrees of freedom.

Sometimes, a further intrinsic parameter is used, the *skew factor* s. It describes a shearing in the case of non-square pixels arising from non-perpendicular x and y axes.

$$K = \begin{pmatrix} \alpha_x & s & c_0 \\ 0 & \alpha_y & c_1 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.13}$$

However, this most general case of a *finite camera*, modelling eleven degrees of freedom, is normally not needed.

3.2 Modelling Image Distortion

Normally, cameras are not ideal linear devices that may be fully described by the theory introduced above. In fact, images will be distorted due to physical phenomena occurring in the optical system. This also holds for the C-arm which internally combines electron optics in the image intensifier with a CCD camera containing optical lenses. Different kinds of distortion are distinguished, radial, spiral and pincushion (barrel) distortion. In order to achieve an appropriate description of the projection performed by a certain camera, it is often not sufficient to determine only the extrinsic and intrinsic parameters. Image distortion should also be considered, thus, mathematical models for the different types of distortion are needed. Pixel positions may then be corrected, whereby the grey level of a corrected pixel is derived by interpolation between the intensity values of pixels in the neighborhood of the distorted pixel.[68].

3.2.1 Radial Distortion

A great deal of nonlinearities is introduced by *radial lens distortion* which occurs in every optical lens system. However, it is normally higher in combination with a short focal length of the lens system[3], as it is found especially in cheap devices. It is defined as deviation of the ideal (undistorted) point $(\tilde{x}, \tilde{y})$, as it would have been captured by the corresponding pinhole camera, from the actual (distorted) image point (x_d, y_d). This is expressed in a function $L(\tilde{r})$.

$$\begin{pmatrix} x_d \\ y_d \end{pmatrix} = L(\tilde{r}) \begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix}, \tag{3.14}$$

[3]This shall not be confused with the focal length of the corresponding pinhole geometry. It normally differs considerably.

where $L(\tilde{r})$ is the distortion function, only dependent on the radial distance from the center of the image $\tilde{r} = \sqrt{\tilde{x}^2 + \tilde{y}^2}$. A correction could be formulated as follows:

$$\hat{x} = x_c + L(r)(x - x_c) \qquad \hat{y} = y_c + L(r)(y - y_c), \tag{3.15}$$

with a center (x_c, y_c) of the image, which is often identified with the principal point, though they do not necessarily coincide, undistorted (corrected) image pixels $(\hat{x}, \hat{y})$ and distorted (uncorrected) image pixels (x, y). r can be written as $\sqrt{(x - x_c)^2 + (y - y_c)^2}$.

$L(r)$ may now be formulated using a Taylor expansion $L(r) = 1 + \kappa_1 r + \kappa_2 r^2 + \ldots$. A correction for radial lens distortion means the determination of the coefficients $\{\kappa_1, \kappa_2, \ldots\}$ which can be considered as additional intrinsic camera parameters. This is accomplished by iteratively minimizing a cost function, e.g. geometric error (see 3.3.1), measuring the deviation of corrected points from a perfect linear mapping. Algorithms suitable for minimizing an error will be outlined in 3.5.

3.2.2 Pincushion and Spiral Distortion

Pincushion distortion occurs in combination of curved "image planes" with flat image planes. The resulting image is then either convex or concave. This error is unspecific for common camera systems. The C-arm, however, contains a curved X-ray detector which in turn emits electrons towards a flat phosphor layer. For a correction of the resulting error, the radius of the curved screen and also other geometric information, like, in the case of C-arms, the distance from X-ray source to image intensifier have to be considered [65].

Spiral distortion (S-distortion) e.g. emerges as a consequence of external magnetic fields interfering with the internal magnetic fields used for focusing of electron radiation within the image intensifier unit of the C-arm. External magnetic fields may either be created by surrounding devices or have its origin in the geomagnetic field. As a consequence of spiral distortion, straight lines have the shape of an "S"in the image as can be seen in figure 3.3. This type of distortion represents a great challenge for C-arm calibration, since it cannot be corrected for using parameters calculated once, but rather changes whenever the C-arm orientation is altered. Therefore, a reasonable approach is the usage of a calibration device which is attached permanently near to the receptor plate of the C-arm [64]. According to the calibration points in the image, resulting from this construction, distortion coefficients can be calculated anew after each image acquisition. Both types, pincushion and spiral distortion, may also be combined in one single correction step [69].

3.3 Camera Calibration Algorithms

In this section, two well-known camera algorithms shall be presented, the gold standard algorithm and an alternative implementation proposed by Tsai, actually for calibrating CCD cameras [72]. Both will be used in our implementation.

3.3.1 A Gold Standard Algorithm

The *Gold Standard* algorithm represents the standard solution of the calibration problem and other algorithms are often compared to this technique. The following ideas are depicted from Zisserman [30]. Initially, a set of point correspondences is given between 3D points X_i and 2D points x_i, respectively. The projection matrix P in equation 3.11 is calculated via a *direct linear transformation* (DLT) [2, 73]. The relationship between corresponding points if given by

$$x_i = PX_i. \tag{3.16}$$

Application of the cross product yields

$$x_i \times PX_i = 0. \tag{3.17}$$

We assume $x_i = \begin{pmatrix} x_{i_1} \\ x_{i_2} \\ x_{i_3} \end{pmatrix}$ in homogeneous coordinates. P may be written as $p = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix}$, with p_i^T the rows of P. Thus, equation 3.17 can be rewritten as

$$\underbrace{\begin{pmatrix} 0^T & -x_{i_3}X_i^T & x_{i_2}X_i^T \\ x_{i_3}X_i^T & 0^T & -x_{i_1}X_i^T \\ -x_{i_2}X_i^T & x_{i_1}X_i^T & 0^T \end{pmatrix}}_{A'} \begin{pmatrix} p^1 \\ p^2 \\ p^3 \end{pmatrix} = 0. \tag{3.18}$$

The rows of A' are linearly dependent, therefore one of them is rejected. This results in a 2×12 matrix A_i for each given point correspondence i. The A_i for all n correspondences are stacked up now, which results in matrix

$$A = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{pmatrix}. \tag{3.19}$$

Thus, the calibration problem has been reduced to dissolving the homogeneous linear equation system

$$Ap = 0. \tag{3.20}$$

A solution of equation 3.20 is provided by the *singular value decomposition* (SVD) [7, 30, 61]. At least $5\frac{1}{2}$ point correspondences are needed for a solution of the system since each point correspondence is a constraint for two degrees of freedom and P has eleven degrees of freedom. However, normally by far more points are used for calibration in order to minimize the influence of noise. Thus, the linear equation system is over-determined and the best solution for P can be found by minimizing a certain error function. One possibility is minimization of the *algebraic error* defined by the norm $\|Ap\|$, subject to a normalization constraint $\|p\| = 1$ or $\|\hat{p}^3\| = 1$ with $\hat{p}^3 = (p_{3,1}, p_{3,2}, p_{3,3})$ the first three entries in the last row of P. Another choice is the *geometric error* $\sum_i d(x_i, PX_i)^2$, where $d(x, y)$ is the Euclidean distance between x and y. This actually is a maximum likelihood estimation of P assuming Gaussian noise in acquisition of 2D points corresponding to exact 3D points, as we will see later on in 3.4. In

the case of also inexact 3D points, the error is defined more generally by a weighted sum of world and image errors, as given by the *Mahalanobis distance*. Determine

$$\min_P \sum_i d_{Mah}(x_i, P\hat{X}_i)^2 + d_{Mah}(X_i, \hat{X}_i)^2,$$

where $\hat{X}_i$ are the estimated 3D points, X_i the exact 3D points and $d_{Mah}(x, y)$ the Mahalanobis distance [30]. Optimization schemes will be discussed later in section 3.5.

3D points X_i and 2D points x_i should be normalized before the actual calculations in order to avoid numerical instability. This is achieved by similarity transformations U and T, which translate the cloud of points with its centroid to the corresponding coordinate system origin and apply a normalization to an average distance of one from the origin. Certainly, this normalization has to be undone afterwards. In summary, the Gold Standard algorithm proceeds as follows:

1. Normalize the system, thus $\tilde{P} = TPU^{-1}$
2. Setup equation 3.20 via DLT.
3. Calculate an initial solution for $\tilde{P}_{init}$ using SVD.
4. Optimize the initial solution $\tilde{P}_{init}$ iteratively by minimization of the chosen error function and obtain $\tilde{P}$
5. Undo normalization step, thus $P = T^{-1}\tilde{P}U$
6. Split up the solution P into a camera calibration matrix K, a rotation R and a camera center C by using equation $P = [M| - MC] = K[R| - RC]$ and a simple RQ-decomposition.

Distortion is not included within this algorithm and has to be considered separately, using the techniques described above.

3.3.2 Tsai's Approach

A slightly different approach has been proposed by Tsai in the context of CCD cameras [72]. It suggests an *image formation pipeline* which, starting from points in world coordinates (X_w, Y_w, Z_w), calculates their corresponding points in image coordinates step by step. First, a rotation R and a translation t are calculated as a transformation from world to camera coordinates.

$$(X_w, Y_w, Z_w) \rightarrow (X, Y, Z) \tag{3.21}$$

Then, an idealized projection from 3D to 2D is carried out, using the initial guess of focal length f.

$$(X, Y, Z) \rightarrow (x_u, y_u) \tag{3.22}$$

Radial lens distortion modelling only the first two coefficients κ_1 and κ_2 of the Taylor expansion described above results in the true image coordinates.

$$(x_u, y_u) \rightarrow (x_d, y_d) \tag{3.23}$$

TV scanning and sampling using the given principal point (p_x, p_y) and a scaling factor s_x yields the image pixel coordinates.

$$(x_d, y_d) \rightarrow (x_f, y_f) \tag{3.24}$$

At least, point correspondences $\{X_i \leftrightarrow x_i\}$, the pixel pitch and the number of pixels in the frame grabber's x-direction have to be given. The algorithm computes all the parameters used in the image formation steps in two stages.

- In stage one, the rotation R and the translation t is computed. Tsai describes a *coplanar* and a *non-coplanar* approach. In the former one, all 3D points are located in a plane, thus their z-coordinate is zero in the object coordinate system. In this case, only the first two entries of t, t_x and t_y are calculated and the scale factor s_x is not computed but assumed to be given as a parameter, in contrast to the latter approach. From the distorted image coordinates (x_d, y_d), five mixed terms $t_x^{-1}r_1, t_y^{-1}r_2, t_y^{-1}t_x, t_y^{-1}r_4, t_y^{-1}r_5$ are computed first. The actual rotation R and translations t_x, t_y are then calculated from the mixed terms.
- In stage two, the effective focal length f, the distortion coefficients (κ_1, κ_2) and the third entry of t, t_z are computed via an intermediate step, for which distortion is still ignored.

In contrast to the Gold Standard Algorithm, Tsai's approach considers also radial lens distortion. For optimization of the over-determined system, a Levenberg-Marquardt (see 3.5) iteration is used. For an extensive explanation of the two calibration steps, please refer to [72].

3.4 Parameter Optimization and Calibration Error

Different influences normally make an exact solution for the calibration matrix P impossible. The most obvious is that the problem is over-determined because more point correspondences are used for estimation of P, than this matrix has degrees of freedom. Furthermore, measured points are at least disturbed by Gaussian noise. Often, this assumption is too optimistic and robust approaches are necessary in order to detect outliers in the given point correspondences. The ideas in the following are depicted from [61, 30].

In the case of Gaussian noise, a *sum of squares* approach is used for optimizing the parameters of the model, in this case the degrees of freedom of calibration matrix P. In abstract terms, this is expressed by

$$\sum_{i=1}^{n} \| \vec{q}_i - f\left(\vec{p}_i; \vec{a}\right) \|_2^2, \tag{3.25}$$

where f is a certain model which is parameterized by $\vec{a} \in \mathbb{R}^m$, and $\vec{p}_i, \vec{q}_i$ are some measured points that shall be functionally related by the model. Therefore, parameters a_j have to be found, which approximate this relationship. The basic principle behind equation 3.25 is a *maximum likelihood estimation*, thus the estimation of a model that most likely yielded the given point correspondences under the assumption of Gaussian error in the distances

between measured and calculated points. The probability of a distinct model k, given a set of point correspondences and an assumed measurement distribution σ around the actual position, is given by

$$P_{\vec{a},k} = \prod_{i=1}^{n} \left(\exp\left(-\frac{1}{2} \frac{\|\vec{q}_i - f\left(\vec{p}_i; \vec{a}\right)\|_2^2}{\sigma^2} \right) \Delta q \right), \tag{3.26}$$

with a weighting constant Δq. It can easily be shown that minimizing equation 3.26 is equivalent to minimizing equation 3.25, hence, the *least-squares* approach is correct under the assumption of Gaussian error. By using many point correspondences, *statistical errors* are averaged out. This does not hold for *systematic errors*, also called *bias error*, caused by either wrong assumptions about the underlying model or systematic error in the measurements. Some optimization algorithms suitable for this minimization problem will be mentioned in chapter 3.5.

In the case of 2D-3D point correspondences necessary for camera calibration, error in terms of Euclidean distances may principally be measured either in 2D or 3D space. Both techniques are described in the following, though for calibration with exact 3D points and measured 2D points, it is meaningful to measure distances on the image plane by performing a forward projection. By this, Gaussian error is assumed for measurements on the image plane, whereas the 3D points are assumed to be correct. This will be described next in 3.4.1. The inverse case of a back-projection from 2D to 3D space will be considered in 3.4.2. Though it is not useful for estimation of the camera model in our case, it will be very important for the extraction of information from the radiograph in order to merge it with the three-dimensional CT model and also for supplementary quality tests.

3.4.1 Forward Projection of 3D Points

In the following, we assume that 3D points $\hat{X}$ are projected to 2D points $\hat{x}$ on the image plane, using a somehow determined projection matrix P. In this case, f in equation 3.25 corresponds to our projection matrix P, $\vec{p}$ is associated with 3D points $\hat{X}$ and $\vec{q}$ with 2D points $\hat{x}$, respectively.

$$\hat{x} = P\hat{X}. \tag{3.27}$$

This is called *forward projection*, since P is used according to its natural purpose. Again, the $\hat{X}$ are assumed to be accurate, in contradiction to the measured 2D points. The Euclidean distances between measured points x and predicted points $\hat{x}$ can be used either for finding the optimal parameterization of P by minimizing equation 3.25 or as a measure of goodness for a given matrix P. For the latter purpose, either a mean distance d_{mean} or a worst-case distance d_{max} between the projected points $\hat{x}$ and the measured points x is calculated

$$d_{mean} = \frac{1}{n} \cdot \sum_{i=1}^{n} \|\hat{x}_i - x_i\| \qquad d_{max} = max_{i=1}^{n} \|\hat{x}_i - x_i\|, \tag{3.28}$$

with n the number of given point correspondences. Different types of image distortion could be considered here and corrected points $\tilde{x}$ would be used instead of $\hat{x}$. This requires, however, an already existing parameterized distortion model, which is commonly not available during the calibration procedure but is subject to optimization itself. They are often assumed to be part of the model parameters $\vec{a}$.

3.4.2 Back-Projection of Points to Rays

Starting with a 2D point x, the projection matrix P may also be used in order to calculate the *back-projection* in form of a line or ray in 3D space, passing through the camera center on the one hand and through the pixel x on the image plane on the other hand. Thereby, at least two points in 3D space have to be known, which both project on this pixel. The line can be derived as follows. If P is written in the form $[M|p_4]$, the camera center is given by $C = -M^{-1}p_4$. It represents the first point. As the second point, the intersection of the ray and the plane at infinity can be used. This point is given by $D = ((M^{-1}x)^T, 0)$, where x is the 2D point for which the ray shall be calculated. This yields the parameterized form of the line equation

$$l(\lambda) = \begin{pmatrix} -M^{-1}p_4 \\ 1 \end{pmatrix} + \lambda \begin{pmatrix} M^{-1}x \\ 0 \end{pmatrix} = \begin{pmatrix} M^{-1}(\lambda x - p_4) \\ 1 \end{pmatrix} = t + \lambda v \tag{3.29}$$

This back-projection may be used for finding corresponding points in world coordinates for a given point in image coordinates. Furthermore, it serves the purpose of estimating calibration error. The ray for a certain 2D point x should intersect all 3D points X_i which fulfill $x = PX_i$. Due to calibration error, the ray will not intersect X_i exactly but pass by in some distance d which may be calculated by an orthogonal projection of X_i onto the ray. This is accomplished by defining a plane including X_i with the line's direction vector as its normal.Let l be the parameterized line, X_i the exact 3D point and E the plane. Thus,

$$l(\lambda) = t + \lambda v = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} + \lambda \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}, \qquad X_i = \begin{pmatrix} X_{i_1} \\ X_{i_2} \\ X_{i_3} \end{pmatrix} \tag{3.30}$$

and

$$E: \; v_1(x_1 - X_{i_1}) + v_2(x_2 - X_{i_2}) + v_3(x_3 - X_{i_3}) = 0. \tag{3.31}$$

Now, E and l are intersected by putting l in E.

$$\begin{aligned} v_1(t_1 + \lambda v_1 - X_{i_1}) + v_2(t_2 + \lambda v_2 - X_{i_2}) + v_3(t_3 + \lambda v_3 - X_{i_3}) &= 0 \\ v_1t_1 + v_2t_2 + v_3t_3 - (v_1X_{i_1} + v_2X_{i_2} + v_3X_{i_3}) + \lambda(v_1^2 + v_2^2 + v_3^2) &= 0 \\ v \circ t - v \circ X_i + \lambda\|v\|^2 &= 0 \\ \lambda = \frac{v \circ X_i - v \circ t}{\|v\|^2}, & \end{aligned}$$

where $x \circ y$ is the dot product of x and y. Now, the orthogonally projected point is obtained by simply putting λ in l.
Mean and worst-case errors are obtained as in equation 3.28, but now, back-projected 3D points are used instead of 2D points obtained by forward projection.

$$d_{mean} = \frac{1}{n} \cdot \sum_{i=1}^{n} \|\hat{X}_i - X_i\| \qquad d_{max} = max_{i=1}^{n}\|\hat{X}_i - X_i\|, \tag{3.32}$$

with n the number of distinct 2D points or rays. In the case of the C-arm, one may assume larger errors using the forward projection, since the given 3D points lay in between the camera center and the image plane, which should lead to larger deviations on the image plane.

3.5 Parameter Optimization Algorithms

Given a single error function f, the objective of optimization is minimizing[4] f with respect to some independent variables. Those may either be the parameters $\vec{a}$ of a certain model in order to fit given data, as in equation 3.25, or arbitrary variables, if there is no model. The problem of finding extrinsic parameters so that the DRR matches a given radiograph best, is of the latter type. In this case, there does not even exist a closed form of f[5], though it can be evaluated at random points in parameter space. Multidimensional systems of nonlinear equations $\vec{f}$ can be minimized by collapsing the system to a master function f, e.g. by the sum of squares of the individual functions f_i. Anyway, the resulting function $f : \mathbb{R}^n \to \mathbb{R}$ may be thought of as an *error surface* in $\mathbb{R}^n$ that has a more or less smooth behavior. In the general case of a highly nonlinear function, there is no simple solution for the optimization problem and iteration is the only possibility to find one at all. The following ideas are mostly depicted from [61, 7, 30].

Finding the global minimum, which is commonly referred to as a *global optimization problem*, is generally very difficult, since every local minimum deceives the algorithm on its way to the global one. A vast array of heuristics is available to solve this problem, lots of them only suited for a certain kind of error behavior. They follow a characteristic sequence of points in search space and, by means of that sequence, may be roughly categorized in two groups. Representatives of the first one start minimization from different points in space, maybe chosen randomly, and thereby try to find all minima. Once they are discovered, the global one can be evaluated. Members of the second group start at only one distinct point but try to detect other minima by restarting iteration at a somehow displaced position on the error surface with respect to the minima that has just been detected. Such techniques are known as *simulated annealing* methods. In any case, global optimization is a very expensive task.

Therefore, *local optimization* is often used. Starting with an initial guess, such algorithms converge to the next minimum, hoping that it is the desired and not an unexpected local one. Thus, a good initial guess is needed for a satisfactory result, where "good" depends on the smoothness of the corresponding surface. Some methods only rely on function evaluations, like e.g. the *downhill simplex method* which for an n-dimensional problem stores error information in the vertices of a simplex with $n+1$ vertices. According to the current information, the simplex is warped by one of four simple geometric constructions, (1) *reflecting* the vertex with highest error by the plane spanned by the other n vertices, (2) *expansion* in direction of lowest error, (3) *contraction* in inverse direction of highest error and (4) *multiple contraction* towards lowest error [61]. Even simpler is the *best neighbor* or *hill climbing*[6] technique which does not remember already evaluated points. In the neighborhood of the current iteration point, error is sampled at a certain distance in each dimension of the search space, both in upper and lower directions. If a point with lower error is found, is will become the new point for the next iteration step. Otherwise, the distance of neighbored points is decreased and error is reevaluated. Henceforth, the iteration runs with smaller distances, until no better point can be found. Iteration is aborted, when the step-widths have become so

[4] The problem could also be formulated as a maximization problem.

[5] In fact, it is a combination of the volume renderer and the quality measure and can be assumed to be highly nonlinear.

[6] In our definition of optimization, we actually move towards the valley.

small that the desired accuracy is reached. This technique was successfully used for finding a rigid transformation in 2D-3D registration [63, 76]. The method can be modified towards a trade-off between those neighbors that have led to a decreased error during evaluation. The weights can be determined by calculating the contribution of one point to the overall improvement, obtained from summing up improvements of all "good" points. This is called *weighted best neighbor* and succeeded also in 2D-3D rigid registration [58]. While best neighbor guarantees a descent in the sense of finding a better point in each step, the latter method may hop to worse points, since nothing is known about the point determined by weighting the "good neighbors", in advance. However, we can fall back to best neighbor in the case of a worse error.

Other methods also use derivative information in order to speed up convergence by making assumptions about the curvature of the error surface in the neighborhood of the current point. *Gradient descent* evaluates the gradient vector and moves a certain distance in its negative direction. One-dimensional line minimization can be performed in order to find the minimum in the negative direction of the gradient. This results in the *steepest descent* method, which is very stable but considered to be inefficient, since in every point, it moves in an orthogonal direction with respect to the last gradient vector, because there is nothing more to minimize in the former direction. This results in slow convergence [7, 61]. However, the basic principle of using conjugate directions is not that bad. The *conjugate gradient* methods benefit from constructing an orthogonal direction using gradient information belonging to the current point and also all previous directions, hence, former minimizations are not spoiled by line-minimization along the new direction. Assuming a perfect parabolic bowl, the minimum can be reached in n iterations, where each of them comes along with one line minimization. This one-dimensional minimization is performed either by *golden section search* or e.g. using second order information as in *Brent's* method [61]. Hybrid methods as *Powell's direction-set* gets by without multi-dimensional derivatives, however, is based on Brent's line minimization technique. It also benefits from a set of mostly conjugate directions. This algorithm has also been used in 2D-3D rigid registration [8].

Newton iteration is probably the most prominent representative of techniques using derivatives. In the current point, it assumes the surface to be approximated by a parabolic bowl. Locally, this assumption leads to quadratic convergence, but globally often results in strange behavior. Therefore, it is often combined with other techniques. *Levenberg-Marquardt* iteration for example switches seamlessly between Newton iteration and a more stable gradient descent, depending on the outcome of error evaluation at a certain point. This combination has become state-of-the-art for least-squares problems [30]. However, the derivatives can only be evaluated, if the function itself is known. Otherwise, they can be approximated by the difference quotient, although correct implementation heavily depends on the kind of error surface. Two points are needed for calculation of the difference in error. If they are located too close at each other, there may be no more difference in error, at least at machine precision, if they are displaced too far, the resulting gradient is a bad approximation for the actual gradient in the current point [61].

Techniques making extensive use of derivative information are not suitable for CANP. The error function is unknown and the calculation of lots of approximations for first or even second order information is not affordable, since every evaluation of the function requires the computation of a DRR, which is rather expensive. This also excludes global optimization schemes. A good initial guess can be assumed for our registration problem anyway. A

comparison between best neighbor, Powell's method and gradient descent with respect to a 2D-3D registration problem can be found in [76]. Since the best neighbor methods had been proved to be suitable in the context of 2D-3D rigid registration, they were chosen for implementation in CANP. Furthermore, weighted best neighbor was also able to improve the calibration of the C-arm, as we will see in 3.6.1. Implementation details can be found in 3.6.2.

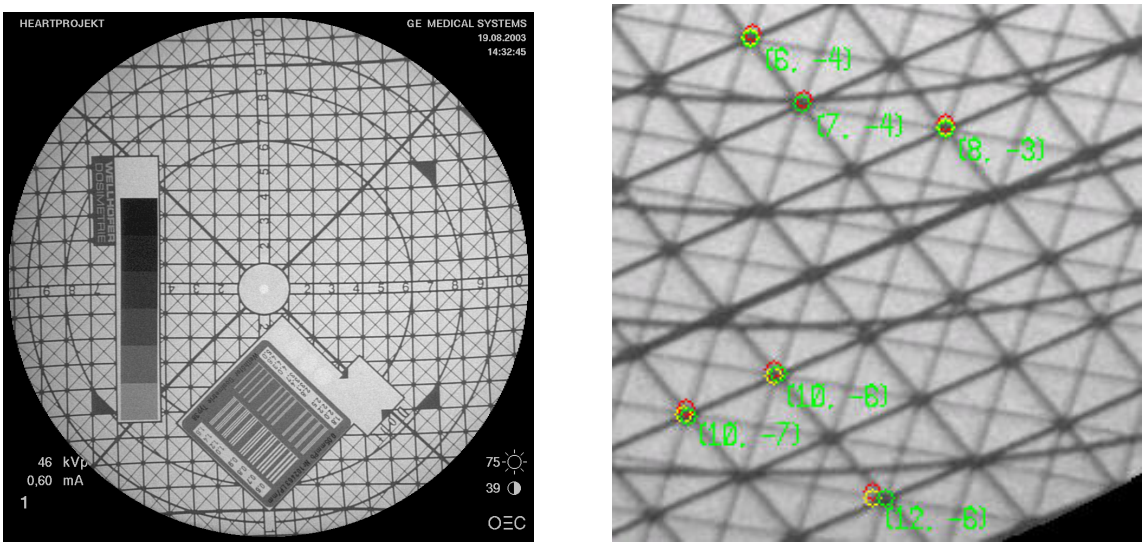
3.6 Implementation

For the implementation of camera calibration, at first, a suitable algorithm is required. Two famous candidates have already been described in section 3.3. Both have been used in our implementation, outlined in 3.6.1. Their results can be further improved by optimizing the model's parameters, which means minimizing calibration error. For this purpose, and also for finding a rigid transformation between CT model and patient, as described in chapter 6, two parameter optimization algorithms have been implemented. A detailed description is given in 3.6.2.

3.6.1 Camera Calibration

For calibration of the C-arm, the camera calibration framework `libcc` has been designed and partially implemented. Internally, it is based on two different algorithms (see section 3.3), the gold standard algorithm and also that described by Tsai. Concretely, the former is included as a part of the OPENCV library which has been provided by INTEL and is now open source [46]. Tsai's algorithm is also freely available in the Internet [80]. A more detailed treatise on `libcc` and the integration of those algorithms is given by Groher, along with some quality tests [25].

In summary, we took 15 images of a calibration plate (see figure 3.3) normally used for dosimetry. Therefore, we have a coplanar calibration approach with all 3D points lying on the plane defined by the plate, and the z-axis in its object coordinate system is always zero. Corresponding 2D points were obtained by measuring pixel coordinates in the fluoroscopic images. Since in the C-arm, the image plane is far behind the object, sub-pixel accuracy was not necessary. The best results have been achieved with Tsai's approach in combination with images, where the normal of the plate was inclined about 30 degrees to the viewing direction of the C-arm and where the full plate was visible. This shows that depth-information is necessary for an accurate estimation of a perspective projection. Images obtained by a pose of the plate near to the X-ray source, thus showing only a part of the plate, were quite useless. The most reasonable result could be further improved by application of the weighted best neighbor parameter optimization algorithm described in 3.5. The differences between the immediate and the optimized solution are depicted in figure 3.3. In particular, the calibration yielded a focal length of $f = 4375.21mm$ and a principal point $(p_x, p_y) = (514.207, 490.170)$. A back-projection was calculated also for all other images, using these values, and the mean distance between measured and back-projected 3D points is $0.5mm$. Image distortion has not been considered at all in our approach.



(a) (b)

Figure 3.3: Calibration images, (a) Dosimetry calibration plate used for coplanar calibration of C-arm, vertices of the grid are located at intervals of one centimeter (b) Section of calibration plate with plotted point correspondences for verification. The green circles depict measured 2D points, red circles show 2D points obtained by forward projection of the known 3D points using the projection obtained by Tsai's algorithm. Yellow circles denote forward-projected 2D points with an optimized projection matrix. The z-coordinate is always zero and therefore not plotted.

3.6.2 Optimization Algorithm

Two different approaches have been implemented in the camera calibration framework `libcc` for parameter optimization, the best neighbor and the weighted best neighbor optimizer which have already been described in 3.5 [25]. An abstract class `CcOptimization` defines the full interface for integration of the optimization algorithms. The algorithms itself are implemented in derived classes `CcBestNeighbor` and `CcWeightedBestNeighbor`. Other algorithms may be added since the interface is as common as possible. It shall be described first.

`CcOptimization` describes an iterative parameter optimization in an arbitrary search space whose dimension is initialized by `initDimension()`. An array of float values containing the initial parameters for optimization is passed by calling `initSearchSpace()`. Furthermore, a callback cost function `eval()` has to be given by `initCostFunction()`. It is called with some parameter vector specified by the algorithm and shall return the error in a float value within the interval $[0..1]$ with 0 depicting no error at all. It is very essential since the algorithm will try to find a vector which minimizes this error.

The optimization is performed in at least two nested loops, the *main optimization loop* in method `mol()` which is already implemented here and the abstract *parameter optimization loop* `pol()`. The optimization itself is started by calling `Optimize()` which in turn enters

the main optimization loop. In each iteration of `mol()`, *delta values* Δp_i will be fetched via the callback function `getDelta()`, which has to be initialized by passing a function pointer to `initRefinement()` before. For each parameter in the search space, a corresponding delta value has to be specified. Then `pol()`, implemented in the derived classes, will be executed exactly once. It uses the given delta values in order to modify the current parameter vector in some manner, each parameter may be refined by addition or subtraction of the corresponding delta value Δp_i to the upper or lower direction. By iterative modifications and evaluations using the callback function `eval()`, the best parameter vector in the sense of minimizing the error is calculated. Now, the new parameter vector becomes the current parameter vector and `pol()` starts over with the next cycle. If no better vector can be found with the given delta values, then `pol()` terminates and `mol` takes over again by fetching new (smaller) delta values via the callback `getDelta()` in order to improve the granularity of the parameter search. Then `pol()` is entered again. The algorithm, thus `mol()` terminates, if `getNextDeltaValues()` returns false which means that the accuracy desired by the application has now been reached.

While the algorithm is running, it may be interrupted by the application via another callback function `control()` which may be setup during initialization via `initControl()`. The manner how a new parameter vector is calculated is subject to the classes inheriting from the `CcOptimization` abstract class. For the two implemented algorithms, best neighbor and weighted best neighbor, this shall be outlined now.

At first, class `CcBestNeighbor` shall be described for it is the simpler one. When the dimension of the search space is initialized by the application, it allocates a vector for *temporary quality measurements* `tqm` containing twice the search space dimension elements. Then, based on the *current parameter vector* `cpv`, a *temporary parameter vector* `tpv` is created by iteratively modifying the parameters — one by one, leaving all others unchanged — either to the upper or to the lower direction. Each `tpv` is evaluated by calling `eval()` and the resulting error is stored in `tqm`. This is why we need a quality vector with twice the number of elements than the search space has parameters. At last, each quality measure from `tqm` is compared to the *current quality measure* `cqm`. The parameter vector, which yielded the best improvement, is recreated (we still know the delta value and the direction of refinement) and becomes the new current parameter vector for the next iteration of `pol()`. If no better than the current parameter vector may be found with the given delta values, however, `pol()` terminates and `mol()` starts over again with fetching new delta values and maybe entering `pol()` again.

The mechanism implemented by class `CcWeightedBestNeighbor` is very similar. The evaluation of the search space by modifying the parameters one by one remains the same. However, the new parameter vector is calculated differently. Provided the temporary parameter vector `tqm` has already been calculated, not the parameter which yielded the best results in `eval()` is selected for ultimate refinement, but all changes which led to a decrease in error, are considered. Thus, each value `tqm[i]` is compared to the current quality `cqm` and all those, which led to an increased error, are set to 0 in `tqm`. For the remaining measurements in `tqm[i]`, the difference Δq_i is calculated between `cqm` and `tqm[i]`. Now, all `tqm[i]` either contain a positive improvement or are zero. They may be interpreted as weights contributing to the sum `totalDif` of all `tqm[i]`. The following loop shows, how a new parameter vector is calculated by linearly weighting the respective delta values Δp_i by $\frac{\Delta q_i}{\sum_{i=1}^{2n} \Delta q_i}$.

```
for ( int i = 0; i < dim; i++ ) {
    // if parameter is marked to be refined in upper direction
    if ( tqm[dim+i] != 0 ) {
        cpv[i] = cpv[i] + ( ( tqm[dim+i] / totalDif ) * cdv[i] );
    }
    // else if parameter is marked to be refined in lower direction
    else if ( tqm[i] != 0 ) {
        cpv[i] = cpv[i] - ( ( tqm[i] / totalDif ) * cdv[i] );
    }
}
```

Subsequently, a test is performed, if the resulting quality of the new vector is really better than those of the current one. Only if this is the case, the new vector will be accepted and become the current parameter vector in the next iteration of `pol()`. Otherwise, the best neighbor solution (which already has been proved to be better) is used as a fall-back solution. As in class `CcBestNeighbor`, `pol()` is aborted when there is no better solution than the current one with the given delta values.

4 Volume Rendering Techniques

The notion *volume rendering* or *volume visualization* in general covers the *representation*, *manipulation* and *illustration* of volume data [11]. A *3D scalar field*, mathematically representing some model, is defined by a function $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ which assigns some scalar value[1] s to each point $\vec{p}$ in 3D space. However, volume data is acquired by either measuring or simulation and is available only at somehow discretized vertices of a mesh. These vertices often have an unnormalized structure in the raw data directly after acquisition. By means of re-sampling by interpolation — e.g. tri-linear or tri-cubic — a *normalized mesh* or *regular grid* may be constructed, consisting of *voxels* which all have the same, cuboid shape. A CT model serves as an example. The raw data consists of 2D axial slices which can have arbitrary distances to each other. A normalization step is necessary in order to transform it to *uniform coordinates*[2]. The finite number of voxels $\vec{v}$ in a regular grid is just an approximation for the infinite number of points $\vec{p}$ lying within the cuboid defined by the scalar field. This means that an interpolation prescription always has to be given in order to derive the scalar value s $(\vec{p})$ for any arbitrary point $\vec{p}$ in space from its surrounding voxels.

Mainly two techniques for volume rendering may be distinguished. They differ in how the representation of the 3D model is chosen in order to perform the manipulation and representation steps. While the indirect methods first transform the voxel model into a *surface representation*, the direct methods work on a regular grid [13]. Both types will be further described in the sections 4.1 and 4.2. This rather theoretical chapter lays the foundations for chapter 5 where different aspects of visualization in medical imaging will be investigated.

4.1 Indirect Volume Rendering

Indirect volume rendering techniques have in common that they compute an intermediate representation from the grid described above which in turn is object to some manipulation steps and is finally used in the illustration step. This transformation comes along with a loss of information which is accepted because of either performance reasons or other advantages of the chosen representation, as we will see below [13, 77].

A very well-known transformation is the *Marching-Cubes-Technique* by Lorensen and Cline [49]. According to the *iso-value*, an *iso-surface* is computed which separates the areas on both sides of the surface into one being higher and one being lower than the chosen threshold. For this purpose, the vertices of the grid are categorized by means of the iso-value as either lying in the inner or outer region. The most interesting are those belonging to an edge with endpoints in both regions. The vertices are interpreted as corners of cubes[3] which now may

[1] Spaces other than $\mathbb{R}$ are not of interest and will be ignored in this context

[2] Another term for a regular grid used in AMIRA

[3] For the sake of generality, also unstructured grids are considered

have one of 255 corner configurations. Due to symmetry, 15 general configurations, as shown in figure 4.1, exist.

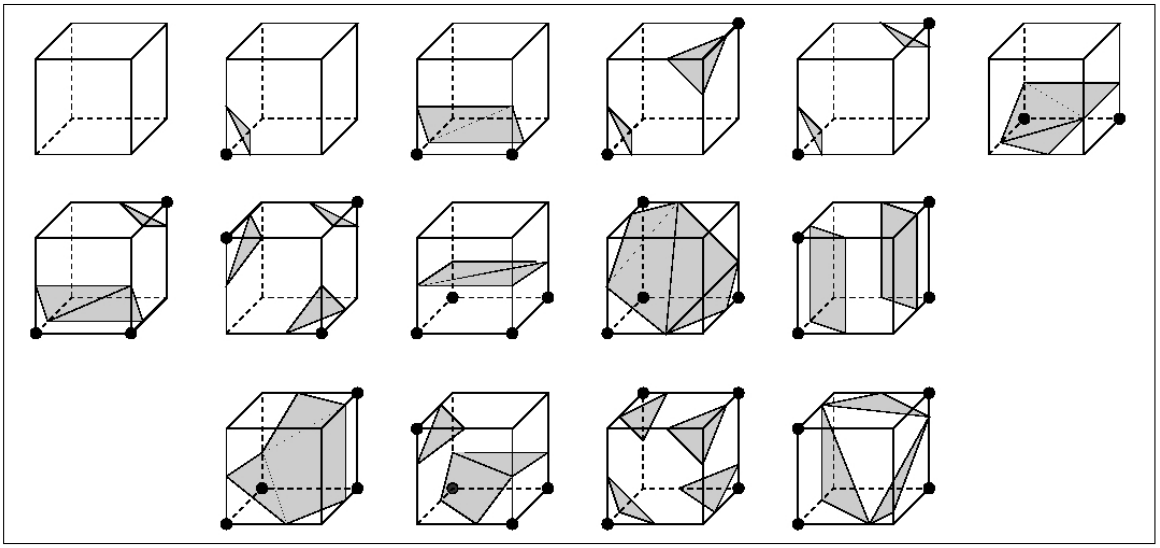


Figure 4.1: Cube-Marching: 15 basic corner configurations [13]

Subject to its general category, triangles are calculated by tri-linear interpolation between the edges of the cube. This is sometimes called *triangulation*. Additionally, the vertices' normals are calculated, also by linear interpolation of the normals in the corners of the cube. Visualization of such triangle models is well supported nowadays by consumer graphics hardware. Since very large numbers of triangles[4] are needed for a high-quality visualization of the model and their rendering still leads to performance problems, a plethora of optimizations has been proposed. One category points at the minimization of necessary triangles by reorganization of the surface model. Such techniques may either be applied during triangulation, or afterwards. Other attempts try to enhance cell-traversing in order to achieve faster access to the triangles of interest. The introduction of an octree-representation by Wilhelms led to a tremendous gain in rendering performance [79]. A good overview over these and many other techniques is given by Engel [13].

The approach of defining an iso-value as a border between an inner and an outer region can be applied repeatedly — we may speak about *multi-thresholding* — in order to separate several materials imaged by the 3D model. This process is called *image segmentation* and shall be described in the following [29]. Its basic principle is to identify different kinds of materials by means of their characteristic density values which may be compared to the scalar values s associated with each voxel in the model. For this purpose, the *histogram* is calculated, which shows the distribution of density values within the voxel model. For some discretized density values, the number of voxels being associated with this value are counted and the result is displayed in a diagram as shown in figure 4.2.

Ideally, the maxima correspond to the characteristic density value of a distinct material or tissue. Therefore, thresholds are set in between in order to separate them from each other.

[4] A realistic value is several million triangles

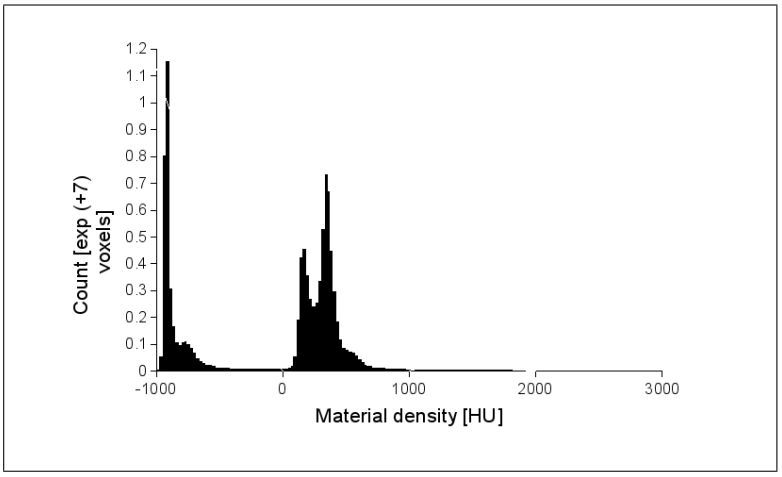


Figure 4.2: Histogram of sample CT model

However, this categorization is often only used as a pre-segmentation step since the density distributions belonging to different materials may overlap in the histogram. By manual intervention, the segmentation has to be improved. This is achieved by usage of different tools, which are also dependent on the scalar values s associated with the voxels, but can produce more specific results, since they are only applied locally. *Edge detection* tools follow user input and suggest probable borders based on the gradients of s_i in the environment of the cursor. *Region growing* models the propagation of some liquid on a more or less sloping surface in time, starting at a given point. Thereby, the dispersion over voxels around the starting point is calculated in discretized steps and the smaller the gradient values s_i are, the more voxels are traversed within this step in a certain direction [16]. Still the most common technique — even though the most time-consuming — is optimizing the boundary of a certain structure or material by manual truncation of undesired voxels. Thereby, either free-form curves are drawn in 2D or planes are defined in 3D, separating unwanted voxels from the rest. A very detailed description of segmentation techniques is given by Groher [25].

4.2 Direct Volume Rendering

Direct volume rendering is a 3D visualization technique, which depends on emission and absorption of *energy* in the broadest sense. In contradiction to the radiosity technique [20], however, the computation is not based on a 3D surface model consisting of polygons but on 3D voxel data. As we will see in chapter 5, this is exactly the data type provided by 3D medical imaging modalities like CT or MRT. Each voxel is associated with some functionally determined parameters influencing its properties concerning emission and absorption of radiation, regardless of its kind. Visualization of transparent objects may be performed by modelling normal light. This applies for direct volume rendering as well as for the real world. For representation of the inside of opaque objects, however, radiation has to be modelled differently. For this reason, medical imaging techniques depending on X-ray radiation have been invented. Direct volume rendering is able to imitate the creation of 2D X-ray

radiographs, if their physical characteristics are adopted correctly to the volume rendering process, as we will see. The efficient calculation of DRRs is absolutely necessary for an image-based registration approach since lots of DRRs will be needed by the registration algorithm, as will be described in chapter 6. Therefore, this section shall be devoted to an in-depth consideration. In the following, the shorter term *volume rendering* will be used instead of direct volume rendering, when its denotation is clear from the context.

Section 4.2.1 will explain in detail the ideas behind the volume rendering technique on the basis of an very intuitive, even though inefficient method, the so-called ray-casting algorithm. With an understanding of the functionality in principle in mind, we can investigate different enhancements in 4.2.2. Different algorithms are available, like the shear-warp factorization of the viewing transformation described in 4.2.2.1, which may be implemented efficiently in software, or the 2D/3D texture-mapping based algorithms as explained in 4.2.2.2, which strongly depend on hardware acceleration available on modern graphics hardware. Somehow different is the interpolation between training images in eigenspace representation explained in 4.2.2.3. It does not depend on the efficient calculation of a DRR intra-operatively but on the efficient fusion of different DRRs which have been acquired pre-operatively.

4.2.1 The Ray-Casting Algorithm

As the name implies, the *ray-casting* algorithm depends on *viewing rays* which are shot through the voxel model. A good introduction to this basic technique is given in [21] or in [15, 14, 13]. In the remainder of this section, I will stick to the notation of the latter three works, since it seems to be more consistent.

4.2.1.1 The Volume Rendering Integral

For each pixel in the image plane, one ray $\mathbf{x}(\lambda)$ with λ the distance from the *viewing point* is considered, starting at the image plane. Each ray is unambiguously defined by the camera model which has been chosen before. Different camera models have already been described in chapter 3. In the case of a perspective camera, all rays intersect in the camera center or viewing point whereas they run in parallel in the case of an orthographic projection (see figure 4.4).

We now assume that the following two functions may be calculated for each point $\vec{p}$ in 3D space.

- Color density $\mathbf{color}(\mathbf{s}(\vec{p}))$
- Extinction density $\mathbf{extinction}(\mathbf{s}(\vec{p}))$

Both functions are applied to the scaler values $\mathbf{s}(\vec{p})$ belonging to the points $\vec{p}$ lying on a ray $\mathbf{x}(\lambda)$. The unit of the *color density* is color intensity per length and is either a scalar or a vector, depending on the chosen color model. The unit of the *extinction density* is extinction factor per length respectively. It denotes the *attenuation* of energy passing through. The attenuation of photons traversing a certain material is described by the *attenuation law*

$$I = I_0 \cdot e^{-\mu \cdot d} \tag{4.1}$$

with primary intensity I_0 and photon intensity I after a distance d [28]. The *linear attenuation coefficient* μ is defined by $\mu = \ln \frac{I_1}{I_2}$ where I_2 is normally measured one centimeter afar from I_1 [67]. With the given definitions, the *volume rendering integral*

$$\int_0^d \mathbf{color}\left(\mathbf{s}\left(\mathbf{x}\left(\lambda\right)\right)\right) \exp\left(-\int_0^\lambda \mathbf{extinction}\left(\mathbf{x}\left(\lambda'\right)\right) d\lambda'\right) d\lambda \tag{4.2}$$

may be set up with d being the maximum length of the rays [40]. It has been described first by Levoy [48]. The integral describes the emission of energy according to $\mathbf{color}\left(\mathbf{s}\left(\mathbf{x}\left(\lambda\right)\right)\right)$ at each point on the ray and the attenuation of energy according to the integrated attenuation coefficients along the first part of the ray (with respect to the viewing point). Unfortunately, this form of the integral is not directly applicable for visualization of a scalar field because the calculation of **color** and **extinction** is unspecified so far.

4.2.1.2 Classification and Shading

Two steps are distinguished for calculation of the above-mentioned functions **color** and **extinction**: classification and shading [15, 14, 13, 77]. In the *classification step*, a so-called *primary color*[5] and an extinction coefficient is associated with a point $\vec{p}$ in space. This is achieved by definition of the *transfer functions* $\tilde{\mathbf{c}}\left(\mathbf{s}\right)$ for color densities and $\tau\left(\mathbf{s}\right)$ for extinction densities respectively. $\tilde{\mathbf{c}}$ denotes either a scalar or a vector, depending on the color model, while τ returns the linear attenuation coefficient μ introduced above.

During the *shading step*, modifications may be applied to the primary color. This is normally achieved through incorporation of classical shading models like e.g. the Blinn-Phong surface shading technique, which can also be implemented, without an explicit surface model, by using the normalized gradients $\bigtriangledown\mathbf{s}\left(\vec{p}\right)$ of the scalar field [52, 34, 35]. This is also referred to as the introduction of *multidimensional transfer functions*, since the scalar value s is replaced by a vector containing together with s also the gradient data and maybe other derivatives. Shading is necessary, if different light sources or scattered light shall be modelled. This can be useful in order to enhance surface structures in the model to be visualized and will be renewed in chapter 5. However, in the trivial case of an X-ray modality, where photons are only attenuated but not scattered, and no sources of energy except the X-ray tube exist, the shading step can be omitted and $\tilde{\mathbf{c}}\left(\mathbf{s}\right)$ is identified with $\mathbf{color}\left(\vec{p}\right)$. The same holds for $\tau\left(\mathbf{s}\right)$ which may be identified with $\mathbf{extinction}\left(\vec{p}\right)$. This yields the following form of the rendering integral.

$$\int_0^d \tilde{\mathbf{c}}\left(\mathbf{s}\left(\mathbf{x}\left(\lambda\right)\right)\right) \exp\left(-\int_0^\lambda \tau\left(\mathbf{s}\left(\mathbf{x}\left(\lambda'\right)\right)\right) d\lambda'\right) d\lambda \tag{4.3}$$

Please refer to [13] for an in-depth treatise on shading. Recent shading techniques have been described by Kniss [36, 34, 35, 37].

4.2.1.3 Pre- and Post-Classification

In 4.2.1.1 it already has been mentioned that a voxel model consists of an arbitrary, but finite number of voxels. Thus, $\mathbf{s}\left(\vec{p}\right)$ has to be evaluated by interpolation. There are two possibilities for the application of the transfer functions. In *pre-classification*, the transfer functions are

[5] Borrowed from the OpenGL terminology

applied to the voxel data *before* the interpolation prescription is used to evaluate the transfer functions for an arbitrary point $\vec{p}$ whereas in *post-classification* they are applied *after* the interpolated scalar value $\mathbf{s}(\vec{p})$ has been calculated.

Since the interpolation does normally not commute[6] with the application of the transfer functions, only post-classification is correct in the sense of applying the transfer functions to a continuous scalar field $\mathbf{s}(\vec{p})$ derived from voxel data along with a given interpolation prescription. However, pre-classification may speed up the numeric approximation of the rendering integral quite a lot [15]. Details about pre- and post-classification can be found in [13] and [62].

4.2.1.4 Numerical Integration of the Volume Rendering Integral

A very common technique of numerically approximating an integral is the calculation of the Riemann sum [7]. This yields that the exponent of the integral 4.3 is approximated by

$$\exp\left(-\sum_{i=0}^{\lambda/d}\tau\left(\mathbf{s}\left(\mathbf{x}\left(i\cdot d\right)\right)\right)d\right)=\prod_{i=0}^{\lambda/d}\exp\left(-\tau\left(\mathbf{s}\left(\mathbf{x}\left(i\cdot d\right)\right)\right)d\right)=\prod_{i=0}^{\lambda/d}\left(1-\alpha_i\right) \tag{4.4}$$

with the opacity of the i-th ray segment being defined by

$$\alpha_i=1-\exp\left(-\tau\left(\mathbf{s}\left(\mathbf{x}\left(i\cdot d\right)\right)\right)d\right). \tag{4.5}$$

This can be further approximated by

$$\alpha_i\approx\tau\left(\mathbf{s}\left(\mathbf{x}\left(i\cdot d\right)\right)\right)d \tag{4.6}$$

for small d. The color $\tilde{C}_i$ emitted in the i-th ray segment can be approximated by $\tilde{C}_i \approx \tilde{\mathbf{c}}(\mathbf{s}(\mathbf{x}(i\cdot d)))d$. This yields the following approximation of the complete volume rendering integral.

$$I\approx\sum_{i=0}^{n}\tilde{C}_i\prod_{j=0}^{i-1}\left(1-\alpha_j\right) \tag{4.7}$$

A *back-to-front-compositing algorithm* could therefore proceed iteratively as follows:

$$\tilde{C}_i'=\tilde{C}_i+\left(1-\alpha_i\right)\tilde{C}_{i+1}' \tag{4.8}$$

Thereby, the iteration starts with $i=n$. Often, $\tilde{\mathbf{c}}_\mathbf{i}$ is substituted by $\tau(\mathbf{s})\mathbf{c}(\mathbf{s})$. This results in the approximation

$$C_i\approx\tau\left(\mathbf{s}\left(\mathbf{x}\left(i\cdot d\right)\right)\right)\tilde{\mathbf{c}}\left(\mathbf{s}\left(\mathbf{x}\left(i\cdot d\right)\right)\right)d \tag{4.9}$$

This denotes that the color $\mathbf{c}_\mathbf{i}$ of the i-th ray segment is already attenuated by the opacity value $\tau_\mathbf{i}$ in of the same ray segment. This yields the more common approximation of the rendering integral

$$I\approx\sum_{i=0}^{n}\alpha_i C_i\prod_{j=0}^{i-1}\left(1-\alpha_j\right) \tag{4.10}$$

[6] The only exceptions are either a constant interpolation prescription or constant or identity transfer functions [15]

and its corresponding back-to-front-compositing equation

$$C_i^{'} = \alpha_i C_i + (1 - \alpha_i)\, C_{i+1}^{'}. \tag{4.11}$$

This form of the equation has been introduced by Porter and Duff, actually for blending images, and has been called *over-operator* [59]. Colors that have been pre-multiplied by its associated opacity value are called *associated -*, *pre-multiplied -* or *opacity-weighted colors*. According to Blinn, associated colors correspond to their opacity, i.e. they are regular colors mixed on black.

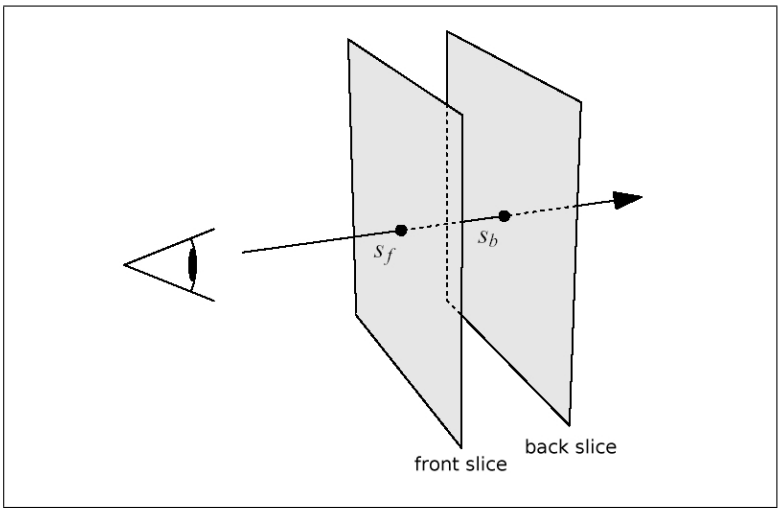


Figure 4.3: Viewing ray traversing slices of the image stack [13]

The number of samples in the approximation of the rendering integral has to be chosen in accordance to the *Nyquist theorem*. This means that the sampling frequency must be at least equal to those of the product of $\mathbf{s}(\vec{p})$ and the maximum frequency of the two functions $\tilde{\mathbf{c}}(\mathbf{s})$ and $\tau(\mathbf{s})$ in the case of equation 4.8. If equation 4.11 shall be used, the frequency of the maximum has to be replaced by the product of the frequency of the two functions $\tilde{\mathbf{c}}(\mathbf{s})$ and $\tau(\mathbf{s})$ [13]. This may be a problem for non-linear transfer functions. Engel describes an approach which he calls *pre-integrated volume rendering* and enables low sampling rates without artefacts also in that case. His idea is to integrate color and opacity values according to the volume rendering integral 4.7 piecewise for arbitrary scalar values $s_{front}(\vec{p})$ and $s_{back}(\vec{p})$ for all ray-segments in advance so that they can be looked up at run-time. Due to this pre-integration step, high Nyquist frequencies of the transfer functions do not influence the sampling rate any longer [14].

4.2.2 Overview of Efficient Volume Rendering Algorithms

There exist several approaches to an enhancement compared to the execution of the back-to-front compositing algorithm described by equation 4.8 for each back-projected ray. A very good overview has been given by Salama [62]. In general, direct volume rendering algorithms can be roughly categorized in two classes [13].

Image Order Techniques The foundation for this class of algorithms is the 2D image. Starting with its pixels, rays are back-projected into the scene in order to compute the contribution of the voxels to its value, as shown in figure 4.4. This means that either intersections of the ray with the voxels have to be calculated[7] or the voxels have to be rearranged in some manner according to the orthogonal or perspective geometry implied by the virtual arrangement of the image and the 3D model in the chosen camera model (see chapter 3 for a description of different camera models).

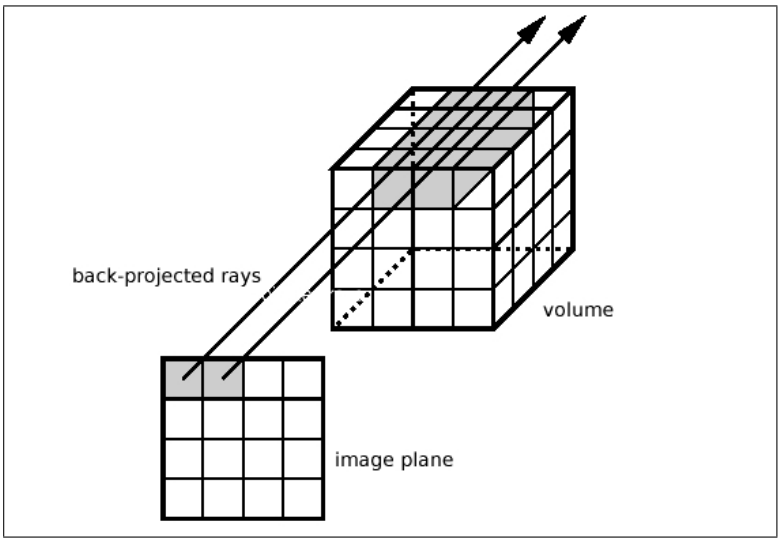


Figure 4.4: Back-projection of rays for the case of an orthogonal projection [13]

Object Order Techniques Here, the sequence is vice versa. For each voxel in the 3D model, its contribution to the pixels in the image plane is calculated. This may be achieved by the (forward) projection of the voxel which results in a *footprint* on the image plane. One representative of this kind is the *splatting*-technique [78]. First, the voxel center is projected and then a region of pixels is approximated around the projected point (see figure 4.5). Like above, the projection depends on the setup of the scene and the chosen camera model. Beyond this classification, there are two orthogonal concepts [13].

Frequency Domain Techniques By transformation of the 3D model to frequency space, the resulting 2D image in frequency space can be computed easily. In order to obtain the desired image in color space it is just necessary to apply the inverse transformation. The *Fourier-domain-volume-rendering* belongs to this category.

Compression Domain Techniques First, the 3D model is compressed by application of hierarchical basis functions, e.g. a wavelet basis. The rendering can now take place on the compressed model with only local decompression. Thus, memory requirements are decreased dramatically which also leads to an increased performance, especially in the case

[7] The back-to-front compositing algorithm may be implemented this way

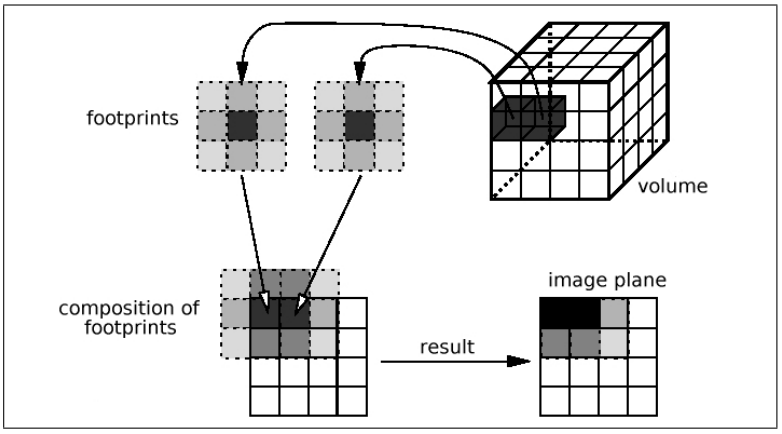


Figure 4.5: Composition of footprints in object order volume rendering [13]

of low bandwidths in data transfer. Combinations of frequency domain and compression domain techniques have also been described in the literature.

Common to the latter two techniques is the approximative evaluation of the volume rendering integral (equation 4.3) [13] for every pixel in the 2D target image, hence the integration of attenuated color and absorption coefficients along a viewing ray [14].

Two representatives of efficient techniques shall be presented in the following. The first approach is commonly referred to as the shear-warp factorization of the viewing transformation and is a typical object-order technique, as we will see. It may be fully realized in software and is therefore hardware-independent. The second approach exploits texture-mapping functionality available on recent graphics hardware, and thereby, achieves an efficient ray-casting. Thus, it is an image order technique. The utilization of texture-mapping yields a heavily hardware-dependent solution. Furthermore, a technique shall be outlined, which does not directly try to solve the volume rendering problem efficiently but computes arbitrary views of a 3D model by linear combination of a set of pre-computed views, so-called *training images* [63]. This method shall also be briefly described.

4.2.2.1 Shear-Warp Factorization of the Viewing Transformation

The *Shear-warp factorization of the viewing transformation* is an efficient rendering technique introduced by Lacroute [44, 42, 43]. The VOLPACK library which implements this approach is also available for download at the authors site [41]. The basic idea behind this approach is to split the rendering process into several steps so that the cost-intensive computations can be performed more efficiently, as we will see. These steps shall be shortly summarized first.

1. By means of shearing, the volume data is transformed from object space to sheared object space in which all viewing rays run in parallel to the z coordinate axis.

2. Now, e.g. the back-to-front compositing algorithm may be applied to compute an *intermediate* image. Because all viewing rays now run in parallel, the algorithm can take advantage of run-length encoded data which leads to an enormous speed-up compared to tracking an arbitrary ray through the volume.

3. Last, the intermediate image in sheared object space is warped by a projection onto the desired image plane and we obtain the coordinates of the final image.

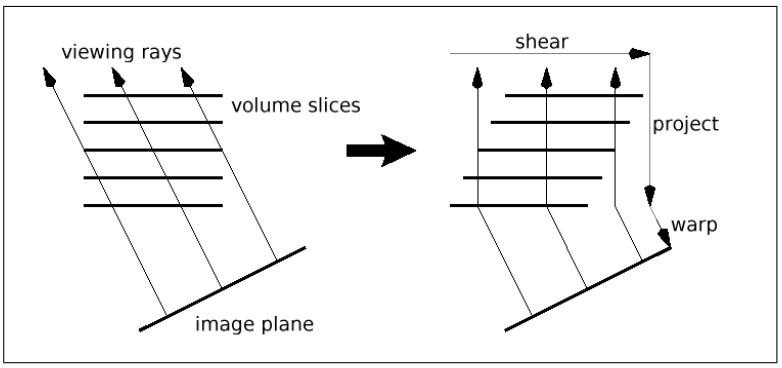


Figure 4.6: Shear-warp factorization of the viewing transformation in the case of an orthogonal projection geometry [44]

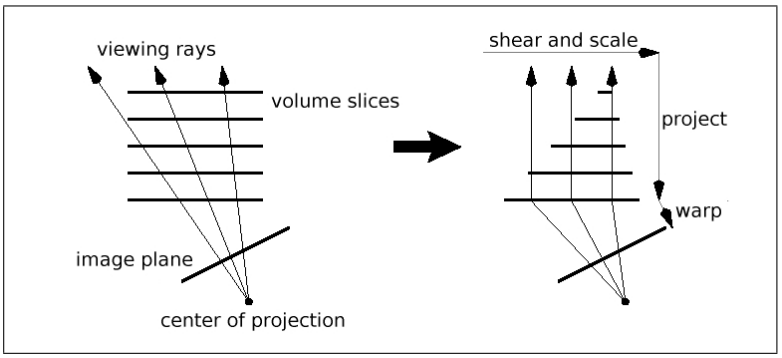


Figure 4.7: Shear-warp factorization of the viewing transformation in the case of an perspective projection geometry[44]

The view transformation matrix $M_{view} \in \mathbb{R}^{3x4}$, which depends on the chosen camera model, can be factorized as follows [42]:

$$M_{view} = P \cdot S \cdot M_{warp} \tag{4.12}$$

where P resembles an orthogonal matrix which transforms the object coordinate system in order to make the z-axis the principal viewing axis, S is a shearing matrix for the transformation to sheared object space and M_{warp} depicts the 2D warp to the final image coordinates. Please refer to chapter 3 for a description of different camera models. Depending on whether a parallel or a perspective projection shall be modelled, S either only contains a shearing component or a shearing combined with a scaling component. Voxel data may be interpreted as consisting either of axial, frontal or sagittal slices. The algorithm is applied to those stack whose z-axis is most orthogonal to the viewing direction, and the volume data is sheared in slice direction. The shearing is visualized in the figures 4.6 and 4.7 for both, the

orthogonal and the perspective case. The corresponding matrices look as follows.

$$S_{par} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ s_x & s_y & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.13}$$

where s_x and s_y can be computed from the elements of M_{view}.

$$S_{persp} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ s_x^{'} & s_y^{'} & 1 & s_w^{'} \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.14}$$

Here, a certain slice z_i is translated by $(z_i s_x^{'}, z_i s_y^{'})$ and then scaled by $1/(1 + z_i s_w^{'})$. The final term of the factorization is a matrix which warps coordinates from sheared object space to final image space.

$$M_{warp} = S^{-1} \cdot P^{-1} \cdot M_{view} \tag{4.15}$$

One advantage of this method is, as already mentioned above, that the scan-lines of the volume data and the intermediate image are always aligned which allows for linear data access during the cost-intensive calculations in 3D-space. Only the last warping step, which actually is a 2D homography, depends on interpolation. Octrees may be used for volume representation which permits to skip uninteresting[8], e.g. transparent voxels. Another advantage is that the implementation is not hardware-dependent in contradiction to the texture-mapping based approach described next. One drawback, however, is a slight loss of quality due to the two-pass re-sampling of the data, using only bi-linear or bi-cubic interpolation. In return, it is a very fast technique.

4.2.2.2 Texture-Mapping Based Approaches

Texture-mapping functionality on modern graphics hardware provides for tremendous enhancements of generic techniques like the back-to-front-compositing algorithm described in 4.2.1.4. Thereby, it exploits the capabilities of the texture-mapping unit which normally is responsible for the mapping of texture maps from texture memory to arbitrary polygons of a 3D model. The exploitation of this feature for volume rendering has been described first by Cabral [9]. Many publications build up on this approach [77, 15, 14, 13].

For an understanding of how the texture-mapping based approach is able to speed up the volume rendering process, we first need a basic understanding of how texture-mapping is implemented [74, 45]. There are two rather different modes, the older 2D texture-mapping and 3D texture-mapping available on recent GPUs[9]. Both have in common that the surface of an object which commonly is approximated by polygons, is overdrawn by one ore more bitmaps called *texture maps*. For simplification of this process, polygons are further divided into triangles which consist of the vertices v_1, v_2, v_3 and the edges $e_1 = (v_1, v_2)$, $e_2 = (v_1, v_3)$, $e_3 = (v_2, v_3)$. It may be assumed that each point $\vec{p}$ on the triangle can be described by a linear

[8] According to the chosen transfer functions
[9] Graphic Processing Unit

combination $\vec{p} = \lambda\vec{e_1} + \mu\vec{e_2}$ with *texture coordinates* $\lambda, \mu \in [0..1]$ and $\vec{e_1}$ and $\vec{e_2}$ the directed edges of the triangle. The two modes differ in the point of time, the texture map is applied to the triangle.

In 2D-mode, the texture map is applied after the triangle has been projected to the screen or frame-buffer. This implies that v_1, v_2 and v_3 are in 2D-space. Furthermore, they correspond to some pixel coordinate in a 2D texture map in a 2D texture memory because the application designer or the user has associated this specific texture map with the triangle. Each pixel of the triangle in the frame-buffer can now be rendered by a lockup with its specific texture coordinates in the texture map. There remains only one problem: How are real numbers λ and μ mapped to discrete pixel coordinates in the texture map? The only solution to this problem is interpolation. At least bi-linear interpolation along the x and y coordinate axes of texture memory is necessary for a satisfactory result. It is one major challenge for the designers of graphics hardware to make this texture map lookup, including the interpolation, as fast as possible. And indeed, modern GPUs are much more suited for solving this problem compared to a current CPU. In addition, features like anti-aliasing are also implemented efficiently.

In 3D-mode, the texture map is applied before projection of the triangle to the frame-buffer. This means that v_1, v_2 and v_3 are in 3D-space. Like above, they have been associated with some coordinates in a texture map by the application designer or the user. But these coordinates can not be pixel coordinates any longer. Instead, a 3D texture memory containing voxels is used and points on the triangle are derived by e.g. tri-linear interpolation along the x, y and z coordinate axes of texture memory. This promises an increased quality compared to 2D-texture-mapping.

In the case of 3D texture-mapping available on the hardware, we may just put 3D voxel data into the texture memory. According to the current viewport of the camera, polygons can be defined orthogonal to the viewing direction and with equidistant spacing. This can be seen in figure 4.9. They represent an arbitrary intersection of a plane with the voxel data in texture memory. The mapping of these intersections to the polygons is performed by the hardware. At last, the single polygons are blended via the over-operator (equation 4.11). Thereby, either pre- or post-classification (see 4.2.1.2) is implemented in hardware, depending on the manufacturer. With 2D texture-mapping, instead of the 3D voxel data, the image stack closest to an orthogonal orientation to the viewing direction is used. Instead of tri-linear interpolation, only bi-linear interpolation is used, which leads to a quality loss compared to the former case. Furthermore, all three stacks have to be available in RAM at the same time, and time-consuming switches between different stacks have to be performed whenever the viewing angle crosses the 45 degree boundary between the perpendiculars of two image stacks as depicted in figure 4.8. The advantages of 2D-mode are better performance due to simplified interpolation and also better availability on current consumer graphics hardware.

An efficient implementation based on this approach is described by Engel [15]. His rendering library OPENQVIS is available for free download at SOURCEFORGE [12]. Modern GPUs combine the application of up to eight texture maps simultaneously. This is called *multi-textures* and standardized in e.g. the OpenGL 1.2 standard. However, there is also defined a strict sequence for the application of different texture maps which constrains the algorithmic design. Therefore, the best implementations are not based on OpenGL but on

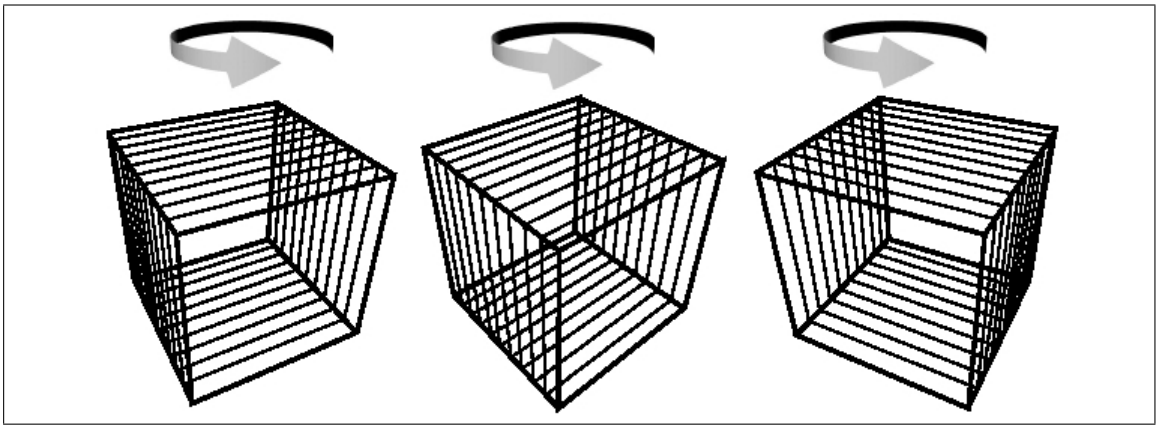


Figure 4.8: Stack-aligned slices in volume rendering based on 2D texture-mapping [13]

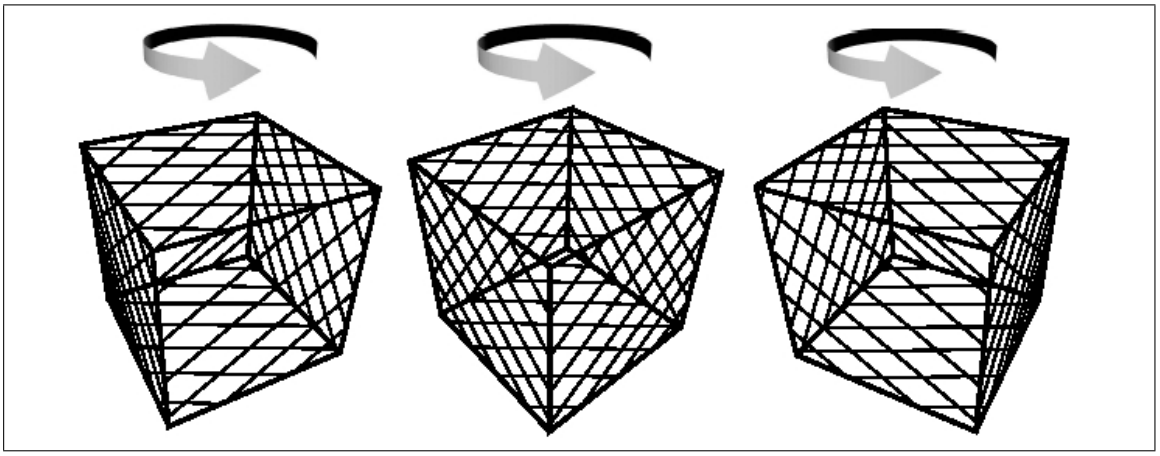


Figure 4.9: Viewport-aligned slices in volume rendering based on 3D texture-mapping [13]

the manufacturer-specific OpenGL extensions, unfortunately.

4.2.2.3 Interpolation of Training Images in Eigenspace Representation

In the following, a very interesting approach shall be briefly addressed. It has been successfully used by Roth also in the context of DRR generation for 2D-3D registration [63]. It is based on the theory of *eigenspace representation* which has been proposed by Murase [54]. The basic idea behind it is not optimization of the volume rendering process but rather the avoidance of volume rendering as often as possible. This may be achieved by a set of *training images* that, once available, serve for computation of arbitrary images by means of interpolation between suited training images. However, only the basic concept can be demonstrated in this context. For a detailed description, please refer to the sources.

The location of an object in the viewport of a camera is determined by six degrees of freedom (see also chapter 3), the translations t_x, t_y, t_z and the rotation angles ϕ, θ, ψ. Only two of

these, namely ϕ and θ significantly change the imaged object, when modified. The four other parameters can be approximated quite good by 2D plane transformations. A translation t_z along the viewing axis just makes the image smaller which can be compensated by 2D scaling. Analogously, rotations ψ around the viewing axis may be represented by 2D rotations in the image plane. This also holds for the translations t_x and t_y, as long as the z-dimension of the object is very small compared to its distance from the camera center and the object's centroid stays near to the viewing direction of the camera. Therefore, it is assumed that training images taken at arbitrary combinations of the remaining two angles ϕ and θ are sufficient for interpolating also the other ones. Within a reasonable range which covers the values ϕ and θ for the given application, these two angles are varied in equidistant steps in order to create training images x_{ϕ_i,θ_j}. From those, an average image

$$\bar{x} = \frac{1}{N^2} \sum_{\phi,\psi} x_{\phi,\psi} \tag{4.16}$$

is calculated with N being the number of steps for the modification of one angle. The average image $\bar{x}$ is subtracted from each of the training images x_{ϕ_i,θ_j} in order to create difference images $\hat{x}_{\phi_i,\theta_j}$. Each difference image shows, how much the corresponding training image differs from the average image $\bar{x}$. The average image $\bar{x}$ and also the difference images $\hat{x}_{\phi_i,\theta_j}$ are now represented by linearization row by row as vectors $\vec{\bar{x}}$ and $\vec{x}_i$ respectively. Thereby, arbitrary image vectors $\vec{x}$ may be approximated by the linear combination

$$\vec{x} = \vec{\bar{x}} + \sum_{i=0}^{n-1} (\omega_i \vec{\hat{x}}_i) \tag{4.17}$$

where $n = N^2$. The vectors $\vec{x}_i$ are combined in a matrix X with row vectors $\vec{x}_i$ in order to perform a main axis transformation [32]. Thus, the symmetric covariance matrix

$$C = XX^T \tag{4.18}$$

is diagonalized by solving of the common eigenvalue-problem

$$Ce_i = \lambda_i e_i \qquad \text{or} \qquad D = E^T C E \tag{4.19}$$

where $D = diag(\lambda_0, ..., \lambda_{m-1})$ consists of the eigenvalues of C, $E = [e_0, ..., e_{m-1}]^T$ is an orthonormal matrix with the eigenvectors in its rows and m is the dimension of the symmetric matrix C. This defines a coordinate transformation which eliminates linear correlations between single training images. The invariant sub-spaces of C are expressed explicitly. The linear independent eigenvectors may also be used for representation of an image as follows.

$$\vec{x} = \bar{x} + \sum_{i=0}^{m-1} (\omega_i \vec{e}_i) \tag{4.20}$$

It may be shown first that the weight ω_i for a distinct eigenvector $\vec{e}_i$ correlates with the magnitude of the corresponding eigenvalue λ_i, and second that the eigenvalues differ considerably. Therefore, a small subset of eigenvectors already represents a very good basis for approximation. Consequently, it would not make any sense to solve the complete eigenvalue problem in equation 4.19 by standard techniques like Jacobi rotations. This will be

impossible in most cases anyway, since X probably contains more column than rows (one image contains more pixels than training images will be needed). From this it follows that C is a singular matrix and only k eigenvalues may be calculated, e.g. by means of the *singular value decomposition* (SVD) [30, 61, 54]. This yields the following approximation for the linear combination in equation 4.20.

$$\vec{x} \approx \bar{x} + \sum_{i=0}^{k-1} (\omega_i \vec{e}_i) \tag{4.21}$$

The weights ω_i for a certain training image $\vec{x}_j$ are obtained by projection of the image vector into eigenspace. It may be shown that the projected points lay on a surface in eigenspace and that the distance between two points on the surface is a direct measure for the difference between images corresponding to these points [54]. Therefore, it is possible to interpolate arbitrary points, e.g. by means of a *bi-cubic spline function* $S(\nu_{r_x}, \nu_{ry})_{\nu_{r_x}, \nu_{ry} \in [0..1]}$. Thereby, the *spline parameters* ν_{r_x} and ν_{ry} are the two angels ϕ and θ that have been mapped linearly to the interval $[0..1]$. The corresponding image is calculated by applying the coordinates of the interpolated points as weighs ω_i in equation 4.21.

5 X-Ray Based Medical Imaging

In chapter 4, techniques have been described, which are suited for transformation and visualization in the context of medical imaging. However, issues specific to the medical context, especially X-ray imaging, have not been considered yet. This will be made up for now.

At first, we have to understand, how medical image data is acquired. Section 5.1 will cover the basic principles common to 2D and 3D X-Ray diagnostic. Next, methods for visualization of 3D volume data acquired by CT scanners shall be described in section 5.2. Also very important for CANP is the computation of artificial X-ray images (DRR) from 3D volume data. Please keep in mind that we will need such DRRs for the 2D/3D registration described in chapter 6. Therefore, it will be covered in detail in section 5.3. Finally, chapter 5.4 will show, how the described techniques have been implemented in CANP.

5.1 Image Acquisition Using X-Rays

Both, CT and X-ray fluoroscopy, make use of the interesting properties of X-rays for acquisition of information about the interior of the human body. Fluoroscopy is a perspective projection from a certain point of view and does not provide depth information. In contrast, CT scanners examine the body from different positions. For the computation of each axial slice, the X-ray source as well as the detector, are moved to different positions around the patient, describing a circle lying in the plane of the axial slice. In this way, many measurements are made. Each of them reflects the integral over attenuation along the ray from the source to the detector. From the gathered information, linear attenuation coefficients for each point within the body are calculated by a reconstruction algorithm. Several of this kind exist, but all proceed iteratively. Starting with an initial assumption of density distribution within the model, back-projected ray integrals are calculated and compared to the measured ray integrals. A subsequent optimization step tries to adjust the model such that both coincide. This is repeated until the desired accuracy is reached [1].

All imaging systems have in common that they use some kind of detection unit for measuring image data. Different systems are used in combination with different types of radiation, e.g. a color film for visible light or a X-ray film for X-rays. Nevertheless, most of these systems share some basic properties, which will be described in 5.1.1. Next, we will narrow our focus and investigate properties specific to X-ray based imaging systems. In 5.1.2 it shall be described, how X-rays are influenced by different materials and which consequences this has for contrast in the resulting image.

5.1.1 The Gradation Curve

At first, some basic definitions are necessary before the coherences will be highlighted. Most of them have been described by Hurter and Driffield in the context of photography in a paper published in 1890 [18]. Those describing the attenuation of energy through dense materials have been investigated in the context of X-ray fluoroscopy [1, 67]. The *linear attenuation coefficient*

$$\mu = \ln \frac{I_1}{I_2} \tag{5.1}$$

has already been introduced in chapter 4, in combination with the attenuation law. I_1 is the incoming intensity of a distinct radiation and I_2 the respective attenuated intensity in some depth d which is commonly measured after 1 cm. *Subject contrast* is the ratio of the highest and the lowest intensity of radiation to be sampled and depends on the object that shall be imaged. It depicts e.g. the intensity of X-rays that already have been attenuated by some object or also the visible light coming from the scene into the camera. *Detector contrast*, in contrary to subject contrast, gives information about the detector used for sampling the incoming radiation. Thereby, the detector is either a photographic or radiographic film or any other kind of measuring tool suitable for the given kind of radiation. It is also often referred to as *film contrast* or *gamma* value.

The detector contrast is expressed by the *gradation curve*. It is also called *characteristic curve* or *H&D curve* in honor of Hurter and Driffield. An example for a typical X-ray film is shown in figure 5.1. Other systems may have different curves, but its general shape will always be similar.

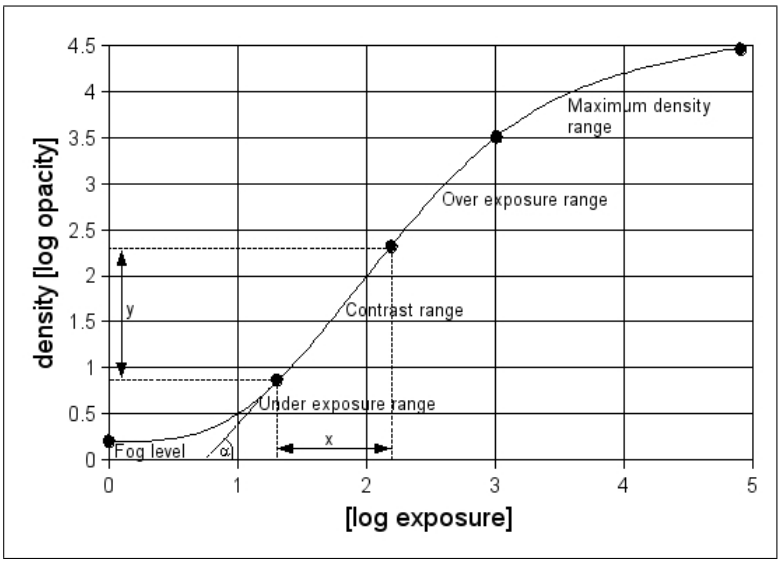


Figure 5.1: Typical gradation curve of a film radiograph

The curve characterizes the behavior of the film or sampling unit subject to distinct exposure rates. The human senses — not only the eye — show logarithmic behavior. That's why both coordinate axis have logarithmic scaling. Since X-ray fluoroscopy systems perform a

conversion from non-visible radiation to visible light, another term has been introduced. The *opacity* describes the properties of the film eyed under visible light. It is defined by

$$opacity = \frac{I_{in}}{I_{trans}}, \tag{5.2}$$

where I_{in} is the amount of light before, and I_{trans} behind the film radiograph. Contrast in opacity across the radiograph is responsible for the contrast observed by the human eye. This impression is called *density* and is defined logarithmically, due to the characteristics of the human eye.

$$density = \log(opacity) \tag{5.3}$$

It may be useful to point out the difference between exposure and opacity. The former is a measure for the intensity of radiation to be measured, while the latter is a measure for amount of visible light, which is absorbed by the radiograph and therefore not reaches the eye of the observer. Certainly, in digital radiographs, as those displayed by the C-arm, the opacity is simulated by CRT[1].

In most cases, it is sensible to image the intensity of a certain kind of radiation by a linear mapping to the resulting media. This ensures a good contrast over the whole intensity range and does not falsify the intensity values. This requirement is reflected in the terms *exposure latitude* or *useful exposure range* which describe the range of exposure which can be reasonably imaged by the detector system. Only the approximately linear part of the curve, which is titled *contrast range* in figure 5.1, is therefore considered. The gradient

$$\gamma = \frac{y}{x} = \tan(\alpha) \tag{5.4}$$

of this part of the curve is one major characteristic trait of a detector system, e.g. photographic film or, in our case, the X-ray detector of the C-arm. The steeper the curve, the more contrast will finally be available in the image, but also the smaller the range of imaged exposure will be. Thus, always a suitable tradeoff has to be found between high contrast on the one hand and a broad range off imaged exposure on the other hand. The characteristic curve is different for different film types but has a general shape as shown in figure 5.1. Another important property is the *speed* of the film or film-screen system, which can be judged from the curve's position along the horizontal axis. The correct speed is also a compromise between *high spatial frequency*, which means a low noise or *fog* level, and the magnitude of exposure that is available, e.g. natural light in nightly photographing or X-ray contamination that is justifiable. The reason for that is that a higher number of photons is expected to be less dependent from accidental behavior. The human eye is able to distinguish opacity differences related by a factor of approximately 100. This constrains the useful exposure range to about two density units, due to equation 5.3, even if the detector unit or film material were able to image more levels.

Recapitulating, it may be stated that the resulting (radiographic) *contrast* depends on the factors gamma and speed on the one hand and subject contrast on the other hand. Furthermore, in the linear region of the gradation curve, the following correlation applies:

$$c_{sample} \approx c_{subject} \cdot \gamma_{detector} \tag{5.5}$$

[1]Cathode ray tube

c_{sample} is the contrast of the acquired data and is also called *radiographic contrast* in the context of X-ray fluoroscopy, $c_{subject}$ is the subject contrast. Both, scattered radiation reaching the detector, and fog will reduce the radiographic contrast.

5.1.2 The Linear Attenuation Coefficient in X-Ray-Based Imaging Systems

The attenuation of X-rays within a compound object as the human body depends on several factors which also influence the X-ray setup used for a certain type of investigation [67]. First, those shall be addressed that are the basic requirement for X-ray fluoroscopy in principle.

- One important factor for the linear attenuation coefficient μ is the *atomic number* of the atoms contained in the material(s) or tissues to be imaged. The higher this quantity is, the higher μ will be.
- Another important quantity is the specific *density* of the material. The more atoms lie on the way of a photon, the more probable is its absorption and thus the higher μ will be.

These two factors are directly responsible for contrast in the radiation reaching the detector system. Beyond, there are some other influences which depend on the voltage used for acceleration of electrons within the X-ray tube and which are also very important.

- Generally, the probability of absorption decreases with increasing *photon energy*. Therefore, μ decreases with an increased voltage of the X-ray tube. Unfortunately, this effect is the more distinctive, the higher the *atomic number* of the elements is, which are contained in the subject of interest. This effect, however, is not very important in medical imaging since the human body mainly consists of relatively lightweight atoms.
- More influencing are the densities of the substances since they differ considerably in the human body. The probability of absorption decreases more for dense materials, compared to lightweight ones, when the voltage is increased. Fortunately, there is a roughly constant ratio between two linear attenuation coefficients μ_1 and μ_2 in two distinct materials, measured at different X-ray energies. This may be seen in figure 5.2.
- *Non-linearities* are introduced to this theory due to dispersion in photon energies produced by a certain X-ray tube. The emission of photons with energy differing from the peak level, cannot be avoided with the current technology. This effect will be ignored in the following since it is not very significant, but difficult to model.

The last but one effect is exploited by CT as follows. Presumed a given gradation curve, a certain useful exposure range is available for mapping contrast in exposure within the detector system. By choosing a higher X-ray energy, the contrasts in exposure for different materials move together and therefore a broader range of substances fits into the linear segment of the curve. In established systems, CT is performed at energy levels about 80 keV while traditional two-dimensional fluoroscopy only uses about 50 to 60keV. This is mainly due to the high radiation exposure at CT energy levels which is unacceptable for mobile devices used in the operating theater, where besides the patient, also other persons are attendant .

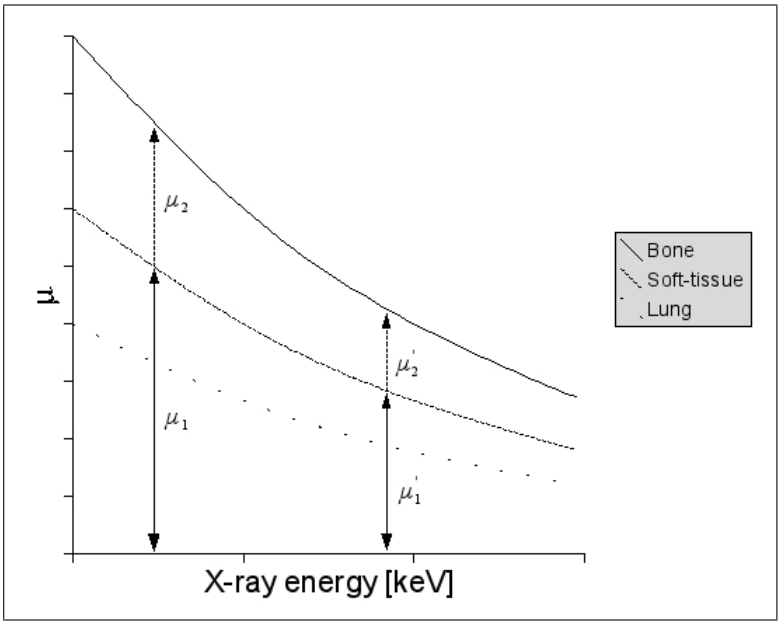


Figure 5.2: Attenuation subject to different X-ray energies

5.2 Three-Dimensional Visualization

In chapter 4, a theoretical overview over different volume rendering techniques has been given. This section describes, how these will be applied in CANP. Different requirements have to be satisfied. For pre-operative planning, the 3D patient model has to be examined by the surgeon in order to plan the stent location and maybe other operational aspects. This includes, by any means, browsing through axial, frontal and sagittal slices directly acquired by a CT scan. This represents the state-of-the-art method in a surgeon's everyday life. But with an additional 3D sight of the patient, additional information may be collected and the topology of the human body becomes much clearer. This is even more important during operation itself when time is very sparse. Information acquisition is by far more intuitive and therefore much faster when using a 3D model. Therefore, these visualization features are indispensable for an intuitive planning and navigation tool. In principal, two distinct methods — direct and indirect volume rendering — exist. Their suitability in medical imaging shall be considered in the following.

At the moment[2], mainly indirect volume rendering is used for visualization. This means that a surface model has to be reconstructed by image segmentation, as can be seen in 1.5 [29, 6]. In 4.1, it has already been described that segmentation can be a very time-consuming procedure and this holds especially for CTA models of the human body. As an example may serve the contrasted vessels which cannot be separated automatically from the bones since both materials roughly have the same linear attenuation coefficients. Due to this reason, surface models are often either of bad quality or completely unavailable to the surgeon. Another disadvantage is the corruption of data during segmentation, not only due to manual interaction, but also because of an additional interpolation step. However, they are presentable

[2]Visualization infrastructure belonging to SIEMENS SOMATOM SENSATION CARDIAC

by common graphic boards fast enough for interactive navigation.

The application of direct volume rendering methods instead, as described in section 4.2, could be an alternative. Since surface structures are not visible with an approach only based on ray-casting, an additional shading step (see 4.2.1.2) should be included. High quality visualizations as e.g. in figure 5.3 can be produced in this way. Since no intermediate surface model has to be generated, this technique is applicable as soon as a CT or MRT model of the patient is available, without the need of a complex segmentation. However, great demands are made on the performance of a visualization system, if the views shall be displayed and changed interactively. With coming generations of computer and graphics hardware, these demands should be satisfied even on a normal desktop PC [36, 34, 35, 37]. Direct volume rendering in combination with shading is not implemented yet in CANP.

Figure 5.3: Visualization of tooth, using direct volume rendering in combination with shading [38]

5.3 DRR-Computation

Our goal is the creation of DRRs which resemble a real radiograph as good as possible. However, there are some hazards which may complicate the achievement of this goal. These can be separated in principal problems arising from the differences between the two imaging modalities involved, CT and X-ray fluoroscopy, and more practical problems arising from different circumstances during image acquisition. The former shall be investigated in 5.3.1, the latter in 5.3.2. Based on the insights so far, an appropriate algorithm for the computation of DRRs shall be selected in 5.3.3, and finally, in 5.3.4, it will be shown, how the characteristics of X-ray fluoroscopy have to be modelled in the transfer functions necessary for the direct volume rendering process.

5.3.1 Principal Differences between DRR and Radiograph

One major source of differences between fluoroscopic images and DRRs are the different imaging modalities which are used for data acquisition.

5.3.1.1 Distortion of Fluoroscopic Images

C-arm images show a considerable amount of *pincushion distortion* and *spiral distortion*. These are caused by nonlinearities in the electron optic of the C-arm on the one hand and the curved face of the X-ray detector on the other hand [58]. Pincushion distortion is already compensated for in the electron optics of our C-arm, but mathematical models are also able to cope with these problems. They have already been described in the context of C-arm calibration in chapter 3.

5.3.1.2 Different Resolution

Due to different resolutions in data acquisition, DRR and X-ray image are of different quality. The CT scanner produces voxel data with a resolution of e.g. $512 \times 512 \times 800$ for a complete thorax while the C-arm detector has a resolution of 980×980 pixels which cover a section of app. $15\ cm^2$. Especially structures normal to the CT slices appear much more blurred in the DRR. For an adaption, the resolutions can be adjusted. However, this is a difficult task since due to the perspective projection, not only the size but also the depth of a voxel has to be taken into consideration for an optimal solution [8]. Nevertheless, blurring X-ray image may be chosen as an approximation [58].

5.3.1.3 X-Ray Tube Heel Effect

X-ray tubes have the characteristic trait that the intensity of the emitted X-rays differs according to the angle in which electrons hit the anode. This leads to a slight intensity difference from one side of the image to the other [58]. This effect seems to be already compensated for in our C-arm hardware, as can be seen in the calibration images in figure 3.3.

5.3.1.4 Different X-Ray Energies

More severe is the difference of X-ray energy in CT with about 80 keV and C-arm with about 50-60 keV [58]. As already described in 5.1.2, at CT energies, the ratio between the linear attenuation coefficients of bone and soft tissue is much lower than at the energy level of the C-arm. This effect will lead to more contrast between the above-mentioned materials using the latter modality. As long as non-linearities due to dispersion in photon energies (see 5.1.2) is ignored, this effect can be compensated for by a linear transformation of the contrast range, as will be described in 5.3.4.

5.3.2 Practical Differences Between DRR and Fluoroscopy

Besides the principal obstacles mentioned in 5.3.1 that have to be overcome, there exist at least two more practical ones. Those shall be addressed now.

5.3.2.1 Interventional Instruments in X-Ray Image

Interventional instruments used in the operating theater like a catheter or scalpels are mostly made of materials with high attenuation coefficients and thus produce a high contrast in the radiograph, but are not present at all during the CT scan. Otherwise, it is also conceivable to have such structures only during CT scan. In our application, both image modalities are used in conjunction with contrast agent in order to better visualize blood vessels, in particular the aorta. However, its flow through the human body is completely different. While it is normally injected through the inguinal in CT, with the single injections harmonized with the pulse, it comes through the upper arm intra-operatively without any timing at all. In the test data of a real patient, this led to a strange phenomenon. While the aorta was well-contrasted in the fluoroscopy, it was relatively invisible in CT, compared to other blood vessels like the pulmonary vein crossing in front of the aorta. The difference can be seen in figure 5.4.

5.3.2.2 Deformation

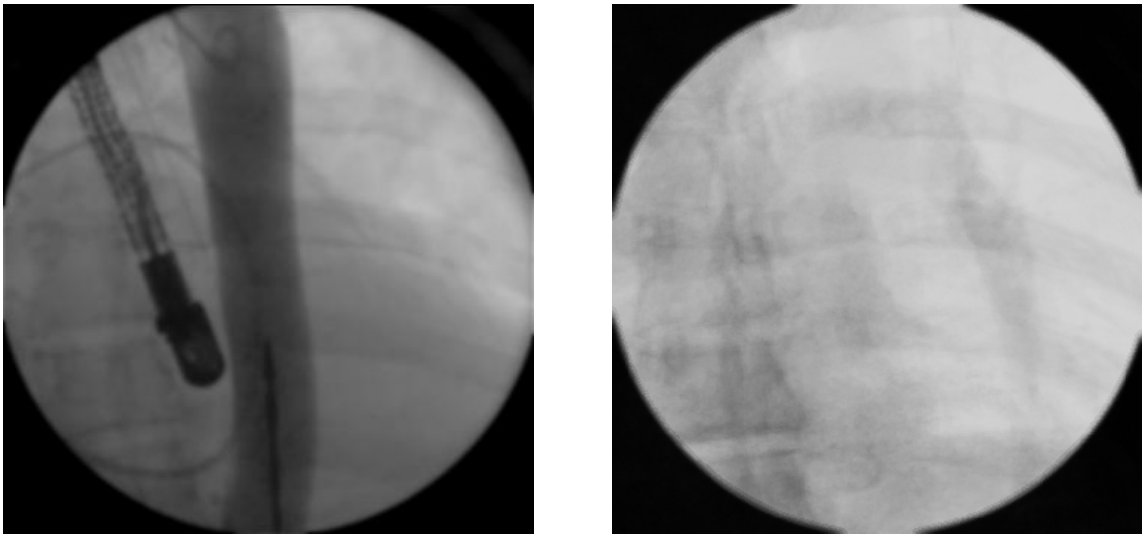
Due to image acquisition taking place at different locations and consequently also different points of time, especially soft-tissue structures like organs or blood vessels tend to translation and deformation. Such occurrences may also be the consequence of a slightly different patient bed and even affect bony structures like the spine in the lower left corner of the figures 5.4 a and b. The ribs, however, show a very good congruence.

5.3.3 Proposed Algorithm

Photons coming from the X-ray source traverse the human body along a ray. Some of the photons are absorbed on their way, the others hit the detector unit. The fraction of scattered radiation is relatively small at X-ray energy levels [67] and may be ignored. Reflection does not take place at all. Thus, X-ray fluoroscopy is ideally modelled by the ray-casting algorithm described in 4.2.1, and shading is omitted.

It has already been mentioned, however, that a high performance will be needed since lot of DRRs have to be generated for the interactive registration as well as for the iterative optimization algorithm. Therefore, only efficient implementations, e.g. based on hardware-accelerated texture-mapping (see 4.2.2.2) or based on the shear-warp factorization, of the viewing transformation may be taken into consideration. An alternative could be the interpolation of training images in eigenspace representation as described in 4.2.2.3.

Another requirement especially for the automatic registration algorithm is the computation of high-quality DRRs, since fine structures visible in the original image should be recognizable by the quality measure also in the DRR. However, its quality lags behind anyway due to the lower resolution of voxel data (see 5.3.1.2) and it is wise not to amplify these



(a) (b)

Figure 5.4: Images used in automatic registration. DRRs are created iteratively, until they resemble the radiograph. The aorta is contrasted much better in the radiograph, even though contrast agent has also be used during CT scan. The vertebrae are slightly displaced. (a) Radiograph, (b) DRR

effects by accepting further simplifications, as long as the resulting computational effort remains justifiable. Under these circumstances, and the fact that a good hardware platform[3] is available, volume rendering based on 3D texture-mapping seems to be a good choice.

Due to the issues described in 5.3.2, which compromise the similarity of fluoroscopic image and DRR, a mechanism has been implemented which makes it possible to restrict the automatic registration to a certain *region of interest* (ROI) containing mainly bony structures, since these can be assumed to be more rigid than softer parts. A good quality measure (see chapter 6) should be able to cope with differences in the images, up to a certain degree.

5.3.4 Selection of Transfer Functions

In order to compute a DRR that resembles a fluoroscopic image, the transfer functions $\tilde{\mathbf{c}}(\mathbf{s})$ for color densities and $\tau(\mathbf{s})$ for extinction densities have to be chosen correctly. In other words, materials imaged in a fluoroscopic image must also be imaged in the corresponding DRR, notabene at the same contrast, thus resulting in similar histograms for both modalities.

5.3.4.1 Correspondence between CT Data and DRR

Since CT data covers a broader range in subject contrast (see 5.1) than a fluoroscopic image does, it is principally possible to extract the necessary data and re-compose a new image

[3] Dual INTEL XEON architecture, *nVidia Quadro 4 900 XGL* graphics, 1.5 GB RAM

using a virtual X-ray source and detector system.

At first, we have to understand, how the actual linear attenuation coefficients $\mu_{CT,i}$ are related to the CT voxel data s, introduced in 4.2.1, which is commonly represented in Hounsfield Units[4] HU. They are related by the affine line transformation

$$HU_i = \frac{\mu_{CT,i} - \mu_{CT,water}}{\mu_{CT,water}} \cdot 1000 \tag{5.6}$$

which results in a range $[-1000..3000]$ of valid Hounsfield Units, commonly represented by 12 bit. This formula represents a kind of standardization for different CT scanners, since air[5] is always represented by $HU_{air} = -1000$ and water by $HU_{water} = 0$ respectively. Therefore, data acquired by different scanners using different X-ray energies can be compared, at least in the range of HU_{water}. For a material very different from water, however, the corresponding Hounsfield Unit may differ due to specific scalings for HU in equation 5.6 and the reasons explained in 5.3.1 [1]. Bone for example has much higher attenuation than water, and therefore, its measured linear attenuation coefficient will be much higher scanning at lower energy levels than at higher energy levels. An alternative representation, used by some older scanners, is given by CT numbers

$$CT_i = \frac{\mu_{CT,i}}{\mu_{CT,water}} \cdot 1000, \tag{5.7}$$

lying in a range $[0..4000]$. In any case, $\mu_{CT,i}$ are linearly related to the 3D voxel data.

Together with the theory of gradation curves explained in 5.1.1, we are now able to interpret the useful exposure range of the CT scanner in such a manner that the measuring of a C-arm can be simulated. For this, we consider a ray of photons at a fluoroscopic energy level with an initial intensity $I = I_0$. According to the attenuation law introduced in chapter 4, in each ray segment i, the intensity decreases by

$$I_i = I_{i-1} \cdot e^{-\mu_{X,i} \cdot d_i}, \tag{5.8}$$

with linear attenuation coefficients $\mu_{X,i}$ and a segment length d_i, which can be assumed constant in the case of equidistant segment spacing. This yields the final intensity I_n of the ray hitting the detector plate. More compactly, this is expressed by

$$I_n = I_0 \cdot e^{-\sum_{i=1}^{n} \mu_{X,i} \cdot d}. \tag{5.9}$$

Linear attenuation coefficients differ considerably between CT and X-ray fluoroscopy. However, they are linear dependent, if non-linearities introduced by broader energy ranges are ignored, as described in 5.1.2. Therefore, we can replace $\mu_{X,i}$ by $\mu_{CT,i}$ in 5.9. This yields

$$I_n = I_0 \cdot e^{-\sum_{i=1}^{n} \left(a_1 \cdot \mu_{CT,i} + b_1\right) \cdot d}, \tag{5.10}$$

with parameters a_1 and b_1 expressing the affine line transformation. Exposure is intensity I_n over time. From the gradation curve, we know that the logarithm of the radiograph's opacity

[4] Named after Sir Godfrey N. Hounsfield who constructed the first CT scanner in 1972 and obtained the Nobel Prize in 1979 for his invention

[5] The attenuation of air is almost zero

is linearly related to the logarithm of exposure, at least for the linear part of the curve. Thus,

$$\log opacity = a_2 \cdot \log \left(I_0 \cdot e^{-\sum_{i=1}^{n} (a_1 \cdot \mu_{CT,i} + b_1) \cdot d} \right) + b_2, \tag{5.11}$$

with an additional affine line transformation with parameters a_2 and b_2. In a last step, we derive the final pixel intensity values of the DRR from the opacity of the virtual X-ray film material or C-arm detector respectively. The digitization is linearly related to the intensity of visible light transmitted by the film radiograph. Furthermore, the transmission of visible light is inversely related to the opacity of the film. This finally yields the overall correspondence between intensity values in the DRR and linear attenuation coefficients acquired by a CT scan

$$\log \left(\frac{1}{a_3 \cdot I_{DRR} + b_3} \right) = a_2 \cdot \log \left(I_0 \cdot e^{-\sum_{i=1}^{n} (a_1 \cdot \mu_{CT,i} + b_1) \cdot d} \right) + b_2, \tag{5.12}$$

including one more affine line transformation defined by a_3 and b_3. By application of equation 5.6, simplification and combination of the constants a_i, b_i, I_0 and $\mu_{CT,water}$, equation 5.12 can be condensed to the following simple form.

$$\ln I_{DRR} = a \cdot \sum_{i=1}^{n} HU_i + b \tag{5.13}$$

This states a linear dependence of DRR intensity values from either HU or CT numbers, described by constants a and b. A very similar formula has been derived by Gottesfeld Brown [8]. Although it introduces some simplifications by ignoring non-linearities, a good quality measure, as described in chapter 6, should be able to compensate for such differences [76, 58].

5.3.4.2 Formulation of Transfer Functions

It is now possible to formulate suitable transfer functions which associate appropriate primary colors and extinction coefficients to each voxel $\vec{v}$. This process is also commonly known as selection of a *window* for visualization. In radiology, predefined windows exist for the visualization of detailed structures, e.g. in the lungs. Contrast is enhanced for the range of attenuation coefficients belonging to the tissue of interest while other attenuation coefficients are completely excluded from visualization. Our task is very similar, since we have to find the window which just visualizes linear attenuation coefficients lying within the useful exposure range of X-ray fluoroscopy and ignores all others.

A window is defined by two parameters specifying its location and width on the axis of linear attenuation coefficients μ_i. Those parameters may be identified with the constants a and b from equation 5.13. Colors are associated to the data within the window used for visualization. Since the useful exposure range is mapped linearly by the gradation curve, we choose a linear grey-scale gradient ranging from white to black. This yields the association of primary colors $\tilde{\mathbf{c}}(\mathbf{s})$. We also need transparency values $\tau(\mathbf{s})$. Those shall be independent of the actual length of a ray segment or voxel respectively. Thus, each grey value defined in the window is assigned a transparency value ranging linearly from black to white, whereas black means full transparency while white is obtained, when absolutely no radiation hits the X-ray detector. In order to adapt this assignment to the actual voxel size, an overall

scaling factor α has to be chosen accordingly. Imagine that α is set to one. This would mean that a single voxel representing bone material absorbs almost the complete radiation in its direction. Experimentally, a window ranging from $-150\ HU$ to $400\ HU$ has been determined to reflect the properties of the C-arm system best. With our demo data, $\alpha = 0.01$ is a good choice.

5.4 Implementation

For implementation of the described imaging techniques in CANP, the software environment AMIRA is used. It aims at processing and visualization of scientific image data and already contains a great deal of the functionality described in chapter 4. It implements a modular architecture with strict separation in three different kinds of *modules*.

- *Data modules* are available for encapsulation of 2D or 3D image data. Scalar fields, as described in chapter 4, are represented by class `HxUniformScalarField3`, whereas surface representations are encapsulated by `HxSurface`.
- *Display modules* visualize data objects. Different kinds of volume rendering techniques are available. Direct volume rendering is provided by the `Voltex` module and surfaces are rendered by the `SurfaceView` module, respectively.
- *Compute modules* care for the transformation of data between different representations. E.g. the `LabelVoxel` module is used for segmentation of a 3D scalar field.

The environment combines a user-friendly GUI for its included modules. Thereby, the modules are represented graphically in the so-called *object pool* and interact via *connections* which may be set visually by the user. Configurational settings are applied to the *ports* of a module. Furthermore, AMIRADEV provides a class library which enables the integration of modules in software projects, as well as writing additional modules, which can then also be used in the GUI. An in-depth consideration of the framework would go beyond the scope of this thesis. Architectural aspects of CANP including those related to AMIRA are described by Groher [25]. Information can also be found in the product documentation [39]. In the following, only the parameterization of modules shall be described that have been used for DRR generation.

Preparation of 3D Patient Model The axial slices acquired by a CT scan are represented in the "DICOM" format commonly used in medical imaging [55]. These can be imported by AMIRA. It may be necessary to attach the `Arithmetic` module in order to transform *stacked coordinates* with arbitrary slice distance to *uniform coordinates*, so that the `Voltex` module can be used later. Then, a re-sampling is applied by means of attaching the `Resample` compute module, since the original resolution of $512 \times 512 \times z$ voxels with z the number of axial slices is to large for interactive handling. Therefore, each dimension has been down-sampled by a factor of two and this has been accomplished using the *Lanczos* interpolation scheme. The resulting scalar field is saved in the *amira mesh* file format.

Direct Volume Rendering AMIRA implements the texture-mapping based approach chosen for generation of DRRs in its `Voltex` module. Both, 2D and 3D texture-mapping modes are supported. Due to quality reasons, the 3D mode is used in CANP, using the maximum number of 255 viewport-aligned slices. No further down-sampling is applied in the context of the `Voltex` module, since the model has already been down-sampled before. The transparency factor α (see also 5.3.4.2) is used by the module to apply an additional scaling of the alpha-channel stored within each RGBA-tuple in the `Colormap` module. It compensates for different pixel intensity values depending on the number of viewport-aligned slices. A value of $\alpha = 0.01$ has been determined experimentally and yields the best results for the voxel size of our or demo data ($1.15625 \times 1.15625 \times 0.7mm$). The factor α and also the slice number may be changed interactively, since these steps do not require a reload of the texture memory. A modification of the color-map, however, implies a reload of texture memory, since the new transfer functions have to be applied to each voxel, at least in the case of pre-classification.

Representation of Transfer Functions Transfer functions are represented via the `Colormap` data module. It contains a sequence of RGBA-tuples, where every tuple specifies a color by a red, green and blue value in the RGB color model. Each value ranges from 0.0 to 1.0 and is represented by a floating point value. A fourth value, the so-called alpha value, defines opacity. It also ranges from 0.0 to 1.0, where 0.0 means that the color is fully transparent, and 1.0 that the color is fully opaque. It should not be confused with the overall scaling factor α introduced above, which is applied in addition the color-specific α values in order to compensate for the number of slices or texture maps projected to the image plane. Usually, 256 different RGBA-tuples are used, other values are obtained by interpolation. Besides those tuples, the raw RGBA values, the color-map stores two additional parameters defining the position and range within valid scalar values used for color interpolation. This resembles the window described in 5.3.4. Color lookup requests for an argument smaller than the minimum color, evaluate to the first color-map entry. Requests for an argument greater than the maximum coordinate, evaluate to the last entry. The window settings derived in 5.3.4 have been applied here.

Generation of DRRs The `Voltex` module described above is responsible for volume rendering. In order to achieve DRRs in "BMP" format, snapshots are made from the viewer which contains the rendered scene. For this purpose, an AMIRA compute module named `DRRfactory` has been implemented as a wrapper around the viewer's `snapshot()` method, which stores a file in "BMP" format at the specified path. The viewer itself is parameterized with the intrinsic camera parameters of the C-arm derived in chapter 3. AMIRA internally uses the OPEN INVENTOR rendering engine for visualization. Perspective projections are described by class `SoPerspectiveProjection`. In contradiction to the intrinsic camera parameters defined in chapter 3, it is configured by the following settings, which become clear from figure 5.5.

- *Aspect ratio*: ratio between camera viewing width and height
- *Height angle*: angle spanning the horizontal dimension of the image plane
- *Near distance*: the distance from the camera viewpoint to the near clipping plane.

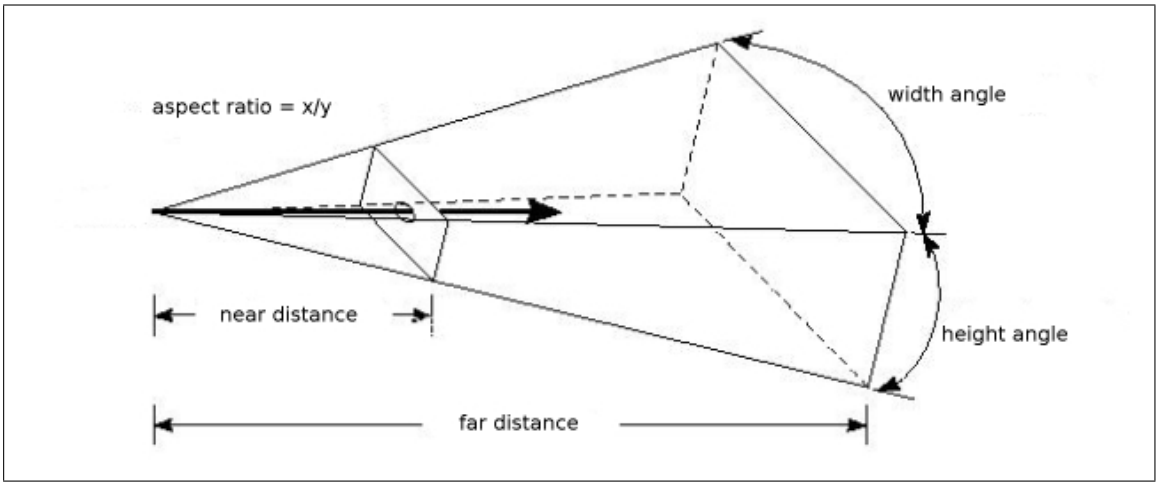


Figure 5.5: Perspective geometry of the OPEN INVENTOR camera model

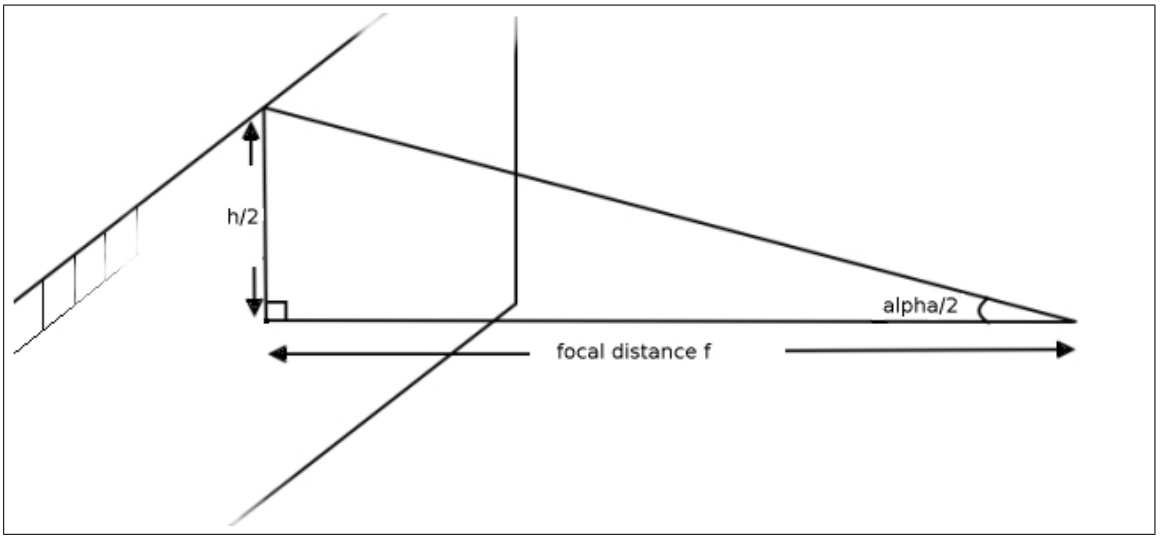


Figure 5.6: Adaptation of C-arm intrinsic parameters to OPEN INVENTOR camera model

- *Far distance*: the distance from the camera viewpoint to the far clipping plane.

Far and near distance are not important, since they do not influence the geometry but define a region of interest for rendering. The aspect ratio has been set to one, since the radiographs also have a square format of 980×980 pixels. Perspective is configured with the height angle parameter only. A principal point may not be specified. Fortunately, the derived parameters $p_x = 514$ and $p_y = 490$ are near to the center of the image at 480 pixels in horizontal and vertical direction. The height angle has been derived from the focal length by the following calculation. Let α be the height angle, f_{pix} the focal length in pixels and $h = 980$ the height of the image.

$$\tan\frac{\alpha}{2} = \frac{h/2}{f_{pix}} \Rightarrow \alpha = 2 \cdot \tan^{-1}\frac{h/2}{f_{pix}} \tag{5.14}$$

This calculation is visualized in figure 5.6 and results in a height angle parameter of $\alpha = 2.81643885°$. In this manner, DRRs with a resolution of 358×358 pixels are produced. A much higher resolution would not make any sense due to the limited resolution of the volume data. Further processing of DRRs acquired by the `DRRfactory` module is described in chapter 6.

6 Registration

Now that we are able to create DRRs from a CT model located between X-ray source and detector of the virtual C-arm, the actual registration algorithm can be set up. The final goal is a description of the correspondence between related points within the patient on the one hand and the CT model on the other hand. This is called registration.

At first, the virtual C-arm setup including its coordinate system will be described in section 6.1. Different iterative optimization algorithms have already been described in chapter 3. A suitable one will be selected here. A quality measure is necessary for this algorithm in order to evaluate the similarity of a certain radiograph with the current DRR and thus, the current registration quality. By minimizing this measure, the registration will be optimized. Suitable candidates will be described in section 6.2. Next, the implementation of algorithms and measures shall be outlined in section 6.3. Finally, performance and convergence tests will be presented in section 6.4, using this implementation.

6.1 Registration Setup

The patient is lying between X-ray source and detector of the C-arm in operating theater and the pose of his body is described in the *camera coordinate system* of the C-arm, though it is unknown at the moment. The direct volume rendering setup described in section 5.3 constitutes a *virtual C-arm setup*, since it simulates the properties of the real C-arm, e.g. with respect to its intrinsic camera parameters (see chapter 3) or the reconstruction of X-ray grey values from CT-data (see 5.3). Thus, we assume the same camera coordinate system for both, the real and the virtual C-arm. It's origin is defined to lay on the negative z-axis of the *global coordinate system*, with the viewing axis in direction to the origin. The *object coordinate system* of the CT model is initialized by CANP in such a manner that the bounding box center, also called the center of gravity, of the CT model is located in the origin of the world coordinate system. The radiograph/DRR lays, orthogonal to the positive y-axis, on the image plane of the real/virtual C-arm. The y-axis intersects the DRR in the principal point in the middle of the image. The principal setup becomes clear from figure 6.1. At the moment, we only have a rough approximation of the patient's location from interactive pre-registration [25].

The initial pose of the CT model at the origin of the world coordinate system has now to be changed accordingly by a rigid transformation. This has been determined in chapter 2, along with an intensity-based approach for registration. By changing the pose of the CT model, also the DRR changes. If the DRR resembles the radiograph, the CT model's pose coincides with that of the patient in the global coordinate system and information may be exchanged between both modalities. Once initialized to global coordinates as described above, the pose is described by a matrix

$$M = \begin{pmatrix} R & \vec{t} \\ \vec{0}^T & 1 \end{pmatrix} \tag{6.1}$$

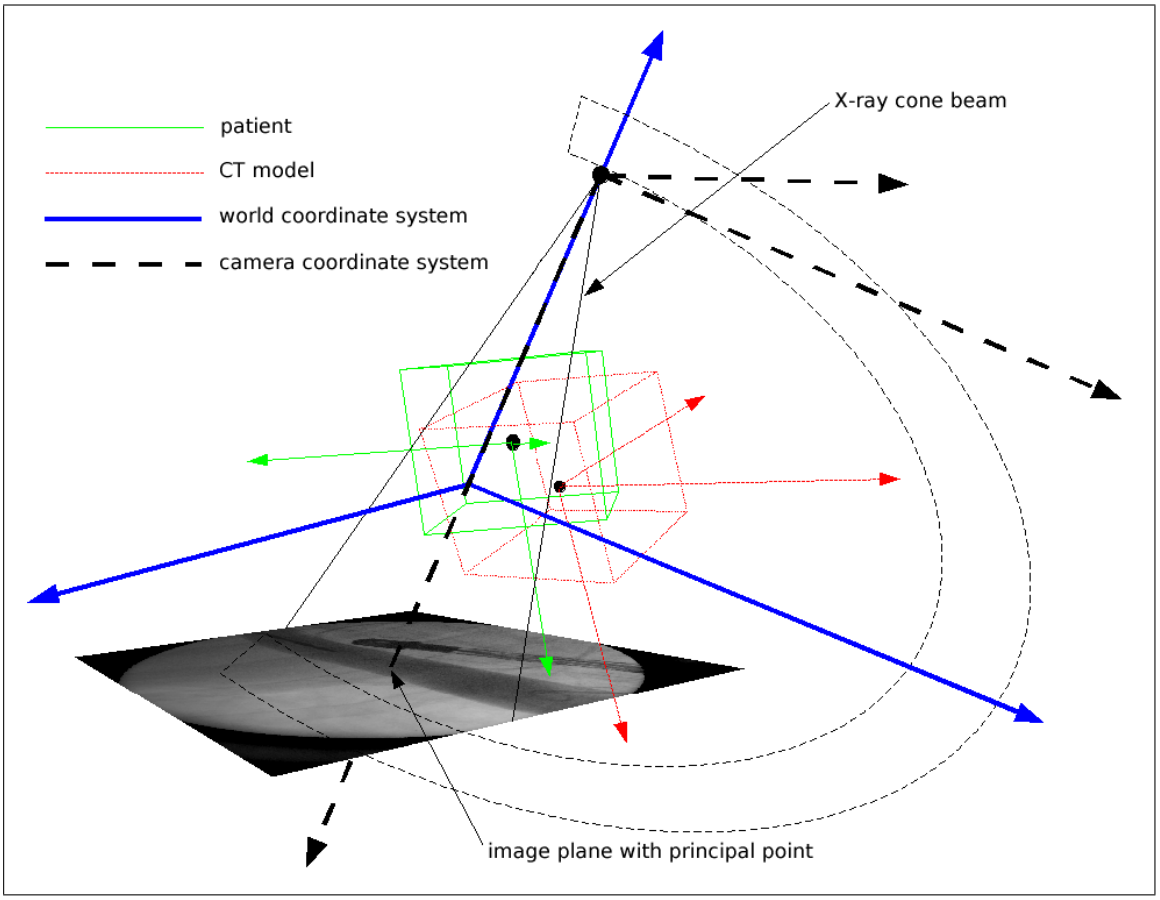


Figure 6.1: Virtual C-arm setup with all coordinate systems. The CT model's coordinate system is to be aligned with the patient's coordinate system.

in homogeneous coordinates, where $\vec{t} \in \mathbb{R}^3$ is a translation vector and $R \in \mathbb{R}^{3\times3}$ is the overall rotation matrix

$$R = R_\phi \cdot R_\theta \cdot R_\psi \tag{6.2}$$

composed from rotations about the x, y and z-axis of the global coordinate system by rotation angles ϕ, θ and ψ respectively. Thus, M has six degrees of freedom. Since the bounding box center is located in the origin, rotations are applied with respect to this point.

Due to the interactive pre-registration, the pose of the CT model already roughly approximates the patient's actual pose, when the automatic optimization is started [25]. An iterative algorithm now optimizes the overlay of radiograph and DRR by evaluating the influence of different transformation matrices M on the similarity of both images. Two algorithms have been implemented in CANP, the best neighbor and the weighted best neighbor optimizer. They have been proved to be suitable for similar registration problems in literature [58, 76, 63]. Their principles have already been described in section 3.5. Similarity has to be optimized with respect to a six-dimensional parameter space corresponding to the six degrees of freedom of transformation matrix M. The increments which shall be applied to the translational and rotational parameters have to be chosen in such a man-

ner that its impact on change in the DRR is roughly identical. This is especially important for the best neighbor optimizer, since it always selects that parameter for final refinement, which yielded the best improvements in the preceding evaluation loop. However, t_z, describing movement on the C-arm's viewing direction, has much lower impact than e.g. t_x or t_y. Thus, larger steps in z-direction have to be performed in order to give the algorithm a chance for optimizing t_z. With respect to this boundary condition, an initial refinement vector $\Delta = (1cm, 1cm, 10cm, 4°, 4°, 4°)$ has been chosen. Four optimization loops will be performed, whereby the step-widths will be halved before each iteration. This parameterization has been proposed by Penney, though with a different Δ [58].

6.2 Quality Measures

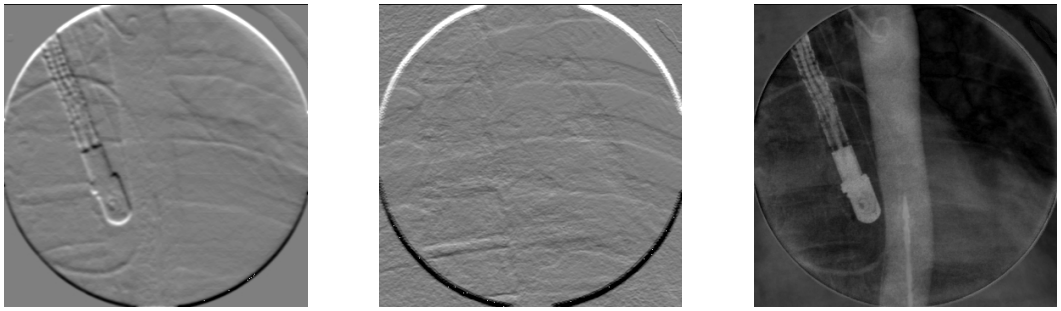
A quality measure is a mathematical formula expressing similarity between two modalities. A wide range of quality measures exists and even more will probably be defined in the future. Each measure has distinct properties and therefore is more or less suited for comparison, depending on the type of modalities existing in a certain application.

Two quality measures have been proposed concordantly by Penney and Wein [58, 76] in the context of 2D-3D registration, *gradient difference* and *pattern intensity*. While the former is based on the gradients of image data, the latter uses a difference image. Both measures have been implemented in CANP and tested in the context of images arising during stent-implantation.

Pixel intensity values will be denoted by tuples $(i, j) \in M$ with M the underlying image model. The definition of an image model within the context of a certain measure expresses the necessity that the images to be compared must have the same dimension and bit-depth. Furthermore, a region of interest is able to restrict the pixels considered by the measure and is therefore also included in the model. Some measures use immediate bitmap images I_{imm}, e.g. those supplied by the C-arm or the volume renderer. Other modalities may also be the difference image I_{dif} created by pixel-wise subtraction of the two images to be compared or also gradient images $I_{grad,h}$, $I_{grad,v}$ which contain edges detected by application of a 3×3 Sobel template, both in horizontal and vertical direction. The quality measures will be normalized to the interval $[0..1]$, with 0 in the case of exact conformance, for interchangeability reasons.

6.2.1 Cross Correlation

Cross correlation or *normalized cross correlation* is widely-used measure, not only for intensity-based registration. Based on mean pixel intensities $\overline{I_{imm,1}}$ and $\overline{I_{imm,2}}$ the measure ignores linear dependent differences $I_{imm,2}(i, j) = aI_{imm,1}(i, j) + b, \forall(i, j) \in M$ in the intensity values. Therefore, differences in both, contrast and brightness, will not influence the error E_{CC}. However, the contribution of one pixel heavily depends on the corresponding intensity difference and thus few pixels, maybe belonging to some interventional instruments, can have



(a) (b) (c)

Figure 6.2: Images used by pattern intensity and gradient difference quality measures. For the gradient images, intensity values have been truncated to eight bit and 128 has been added in order to visualize negative gradients. (a) Horizontal gradient of blurred radiograph, (b) Horizontal gradient of DRR, (c) Difference between radiograph and DRR

substantial effect on the measure [58].

$$\begin{aligned} E_{NCC} &= \frac{1}{\sigma_1 \sigma_2} \frac{1}{n} \sum_{(i,j \in M)} \left(I_{imm,1}(i,j) - \overline{I_{imm,1}}\right) \cdot \left(I_{imm,2}(i,j) - \overline{I_{imm,2}}\right) \\ &= \frac{\sum_{(i,j \in M)} \left(I_{imm,1}(i,j) - \overline{I_{imm,1}}\right) \cdot \left(I_{imm,2}(i,j) - \overline{I_{imm,2}}\right)}{\sqrt{\sum_{(i,j \in M)} (I_{imm,1}(i,j) - \overline{I_{imm,1}})^2} \cdot \sqrt{\sum_{(i,j \in M)} (I_{imm,2}(i,j) - \overline{I_{imm,2}})^2}} \end{aligned} \tag{6.3}$$

The denominator contains the product of the standard deviations σ_1 and σ_2 belonging to the immediate image modalities $I_{img,1}$ and $I_{img,2}$.

It is also possible to apply the formula only locally for small windows within both images and use the sum over all windows as an indicator for similarity. This is called *sum of local normalized cross correlation*. Wein achieved very good registration results using this approach, however too late for our implementation [76].

6.2.2 Pattern Intensity

A difference image $I_{dif} = I_1 - s \cdot I_2$ with s an intensity scale factor should have constant intensity in the case of I_1 and I_2 being perfectly registered. Changes in intensity within I_{dif} are interpreted as part of some structures belonging to either one of the images, or both. *Pattern intensity* tries to detect such structures by considering a pixel to be a part of them, if it shows a considerably different intensity from pixels in its neighborhood. This is parameterized by a radial distance r. The corresponding error is given by

$$E_{PI} = \sum_{(i,j) \in M} \sum_{(v,w) \in N} \frac{\sigma^2}{\sigma^2 + \left(I_{dif}(i,j) - I_{dif}(v,w)\right)^2} \tag{6.4}$$

where

$$N = \{(k,l)\,|r^2 < (i-k)^2 + (j-l)^2\} \tag{6.5}$$

specifies the neighborhood set of pixels centered at pixel (i,j). The error E_{PI} asymptotically tends to zero when the number of structures is increased. A maximum, depending on the resolution of I_{dif} and also the choice of r, is reached when there are no structures at all. The constant σ is used for weighting the function in order to reduce the influence of small intensity differences caused by Gaussian noise. The measure is robust against large differences in pixel intensity due to its asymptotic behavior. This means that large intensity differences have roughly the same effect on the measure regardless of the actual magnitude. Structures with low spatial frequency are suppressed by a small radius r. This enables the measure to confine itself to detection of bony details in our application.

The factor σ influences the weighting of intensity differences between a pixel and a certain neighbor. With small σ, the measure becomes much more sensible to very small intensity differences in comparison to larger differences. With big values however, the measure is sensible to differences in magnitudes. Choosing $\sigma = 10$ has already been proposed by [58] and seems to be a good compromise for difference values ranging in the interval $[0..255]$ available with a bit-depth of eight. These coherences are visualized in figure 6.3.

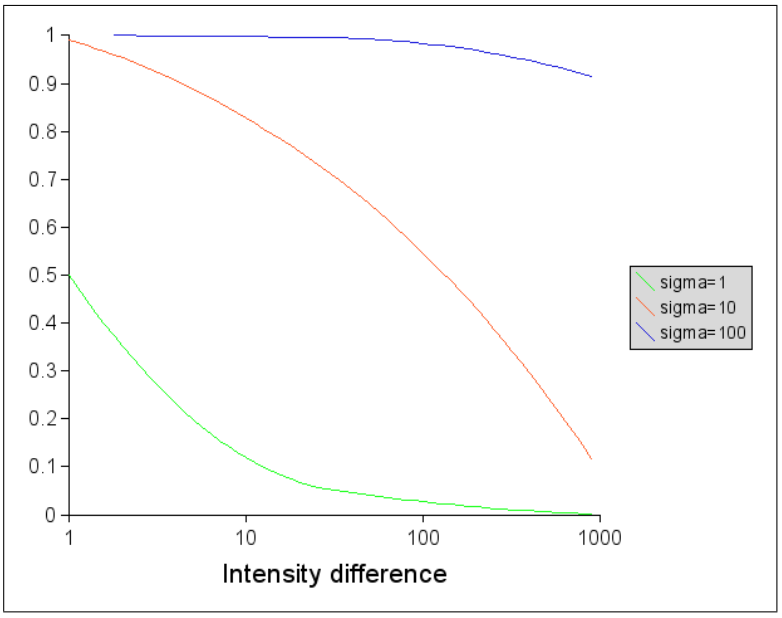


Figure 6.3: Correspondence of σ and d in the term $\frac{\sigma^2}{\sigma^2+d}$ which is used in the pattern intensity and gradient difference measures.

The normalization of E_{PI} in 6.4 to an interval $[0..1]$ is accomplished by

$$E_{NPI} = 1 - \frac{1}{n_{ROI}} \cdot \sum_{(i,j\in M)} \sum_{(v,w)\in N} \frac{1}{n_r} \cdot \frac{\sigma^2}{\sigma^2 + (I_{dif}(i,j) - I_{dif}(v,w))^2} \tag{6.6}$$

with n_{ROI} the number of pixels lying in the chosen region of interest and n_r the number of pixels lying in the circle with radius r around pixel (i,j), thus in the neighborhood of pixel (i,j).

6.2.3 Gradient Difference

The *gradient difference* measure has also been successfully tested in the context of 2D-3D rigid registration [58, 76]. In contradiction to pattern intensity, it is based on gradient images in both, horizontal and vertical direction. Therefore, four images $G_{DRR,h}$, $G_{DRR,v}$, $G_{X,h}$ and $G_{X,v}$ have to be considered, two for each image modality. The advantage of gradient-based measures is their insensitivity to soft-tissue deformation since they produce almost no edges in the gradient image. The usage of two images provides the possibility to consider the direction as well as the magnitude of the gradients. The measure is defined by

$$E_{GD} = \sum_{(i,j)\in M} \frac{\sigma^2}{\sigma^2 + (G_{X,h} - s \cdot G_{DRR,h})} + \sum_{(i,j)\in M} \frac{\sigma^2}{\sigma^2 + (G_{X,v} - s \cdot G_{DRR,v})} \tag{6.7}$$

with a scale factor s in order to compensate for intensity differences in the gradient images. It is set to one since the image modalities have been prepared to result in similar images as described in 5. The same $\frac{\sigma^2}{\sigma^2+d}$ pattern is used as in pattern intensity. Again, this ensures that small differences in the gradient images are not overestimated and thereby makes the measure robust to thin line structures arising from bones. σ is set to the variance of the corresponding fluoroscopy gradient images though the corresponding variance of the DRRs gradients' should be quite similar. A normalized form of the measure is given by

$$\begin{aligned} E_{NGD} = 1 - \frac{0.5}{n_{ROI}} \cdot & \sum_{(i,j\in M} \frac{\sigma^2}{\sigma^2 + (G_{X,h} - s \cdot G_{DRR,h})} \\ & + \sum_{(i,j)\in M} \frac{\sigma^2}{\sigma^2 + (G_{X,v} - s \cdot G_{DRR,v})}, \end{aligned} \tag{6.8}$$

since E_{GD} in formula 6.7 is twice the number of pixels within the chosen region of interest in the worst case.

6.3 Implementation

Several functionally independent components are necessary in order to perform the automatic 2D-3D registration. These are

- Generation of DRRs
- Image file handling
- Quality measuring
- Parameter optimization
- Application logic

The first one, generation of DRRs, has already been explained in 5.4 and is only mentioned here again for the sake of completeness. This chapter focuses on the registration itself and

therefore, the other four items of the list will be explained here. The second and the third issue are combined in the `Bitmap` library, since the latter makes extensive use of data structures provided by the former. Parameter optimization is encapsulated in the camera calibration framework `libcc` that has already been mentioned in chapter 3. The application logic is provided as an AMIRA module which has been a non-functional requirement in this project. The following subsections will cover important implementational aspects in detail. However, not the whole source code may be presented here.

6.3.1 Bitmap Library

Once X-ray images[1] and DRRs are available, we need a library for their handling. This has been achieved in terms of the `Bitmap` library which actually is a procedural implementation written in C. Grey-scale and gradient images are represented by the following data structures:

```
typedef struct {
    unsigned int uiWidth;
    unsigned int uiHeight;
    unsigned char ** ppucImageData;
} GREYSCALEIMAGE;

typedef struct {
    unsigned int uiWidth;
    unsigned int uiHeight;
    int ** ppiGradientData;
} GRADIENTIMAGE;
```

`GREYSCALEIMAGE` is an encapsulation of a grey-scale image with width `uiWidth` and height `uiHeight`. `ppucImageData` is a two-dimensional array with rows in the first and columns in the second dimension. `GRADIENTIMAGE` is the corresponding encapsulation of a gradient image. Instead of storing eight bit grey-scale, `ppiImageData` contains integer values, since the gradients may have positive or negative direction and do not necessarily fit in the interval $[-128..127]$.

Load and store functionality for grey-scale images is provided by the procedures `readBMPfile()` and `storeBMPfile()`. Only 24 bit RGB images are able to be loaded or stored since this format is provided by the X-ray images exported from the C-arm and also the snapshots from the viewer in AMIRA. The conversion from and to grey-scale values is achieved by the following equation defined by the NTSC TV standard

$$Y = 0.3 \cdot R + 0.59 \cdot G + 0.11 \cdot B \tag{6.9}$$

with Y being the desired luminance or grey-scale value [20]. Via `deleteGreyscaleImg()`, an instance of type `GREYSCALEIMAGE` is removed from memory.

[1] At the moment, fluoroscopic images can be exported via a smart-drive only, since no other interface has been provided yet by GENERAL ELECTRIC

Since images of both modalities, DRR and X-ray, need to be available at the same resolution, bilinear scaling has been implemented in `scaleBMPfileBilinear`. Furthermore, it is necessary to blur the X-ray image, since generation of DRRs is impossible at the same quality, which leads to a difference in sharpness disturbing the quality measures. This is achieved by calling `blurImage()` which is done 15 times in CANP. This number has been determined by visual examination. Internally, the following 3×3 blurring kernel will be applied to each pixel.

```
typedef float KERNEL_3X3[3][3];
static const KERNEL_3X3 tBlurKern = \
    {{0.11,0.11,0.11},{0.11,0.11,0.11},{0.11,0.11,0.11}};
```

After that, grey-scale images can be converted to gradient images by applying the following two 3×3 Sobel kernels via the `applyKernel()` procedure.

```
static const KERNEL_3X3 tSobKernX = \
    {{-1,0,1},{-2,0,2},{-1,0,1}};
static const KERNEL_3X3 tSobKernY = \
    {{1,2,1},{0,0,0},{-1,-2,-1}};
```

The first one detects horizontal edges, the second one vertical edges, respectively. Both are necessary for the gradient difference quality measure and may also be stored stored via `storeGradientFile()` for visualization purposes only. Internally, this procedure truncates and shifts the gradient values so that they can be represented by eight bit characters used in the data type `GREYSCALEIMAGE`. After usage, they are removed from memory via `deleteGradientImg()`. The difference images needed by the pattern intensity measure are calculated by `calcSubtraction()` and the result will be still of type `GREYSCALEIMAGE`.

The quality measures pattern intensity and gradient difference have also been included in this library. The procedures `calcPatternIntensityMeasure()` and `calcGradientDifferenceMeasure()` implement the respective weighted form of the measure, as expressed in equations 6.6 and 6.8. Besides the necessary images (one difference image for pattern intensity and four gradient images for gradient difference), a region of interest may be specified which will restrict error evaluation to the pixels lying within the defined square. For pattern intensity, the radius r, which specifies the neighborhood, and also the weighting factor σ have to be specified. For gradient difference, weighting factors have to be specified for the horizontal and the vertical image. These can be calculated by `calcROIvariance256()` in advance for the same region of interest that will also be specified in `calcGradientDifferenceMeasure()`. Both measure functions return a float value in the interval $[0..1]$ with zero in the case full image correspondence. Their implementation is straight-forward and therefore omitted here.

6.3.2 A 2D-3D Registration Module for AMIRA

The modules described in the previous subsections are now integrated in an AMIRA module called `AutoRegistration`. The corresponding class is named `HxAutoRegistration`

and represents a compute module, thus, it is derived from class `HxCompute` and we have to implement the `compute()` method which is called whenever GUI interaction with the module takes place or other modules request some computations to be performed. For more information about the AMIRA programming paradigm and its class hierarchy please refer to Groher [25]. Furthermore, this module inherits from class `CcRigidRegistration` which is part of the camera calibration framework `libcc` and determines an interface for rigid registration to be implemented. The inherited abstract method is called `estimateTransRot()` and describes an arbitrary rigid registration which could be done fully automatically in this module, e.g. by combining a marker based pre-registration with an intensity based optimization. However, the pre-registration is performed manually at the moment via the AMIRA module `RegistrationDisplay` implemented and described by Groher who also describes the `libcc` package in detail [25].

Our focus now lays on the implementation of the automatic optimization of a given pre-registration. This means that in figure 6.1, the CT-model is already roughly aligned to a proper position, expressed in an instance of `SoTransform` named `pInitTransform`. This transformation is now to be optimized iteratively towards the hopefully optimal solution. At first, it shall be described, how an arbitrary transformation is applied to the model. Since the initial transform in `pInitTransform` is already given, we take this as a basic value for additional transformations calculated by the algorithm. The calculation of an overall transformation based on `pInitTransform` is implemented in a private method which shall be described next.

```
void HxAutoRegistration::setRelativeTransform( float * params )
{
    // setup rotation matrices Rx, Ry and Rz using given angles
    SbMatrix Rx, Ry, Rz;
    SbRotation rotX( SbVec3f( 1, 0, 0 ), params[3] );
    SbRotation rotY( SbVec3f( 0, 1, 0 ), params[4] );
    SbRotation rotZ( SbVec3f( 0, 0, 1 ), params[5] );
    Rx.setRotate( rotX );
    Ry.setRotate( rotY );
    Rz.setRotate( rotZ );
    // setup translation matrix T using the given translation
    SbMatrix T;
    T.setTranslate( SbVec3f( params[0], params[1], params[2] ) );
    // get initial transformation
    SbMatrix transSpace, transSpace_inv;
    pInitTransform->getTranslationSpaceMatrix( transSpace, \
        transSpace_inv );

    // translate to origin in object space
    McVec3f origin = input3D->getBoundingBoxCenter();
    SbMatrix Tc;
    Tc.setTranslate( SbVec3f( origin[0], origin[1], origin[2] ) );

    /* setup modified transformation in addition to the initial
       transformation */
```

```
    SbMatrix M = Tc.inverse() * Rx * Ry * Rz * Tc * transSpace * T;
    // apply new transformation
    input3D->setTransform( M );
}
```

The only parameter is an array containing the six rigid body transformations $(t_x, t_y, t_z, \phi, \theta, \psi)$. According to these, a translation matrix `T` and three rotations matrices `Rx`, `Ry` and `Rz`, one for each axis of the object coordinate system, are calculated. Next, the already existing initial transformation is obtained from the `pInitTransform` object and stored in the `transSpace` matrix. Thereby, `getTranslationSpaceMatrix()` provides the full transformation from object to world coordinates. In order to apply the rotations to the center of the 3D model `input3D`, a translation of the bounding box center to the center of the object coordinate system has to be performed. This translation is acquired via `input3D->getBoundingBoxCenter()` and the translation matrix `Tc` is set up. In the OPENINVENTOR[2] transformation model, row vectors describing point coordinates are multiplied from the right hand side by transformation matrices. These are applied in three steps; at first scaling, then rotations and finally translations are carried out. Since we want to rotate around the bounding box center, which is provided in object coordinates by `getBoundingBoxCenter()`, we have to apply the rotations before the object is transformed to world coordinates via `transSpace`. Thus, a translation of the bounding box center to the origin (in object coordinates) `Tc.inverse()` is applied first, then our additional rotations `Rx`, `Ry` and `Rz`. Afterwards, the translation of the bounding box center is undone again by `Tc` and the existing transformation `transSpace` is applied. At last, our translation `T` is applied in world coordinates. This is aggregated in the assignment

```
SbMatrix M = Tc.inverse() * Rx * Ry * Rz * Tc * transSpace * T;
```

The resulting transformation `M` is applied to the 3D model `input3D` and a DRR resembling this pose may be calculated. Assuming a certain pose of the model within the virtual C-arm, DRRs are generated by making snapshots of the viewer in which the 3D model is rendered. This is performed by the AMIRA module `DRRfactory` which already has been described in section 5.4. Whenever its `compute()` method is called, it provides a snapshot using the current transformation of the model in the `/tmp/` directory.

We now have to setup and run the optimization algorithm described in section 6.1.

```
void HxAutoRegistration::estimateTransRot( void )
{
    ...
    // allocate memory for search space and optimized parameters
    float params[6];
    float optimizedParams[6];
    // initialize counters for callback functions
    deltaCount = 0;
    evalCount = 0;
    /* initialize rigid transformation that will be applied
```

[2]TGS OPENINVENTOR is used internally in AMIRA for representation and visualization of 3D scenes

```
       in addition to the initial transformation */
    params[0] = 0;
    params[1] = 0;
    params[2] = 0;
    params[3] = 0;
    params[4] = 0;
    params[5] = 0;

    // create instance of optimizer
    CcOptimization * myOptimizer;
    myOptimizer = new CcWeightedBestNeighbor();
    // initialize search space dimension
    myOptimizer->initDimension( 6 );
    // initialize search space
    myOptimizer->initSearchSpace( params );
    // initialize callback functions
    myOptimizer->initRefinement( this, &getDelta );
    myOptimizer->initCostFunction( this, &evalQuality );
    ...
    // start optimization
    myOptimizer->Optimize ( optimizedParams );
    ...
}
```

This code fragment only contains the important parts of the method. We may assume that the initial transformation `pInitTransform` is already initialized properly. It is shown, how the optimization algorithm is configured and invoked. The array `params` contains the transformation delta values Δp_i. This is the same vector that will be passed to the `setRelativeTransform()` method later on. These parameters are initialized to zero, expressing that the optimization starts with the initial transformation acquired by pre-registration. Also, an array for the optimized parameters is allocated with the same size. It will contain the final, optimized parameters after `Optimize()` has returned. But before, the optimization algorithm (see 3.6.2) has to be set up. Thus, the search space dimension is initialized to six and the initial values in `params` are passed. Then, callback methods are provided which enable the algorithm to create and evaluate modified parameter vectors. Please keep in mind that `getDelta()` is called once in each cycle of the main optimization loop. It describes the changes that shall be applied to each parameter within one cycle of the parameter optimization loop, thus the vector Δ that has been defined in 6.1. Furthermore, it determines the overall number of cycles in the main optimization loop by returning `false` when the desired accuracy has been reached. The `getDelta()` method is a static wrapper for the following method. Therefore, a pointer to the corresponding instance of `HxAutoRegistration` has to be passed also via the `this` pointer. This mechanism is necessary since for direct invocation of member methods via a callback mechanism, the optimization algorithm in class `CcOptimization` and its derivatives would have to know about the caller's type, which would not be a proper solution with respect to modularity. Therefore, this typecast is performed in a static class method.

```
bool HxAutoRegistration::delta( float * deltaValues )
{
    deltaCount ++;

    switch ( deltaCount ) {
    case 1:
        ...
        // set initial delta values
        deltaValues[0] = 1;
        deltaValues[1] = 1;
        deltaValues[2] = 10;
        deltaValues[3] = 4 * 0.01745;
        deltaValues[4] = 4 * 0.01745;
        deltaValues[5] = 4 * 0.01745;
        // continue optimization
        return true;
    case 2:
        ...
        // halve delta values
        deltaValues[0] = 0.5;
        deltaValues[1] = 0.5;
        deltaValues[2] = 5;
        deltaValues[3] = 2 * 0.01745;
        deltaValues[4] = 2 * 0.01745;
        deltaValues[5] = 2 * 0.01745;
        // continue optimization
        return true;
    case 3:
        ...
        // halve delta values
        deltaValues[0] = 0.25;
        deltaValues[1] = 0.25;
        deltaValues[2] = 2.5;
        deltaValues[3] = 0.01745;
        deltaValues[4] = 0.01745;
        deltaValues[5] = 0.01745;
        // continue optimization
        return true;
    case 4:
        ...
        // halve delta values
        deltaValues[0] = 0.1;
        deltaValues[1] = 0.1;
        deltaValues[2] = 1;
        deltaValues[3] = 0.01745 / 2;
        deltaValues[4] = 0.01745 / 2;
        deltaValues[5] = 0.01745 / 2;
```

```
        // continue optimization
        return true;
    default:
        ...
        // stop optimization
        return false;
    }
}
```

This implementation results in four cycles of the main optimization loop. In the first execution of the parameter optimization loop, the translational parameters will be modified by steps of $1cm$ in x-axis and y-axis direction and $10cm$ in z-axis direction, since the latter has much less impact on the image than the former ones. Rotation angles will be changed by $4°$. The values are approximately halved in each consecutive cycle of the main optimization loop. After four cycles, the optimization will be stopped.

The cost function described in the following is called by the same mechanism via a static method `evalQuality()` and gets an arbitrary parameter vector as argument.

```
float HxAutoRegistration::eval( float * params )
{
    float quality;
    ...
    // apply new transformation
    setRelativeTransform( params );
    // make a new snapshot
    drr->compute();
    ...
    // load DRR
    GREYSCALEIMAGE * pDRR = readBMPfile( "/tmp/DRR.bmp" );
    // apply sobel filter to DRR
    GRADIENTIMAGE * pDRRv = applyKernel( pDRR, tSobKernX );
    GRADIENTIMAGE * pDRRh = applyKernel( pDRR, tSobKernY );
    quality = calcGradientDifferenceMeasure( pAngioH, pAngioV, \
        pDRRh, pDRRv, \
        bBoxMinX, bBoxMinY, bBoxMaxX, bBoxMaxY, \
        varAngioH, varAngioV );
    ...
    return quality;
}
```

The given vector is passed to `setRelativeTransform()` which applies it to the 3D model in the viewer, as already explained. Then, a snapshot is made and read from disk. The Sobel kernels are applied in order to create gradient images in horizontal and vertical direction. These, together with the X-ray gradient images which have already been calculated during module setup, are passed to the quality measure, e.g. `calcGradientDifferenceMeasure()`. Its result value will be returned to the optimization algorithm.

The code demonstrated so far, resembles the default solution with gradient difference quality measure and weighted best neighbor optimization. Unimportant sections e.g. the selection of a region of interest via AMIRA's `SelectROI` module as well as debug information and options (which may be activated by defining the `DEBUG` constant) have been left out for the sake of clarity. The actual implementation e.g. allows also for selection of pattern intensity as quality measure and best neighbor optimization via the GUI of the module. For a full description, please refer there.

6.4 Reliability and Performance of Optimization

In this section, evaluations of plausibility of the described techniques shall be presented. Since the smoothness of the error surfaces in the six-dimensional parameter space of rigid registration is essential for the overall stability and performance of the system, at first some experiments with respect to the quality measures gradient difference and pattern intensity have been carried out. They will be described in 6.4.1, 6.4.2 and 6.4.3. Subsequently, registration tests have been performed in order to evaluate the performance and the stability of the complete implementation. Their results will be shown in 6.4.4. A short summary of the investigations and their consequences will be given in 6.4.5.

For all tests, data from a real patient was used instead of a phantom. Unfortunately, no ground-truth data (e.g. by means of point correspondences belonging to fiducial markers) was available. Therefore, we were not able to perform a gold standard registration for a comparison of our optimization results [30]. However, a quite good pre-registration could be achieved interactively and served as a starting point for the following evaluations.

6.4.1 Evaluation of Local Behaviour

In order to become acquainted with the behavior of the implemented quality measures, gradient difference and pattern intensity, the error surface was sampled in an environment of our pre-registration. In this first step, only the influence of translations in x and y-axis directions were considered, the other parameters were kept at their registered position. The resulting curves for both measures, evaluated with different volume rendering parameters, can be seen in appendix A.1. For sampling, a step-width of $0.1cm$ was used and ten values were sampled in each direction. The overall transparency factor α (see section 5.3) was set to 0.01 and 0.02, and two different windows, $HU \in [-250..400]$ and $HU \in [-150..400]$, were evaluated. All measurements were made with three different regions of interest (ROI), a small one covering only the ribs in figure 6.2, a medium one, defined by a square leaving out only the black borders and the complete image. The best result can be seen in figure 6.4. With both measures, the surface shows a clear minimum in the middle, which confirms our pre-registration. Surprisingly, the selection of a small ROI did not lead to an advantage, but rather decreased the smoothness. The best results have been achieved using a window $HU \in [-150..400]$ and a transparency factor $\alpha = 0.01$, although, optically, the DRR is more similar to the radiograph at $\alpha = 0.02$. The window settings had only a slight impact on pattern intensity and none at all on gradient difference. The obtained results will be used in the following experiments.

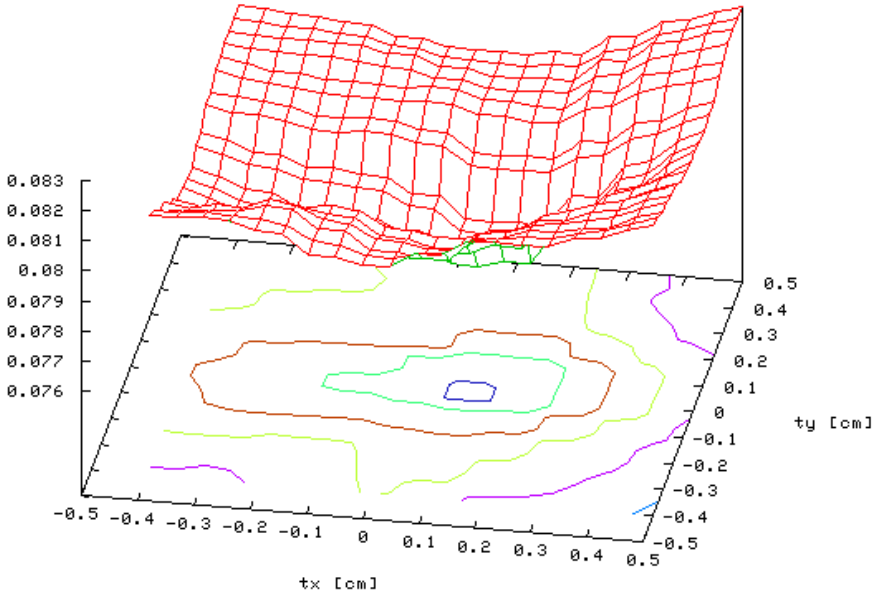


Figure 6.4: Gradient difference in local environment (translation along x and y-axis) of registration minimum, no ROI, overall transparency factor $\alpha = 0.01$, transfer function window $HU \in [-150..400]$

6.4.2 Evaluation of Local Minima

The next step was a sampling of the error curves in a broader range with respect to the pre-registration. The step-width was doubled to $0.2cm$ and the number of samples in each direction was increased to 30. All tests were performed with the ROIs defined above. The results can be seen in appendix A.2. They confirm that a small ROI covering only the ribs is not suitable, since no distinct minima are visible. With a big ROI or no ROI at all, both measures show clear minima for the registered position, though no ROI seems to be a slightly better choice. Besides the minima at the registered position, there exist others in the nearer environment, unfortunately. For example, clear minima can be seen in figure 6.5 at a distance of about $3cm$ in upper and lower y-axis direction. This coincides with the distance of ribs. It should not be a problem, since pre-registration can be achieved more accurately. More severe is the area of low error in the left of the diagram. It can be explained with concordance of the sonograph in the radiograph with the contrasted pulmonary vein in the DRR.

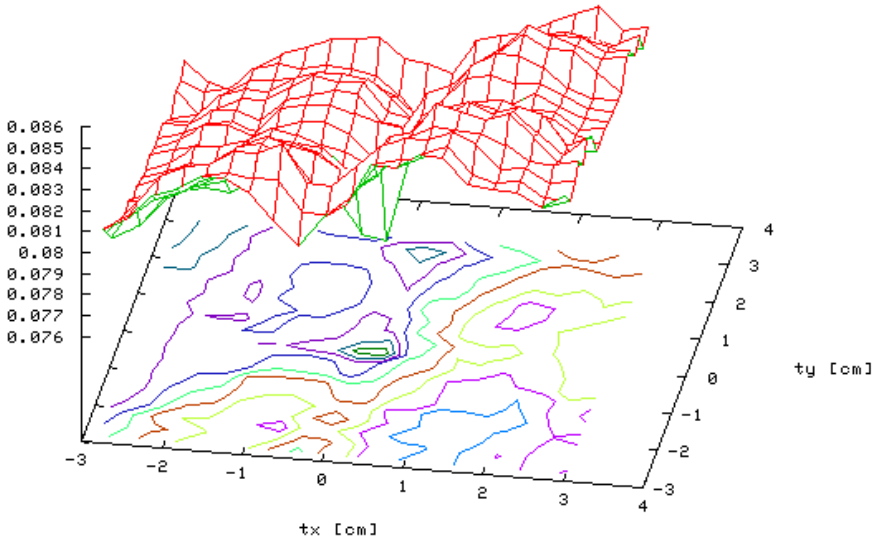


Figure 6.5: Gradient difference in environment (translation along x and z axis) of registration minimum, no ROI

6.4.3 Pattern Intensity

A small test was performed in order to evaluate the value of the weighting constant σ in pattern intensity. The different images in A.3 show that different values of σ have considerable influence on the magnitudes, but not on the monotony of the surfaces. Therefore, the suggested value $\sigma = 10$ has been adopted [58].

Now, also the other degrees of freedom, t_z, ϕ, θ and ψ were included for a test with pattern intensity only. The combination of rotations only, as well as the combination of translations only, shows good-natured error surfaces. However, the combination of a rotation and a translation shows mystifying results. As an example, figure 6.6 shows similarity for a translation along the x-axis in correspondence to a rotation about the z-axis. Only a small bowl can be seen at the expected position in the middle of the surface, whereas to the left, much lower errors were obtained. Recapitulatory, one may assert that the translations t_x and t_y are allocatable best. This does not hold for t_z (zoom), it can be determined visually very well during pre-registration, however. Change of the angles ϕ and θ did not have much impact on pattern intensity in small ranges of about 3°. Therefore, it is expected that they cannot be determined very exactly by the intensity-based optimization.

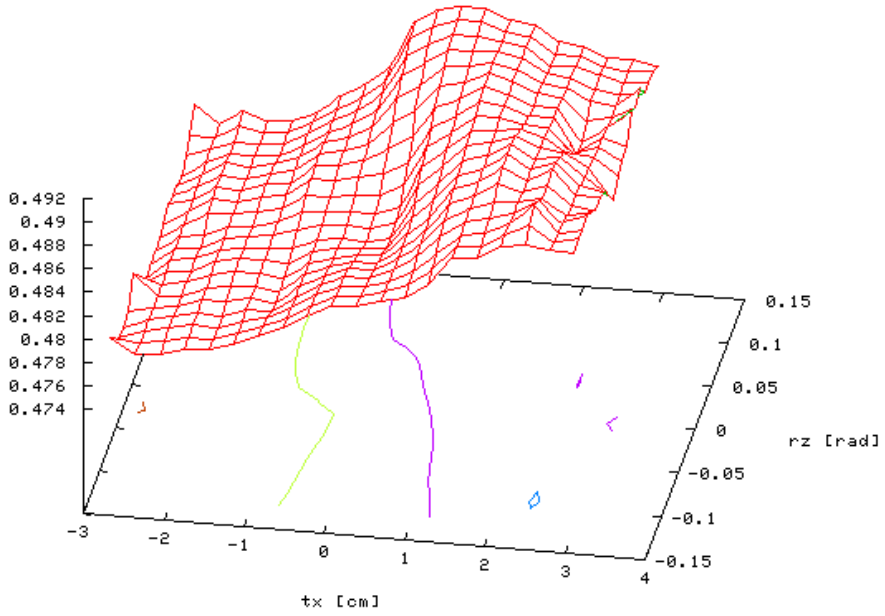


Figure 6.6: Pattern intensity in environment of registration minimum, no ROI, translation along x-axis and rotation about z-axis

6.4.4 Convergence

With the settings obtained from the experiments described in 6.4.1, 6.4.2 and 6.4.3, convergence tests were performed. The optimization algorithm was started from different positions in a search space with six degrees of freedom. For each pass, a displacement $(\Delta t_x, \Delta t_y, \Delta t_z, \Delta\phi, \Delta\theta, \Delta\psi)$ was added to the pre-registered position. Both measures, pattern intensity and gradient difference, were tested in combination with both optimization algorithms, best neighbor and weighted best neighbor. Since no ground-truth registration was available, the results could not be compared to a trusted value, but were judged by visual inspection. Table B.1 shows the results for the first experiments with an initial increment vector $\Delta_1 = (1cm, 1cm, 10cm, 2°, 2°, 2°)$. As may be expected from the availability of other minima and the bad error behavior with respect to the z-axis (see 6.4.2), optimization diverged especially for start values displaced to the left on the x-axis or considerably displaced on the z-axis. The best results have been achieved with gradient difference in combination with weighted best neighbor optimization. The other combinations diverged even more often, whereby almost always the same start values were involved. This appears to be a problem caused by the error functions and no accidental behavior.

Altogether, the results from this experiment were not satisfactory, hence, the step-widths were adjusted to $\Delta_2 = (1cm, 1cm, 10cm, 4°, 4°, 4°)$. Another test series was performed and

their results are depicted in table B.2. This time, the results were better, though stability was still not satisfactory for offsets larger than $0.5cm$ in x and y direction. Again, gradient difference in combination with weighted best neighbor optimization yielded the best results. Pattern intensity performed only slightly worse, also in combination with weighted best neighbor. The number of tests, however, is not sufficient for a final statement. Best neighbor is worse, equally for both measures. Generally, offsets higher than the corresponding initial step-width are not detected very well.

So far, registration error has not been quantified. Since no ground-truth data was available, quality could only be judged by comparing the resulting transformations for the CT model in global coordinates among each other. This was done for both experiments. The standard deviation was calculated, separately for each translation t_x, t_y and t_z, for all optimization passes, which had been qualified as "ok" by visual inspection above. With Δ_1, this yielded $\sigma_{t_x} = 0.46cm$, $\sigma_{t_y} = 0.010cm$ and $\sigma_{t_z} = 6.07cm$. With Δ_2, similar values $\sigma_{t_x} = 0.36cm$, $\sigma_{t_y} = 0.050cm$ and $\sigma_{t_z} = 7.27cm$ were obtained. As expected, the zoom value t_z is not very accurate, since it has by far less influence on the image than the other translations. Anyway, it is not needed in this application, since the position of the stent is obtained from a 2D position in the radiograph and calculated by intersecting its back-projection into 3D space with the aorta in the surface reconstruction of the CT model [25]. Very good results were obtained for t_y. This can be explained by the ribs running in horizontal direction.

6.4.5 Results

The results show that the registration approach may not be simply adapted from other research projects, where either only single femurs [58, 63] or a complete thorax [76] are involved, to the context of stent-grafting. Registration is possible only in a local area around the true point of registration. Optimization of the ratio between step-widths in different directions could still lead to an improvement, since the results with Δ_2 have been considerably better than those with Δ_1 in 6.4.4. Furthermore, the initial step-widths $\Delta = (1cm, 1cm, 10cm, 4°, 4°, 4°)$ could be reduced to some smaller values in order to increase stability. Keep in mind that a displacement of $1cm$ probably makes the algorithm consider the wrong local minima. In this case, the outgoing of iteration is very unclear. A decreased Δ, however, makes higher demands on the accuracy of an initial pre-registration.

The differences between both modalities are evident in the error surfaces, as explained in 6.4.2 and 6.4.3. This probably also holds for another quality measure, which has recently been successfully tested for registration of a complete thorax, local normalized cross-correlation [76]. Nevertheless, it should also be tested. Special attention has to be paid on data acquisition, especially concerning the influence of contrast agent in CTA. For medical accounting purposes, possibilities could be found in order to achieve similar results as in intra-operative angiography. Maybe similar results can be achieved by completely ignoring pixels with a very low intensity in the radiograph, since they are probably either caused by contrast agent or belong to an interventional instrument.

7 Outlook

Many different issues had to be considered in order to build a complex system such as CANP. First of all, the C-arm had to be calibrated. Direct volume rendering techniques were to be evaluated for the computation of DRRs. Together with C-arm calibration, this led to an efficient and accurate approach for this cost-intensive task based on 3D texture-mapping. DRRs represent the basis of our intensity-based registration approach. Now, adequate methods had to be found in order to state their similarity to a given radiograph by means of suitable similarity measures. The functionality of all those subsystems finally meets an optimization algorithm which is able to find, based on the provided images and measures, an optimal registration between the patient and the CT model. Last but not least, visualization techniques had to be found for an intuitive representation of data acquired by registration of both modalities.

In section 7.1, a summary shall be given about what has been achieved so far. Both, the overall system as well as its constituents will be covered. After that, issues shall be addressed in section 7.2 that have been beyond the means of this thesis. Thereby, ideas for improvements of the system and its future design will be considered.

7.1 Achievements

Systems are always treated on different layers of abstraction. From a general point of view, the primary goal was to build a fully functional demonstrator for intensity-based 2D-3D registration in the context of treating aortic aneurysms. Even if the overall performance and stability of the implementation still have to be increased in the future, before the system can finally be integrated into the work-flow in operating theater, it turned out that our approach is principally suitable for the problem at hand. The prototype achieves, a good pre-registration presumed, the automatic registration in one or two minutes. Thereby, the underlying image data, which has been used for testing, represented a sort of worst-case due to immense differences in intensity of contrasted vessels and also the existence of interventional instruments in the region of interest. This suggests an ample scope of improvements. Accuracy of registration has not been tested yet, since no ground-truth data for a gold standard registration has been available.

At a more detailed view, it can be stated that some of the individual subsystems mentioned in the beginning of this chapter still have to be improved. Due to a tight time schedule, trade-offs had to be found between the respective optimal solution and a feasible implementation. Camera calibration has been carried out by means of algorithms designed for CCD cameras. The obtained intrinsic parameters yield good results when applied to the calibration images itself. However, the influence of external magnetic fields, like e.g. the geomagnetic field, on the electron optics with respect to different orientations of the C-arm has not been considered at all. A calibration body could be fixed rigidly to the detector plate for this purpose

[63]. Furthermore, the fluoroscopic images show a considerable amount of spiral distortion, which is also not compensated for, at the moment.

The computation of DRRs is quite fast and of good quality, though an increase in performance is still possible by using a multi-resolution approach [58, 76]. Thereby, DRRs are created at a low resolution in the first cycle of the main optimization loop, when the step-widths of the optimization algorithm are still big. This reduces computational costs by an order of magnitude in the beginning of optimization. The image quality is then increased in each cycle, until the full quality is exploited by fine-optimization with small step-widths. Due to the inflexible setup of viewers in AMIRA, DRRs are created with a resolution of 358×358 pixels at the moment, which is too high compared to the resolution of the three-dimensional scalar field with 256×256 voxels in x and y dimensions. Speed could be increased easily be correcting this grievance.

Two quality measures, pattern intensity and gradient difference are available in order to state the degree of compliance of radiograph and DRR, based on their pixel intensities. Both seem to be principally adequate for our registration problem, despite the partly strange behavior described in section 6.4, which can be traced back to the fact of large differences between both modalities due to contrast agent and interventional instruments. However, these problems can be get under control. From the medical point of view, the requirement of more similar images could be incorporated in the future during image acquisition. For example, the registration could be done with no contrast agent at all, relying only on bony structures. Unfortunately, such image data was not to our disposal for testing. Furthermore, image pre-processing could be performed in order to exclude pixels belonging to some kind of instrument. Since they mostly consist of metal, they produce quite dark pixels in the radiograph, whereas even such dense materials as bone result in higher intensities. Therefore, such pixels can be detected quite easily. Techniques for detection and treatment of such pixels are described by Roth [63]. Another quality measure called local normalized cross correlation has recently been suggested in the context of 2D-3D registration of radiographs [76]. This is not implemented yet.

For optimization, two quite similar algorithms, best neighbor and weighted best neighbor, have been implemented. In each iteration, they rely on the evaluation of the error surface at points which are a certain distance away from the current point in search space. The results in 6.4.4 show that the choice of those step-withs, not only their magnitude but also their ratio among each other, is crucial for stability. This essential point could not be investigated thoroughly and improvements can probably be achieved by further evaluation. Very large steps can only be taken safely, if the error surface is smooth enough. This fact will finally decide over the feasibility of our approach, because pre-registration has to be already quite accurate, if the optimum shall be found using only small steps. Maybe, smoothness can be improved by a multi-resolution approach as described above [58]. With our demo data, it took quite a long time to find an appropriate pre-registration which made the algorithm converge and this task has to be done under critical time constraints in the operating theater. However, we did not know a very crucial setting, the inclination of the C-arm. It had to be determined by cumbersome trials for our data. It actually can be read off directly at the C-arm and is therefore assumed to be known. Assuming further a horizontal alignment of the patient, only four degrees of freedom are left, which are much easier to determine for an experienced system user. Depending on the outcome of the efforts for improvement of similarity between both image modalities (verbose camera calibration and reduction of

qualitative differences in the images), the step-widths have to be evaluated anew. Thereby, it should also be considered to restrict registration to a subset of the six degrees of freedom. As already mentioned, the inclination of the C-arm, as well as the horizontal alignment of the patient are already known. Thus performance and probably also stability of optimization can be increased.

Radiologists currently use a SIEMENS SENSATION CARDIAC proprietary system for segmentation and visualization of CT data. An extensive segmentation software including predefined windows and other verbose tools like region-growing etc. is included in this system. However, it is scarcely used, since standard interfaces like the "DICOM" file format are disregarded for three-dimensional models. The only possibility for surgeons to obtain a three-dimensional impression of the patient, is by an "AVI" film sequence which shows the model from different points of view. No interactive navigation is possible and thus, the benefit is very limited. In the surgeon's everyday life, actually only the two-dimensional axial, sagittal or frontal slices are extensively used. Therefore, different techniques for visualization of three-dimensional models have been evaluated. The indirect rendering approach, using a surface model of the patient obtained via segmentation, works satisfactory and is ready to replace the actual state. Direct rendering techniques based on surface representation with shading could contribute to a more accurate representation in the future, without the need of a time-consuming explicit segmentation step. This, however, would make by far greater demands on the underlying hardware platform than conventional surface shading. Suitable multidimensional transfer functions for visualization of the human body have not been evaluated so far.

7.2 Future Work

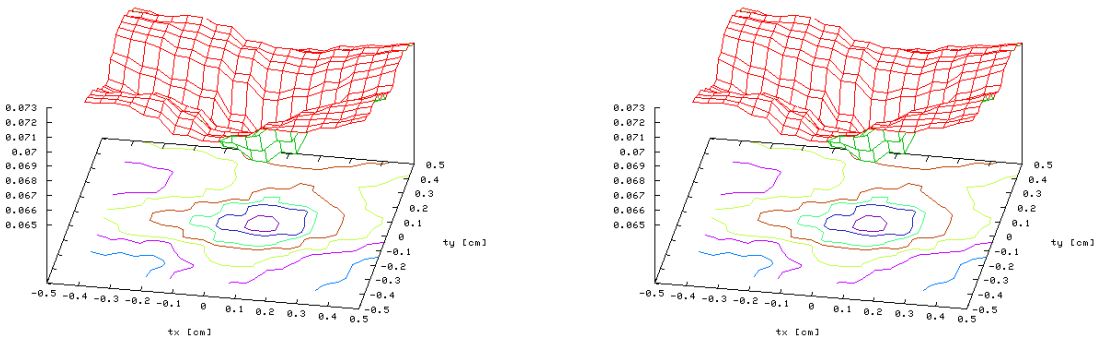
In the last section, already a lot of starting points for improving distinct components towards more overall system stability and performance have been described. They shall be briefly reviewed here. A better camera calibration including influence of magnetic fields and image distortion is desirable. Furthermore, methods have to be evaluated in order to cope with differences in both image modalities, either by changing image acquisition or by exclusion of affected intensity values. A multi-resolution approach can help to smooth the error surface at points quite afar from the optimum. Additionally this would increase performance of DRR computation. Computation speed may also be increased by reducing the resolution of the DRR or by computation of DRR and evaluation of quality measure in parallel, which is not implemented yet.

Concerning the overall system, two issues are of major interest. At first, it has to be evaluated, if an accurate pre-registration can be obtained by the means provided by CANP, in a time acceptable in operating theater. Some few metallic markers could be placed on the patient's ribs and thereby help to make their alignment easier and safer. Alternatively, the pre-registration may also be performed by tracking of C-arm and patient [71]. This, however, would foil our elegant approach which gets by without any hardware except a fast PC, especially no expensive and error-prone tracking hardware is needed. The investigation of pre-registration under realistic conditions should therefore be the next step, since it is crucial for the question, if the current design is actually reasonable.

A second major issue is the proof of accuracy and correctness of the complete system by means of a phantom thorax and fiducial markers. Such a gold standard registration is necessary in order to have trusted values for the comparison of results. As an alternative, a different and less complex setup could be used, a phantom thorax containing a replica of the aorta. CT and X-Ray images could be taken with a catheter at the same, fix position. In this way, the registration procedure could be tested by directly comparing the real location of the stent in the CT voxel data to the location predicted by CANP.

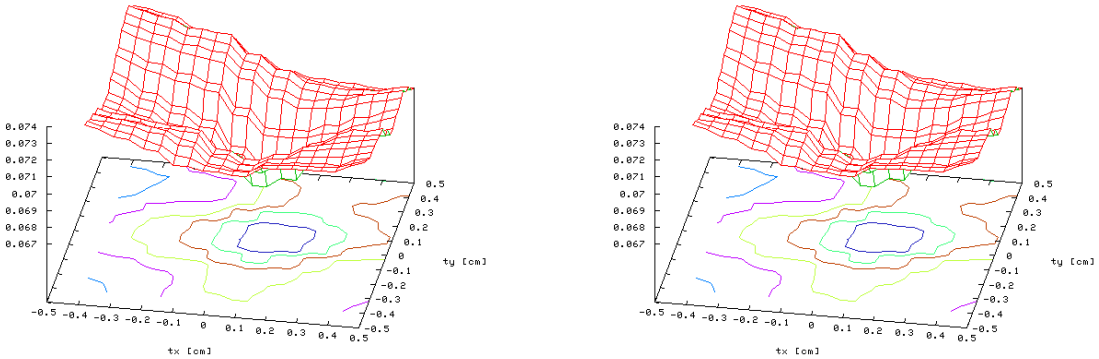
A Analysis of Quality Measures

A.1 Evaluation of Local Behavior



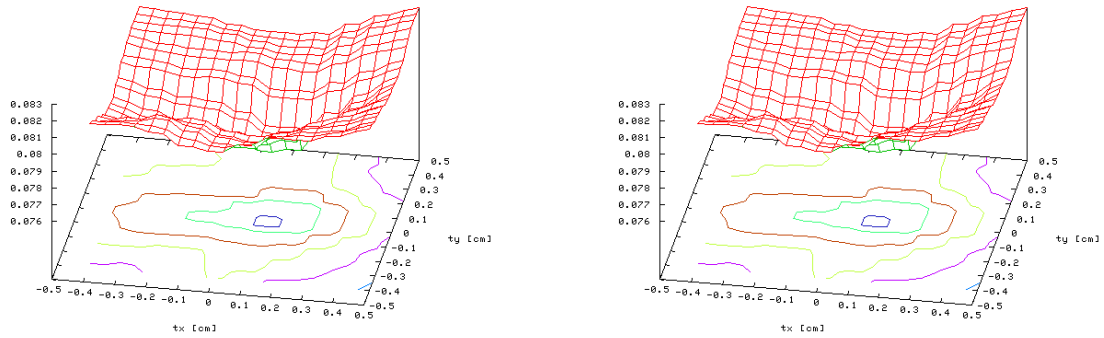
(a) (b)

Figure A.1: Gradient difference in local environment (translation along x and y-axis) of registration minimum, large ROI, overall transparency factor $\alpha = 0.01$, (a) transfer function window $HU \in [-150..400]$ (b) transfer function window $HU \in [-250..400]$



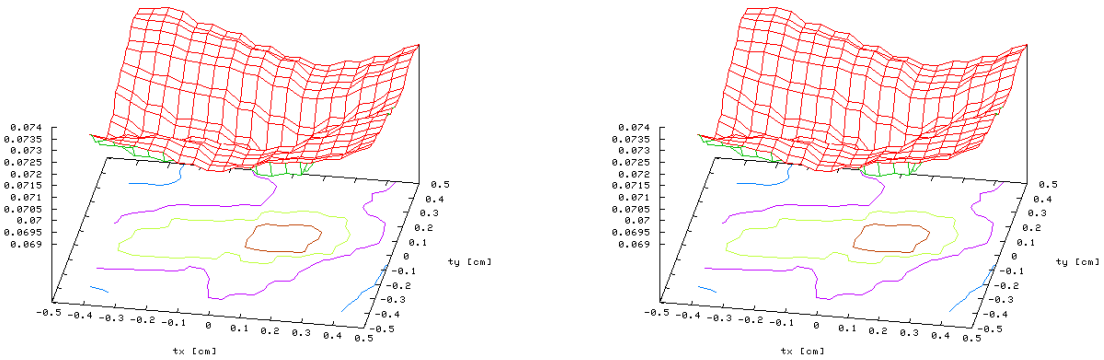
(a) (b)

Figure A.2: Gradient difference in local environment (translation along x and y-axis) of registration minimum, large ROI, overall transparency factor $\alpha = 0.02$, (a) transfer function window $HU \in [-150..400]$ (b) transfer function window $HU \in [-250..400]$



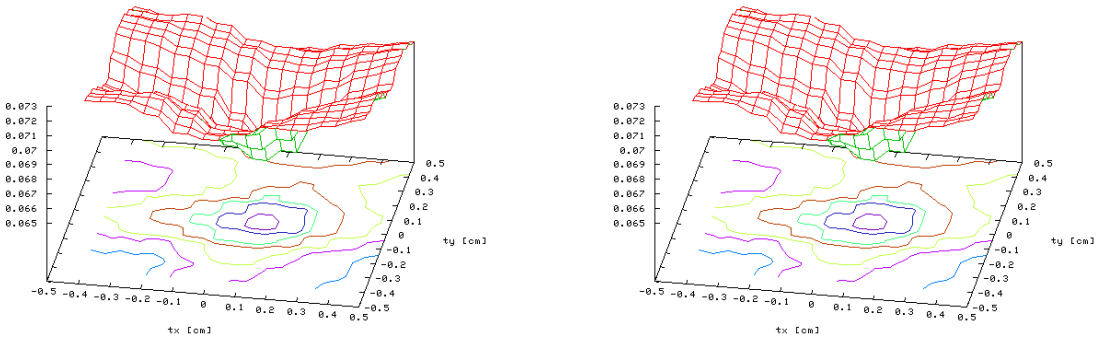
(a) (b)

Figure A.3: Gradient difference in local environment (translation along x and y-axis) of registration minimum, no ROI, overall transparency factor $\alpha = 0.01$, (a) transfer function window $HU \in [-150..400]$ (b) transfer function window $HU \in [-250..400]$



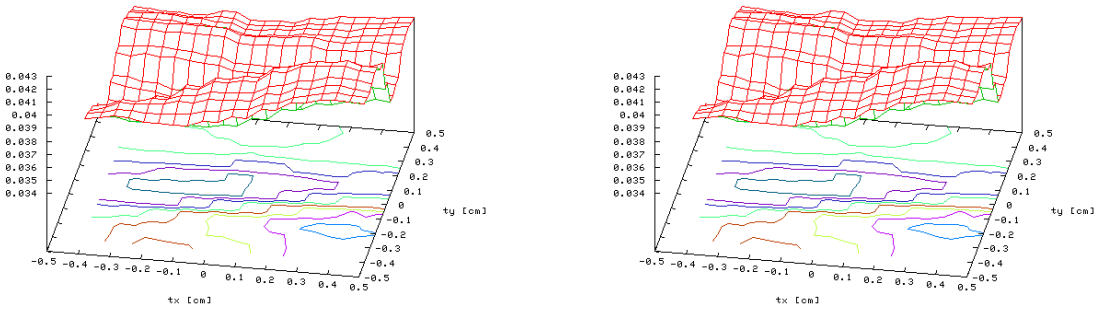
(a) (b)

Figure A.4: Gradient difference in local environment (translation along x and y-axis) of registration minimum, no ROI, overall transparency factor $\alpha = 0.02$, (a) transfer function window $HU \in [-150..400]$ (b) transfer function window $HU \in [-250..400]$



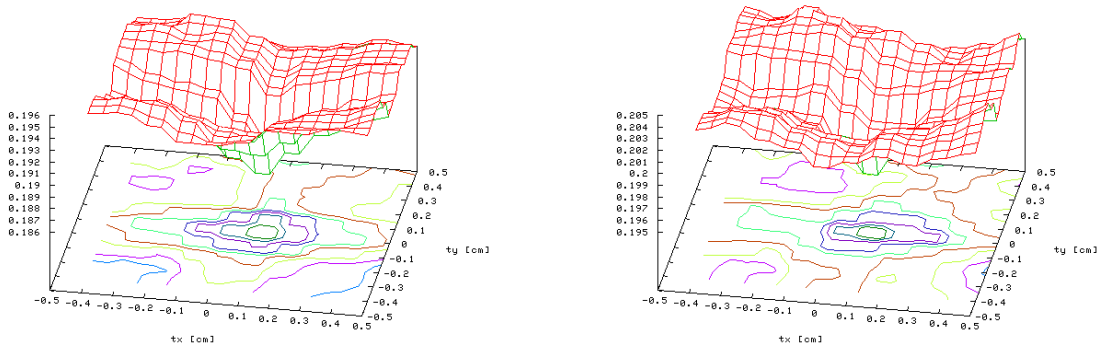
(a) (b)

Figure A.5: Gradient difference in local environment (translation along x and y-axis) of registration minimum, small ROI, overall transparency factor $\alpha = 0.01$, (a) transfer function window $HU \in [-150..400]$ (b) transfer function window $HU \in [-250..400]$



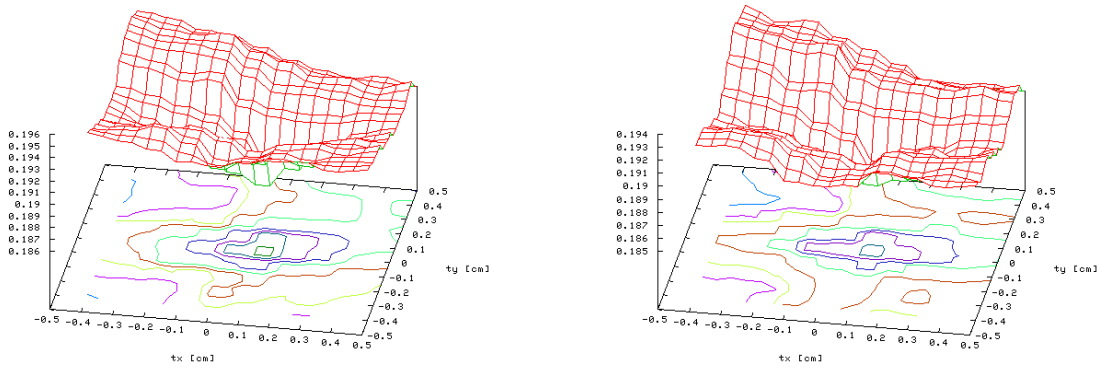
(a) (b)

Figure A.6: Gradient difference in local environment (translation along x and y-axis) of registration minimum, small ROI, overall transparency factor $\alpha = 0.02$, (a) transfer function window $HU \in [-150..400]$ (b) transfer function window $HU \in [-250..400]$



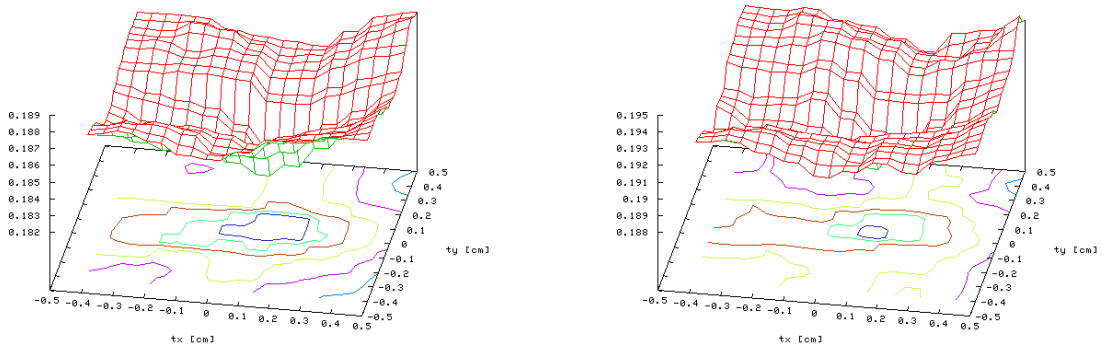
(a) (b)

Figure A.7: Pattern intensity in local environment (translation along x and y-axis) of registration minimum, large ROI, overall transparency factor $\alpha = 0.01$, (a) transfer function window $HU \in [-150..400]$ (b) transfer function window $HU \in [-250..400]$



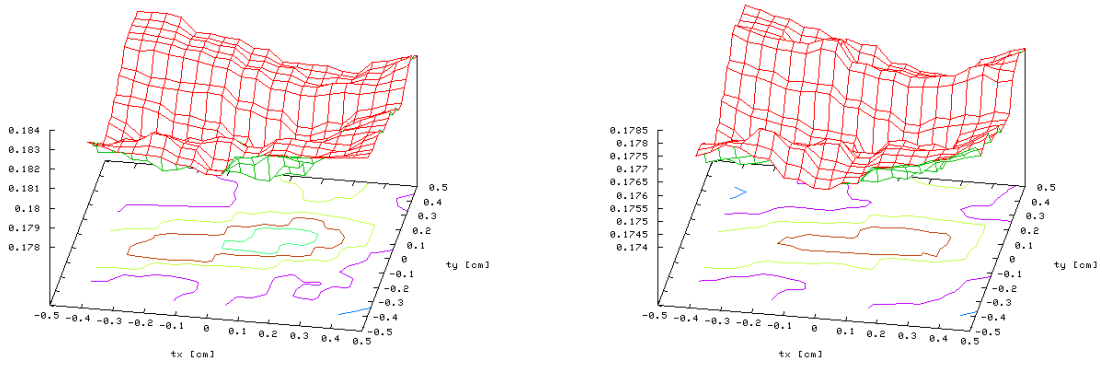
(a) (b)

Figure A.8: Pattern intensity in local environment (translation along x and y-axis) of registration minimum, large ROI, overall transparency factor $\alpha = 0.02$, (a) transfer function window $HU \in [-150..400]$ (b) transfer function window $HU \in [-250..400]$



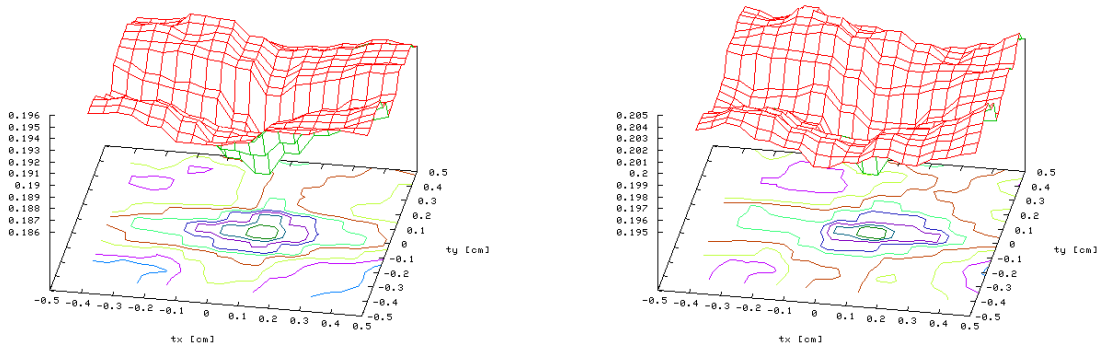
(a) (b)

Figure A.9: Pattern intensity in local environment (translation along x and y-axis) of registration minimum, no ROI, overall transparency factor $\alpha = 0.01$, (a) transfer function window $HU \in [-150..400]$ (b) transfer function window $HU \in [-250..400]$



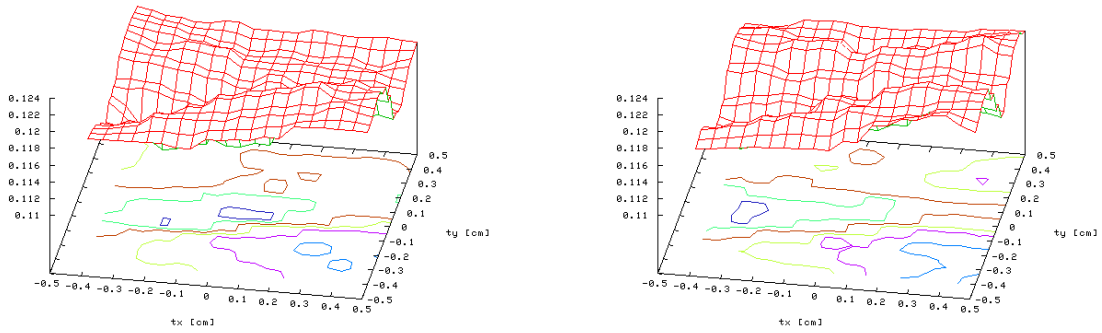
(a) (b)

Figure A.10: Pattern intensity in local environment (translation along x and y-axis) of registration minimum, no ROI, overall transparency factor $\alpha = 0.02$, (a) transfer function window $HU \in [-150..400]$ (b) transfer function window $HU \in [-250..400]$



(a) (b)

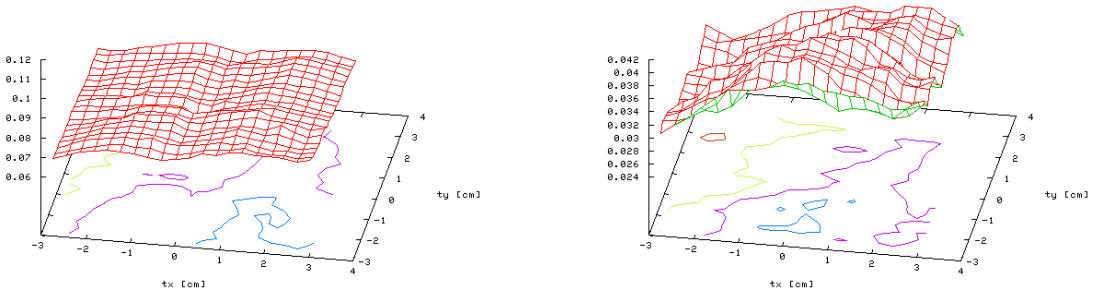
Figure A.11: Pattern intensity in local environment (translation along x and y-axis) of registration minimum, small ROI, overall transparency factor $\alpha = 0.01$, (a) transfer function window $HU \in [-150..400]$ (b) transfer function window $HU \in [-250..400]$



(a) (b)

Figure A.12: Pattern intensity in local environment (translation along x and y-axis) of registration minimum, small ROI, overall transparency factor $\alpha = 0.02$, (a) transfer function window $HU \in [-150..400]$ (b) transfer function window $HU \in [-250..400]$

A.2 Evaluation of Local Minima



(a) (b)

Figure A.13: Gradient difference in environment (translation along x and y-axis) of registration minimum, (a) big ROI (b) small ROI

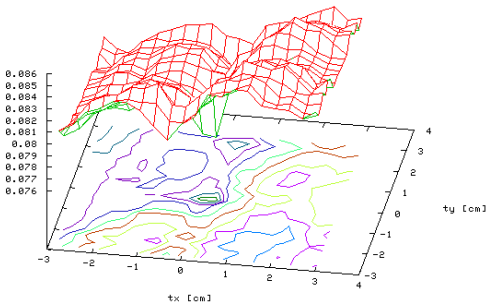
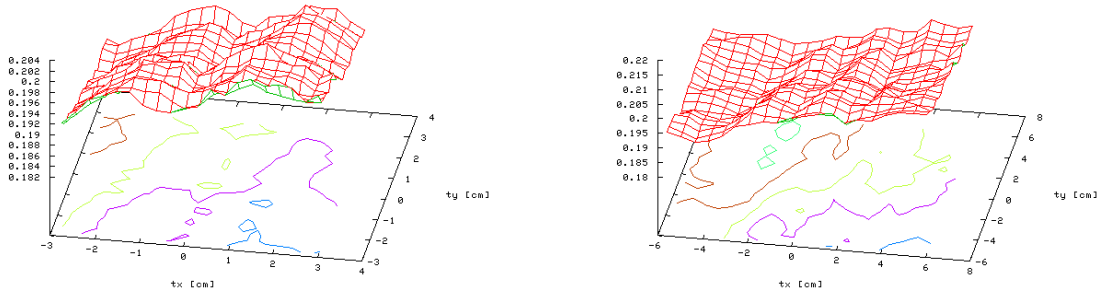
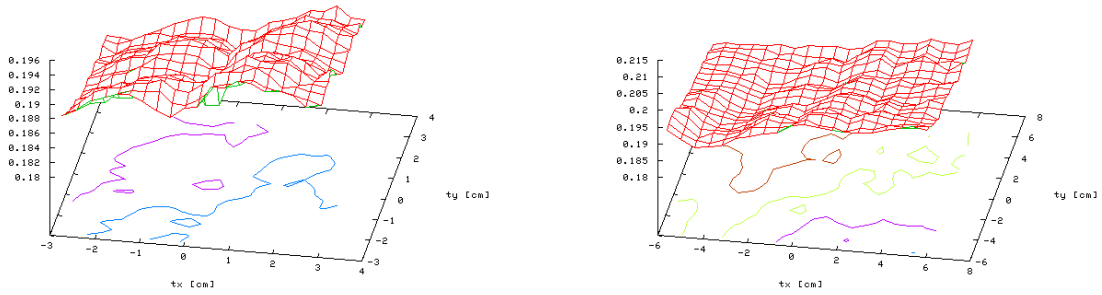


Figure A.14: Gradient difference in environment (translation along x and z axis) of registration minimum, no ROI



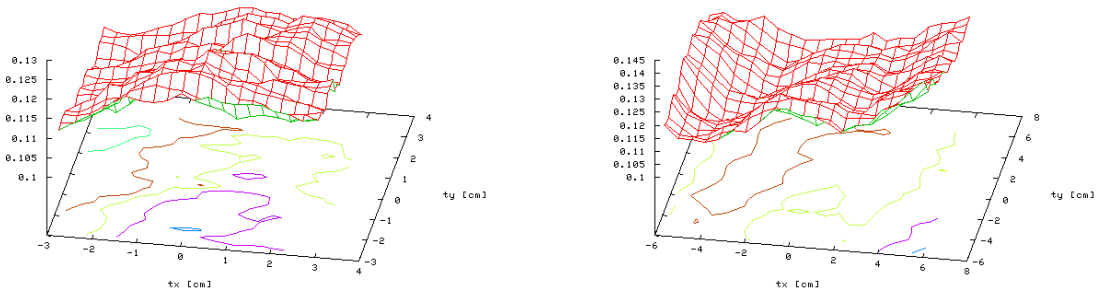
(a) (b)

Figure A.15: Pattern intensity in environment (translation along x and y-axis) of registration minimum, big ROI



(a) (b)

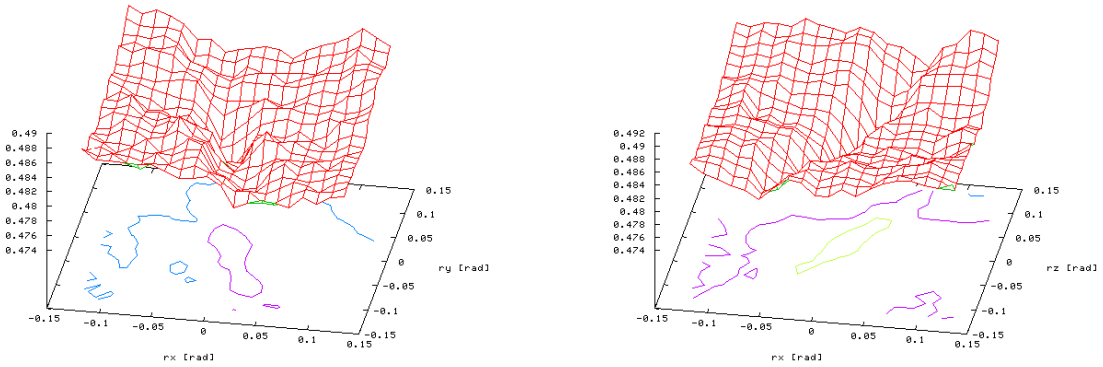
Figure A.16: Pattern intensity in environment (translation along x and y-axis) of registration minimum, no ROI



(a) (b)

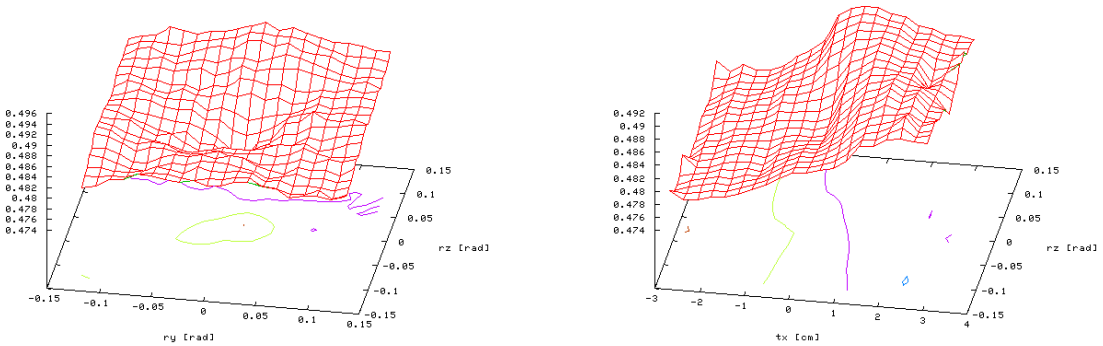
Figure A.17: Pattern intensity in environment (translation along x and y-axis) of registration minimum, small ROI

A.3 Pattern Intensity



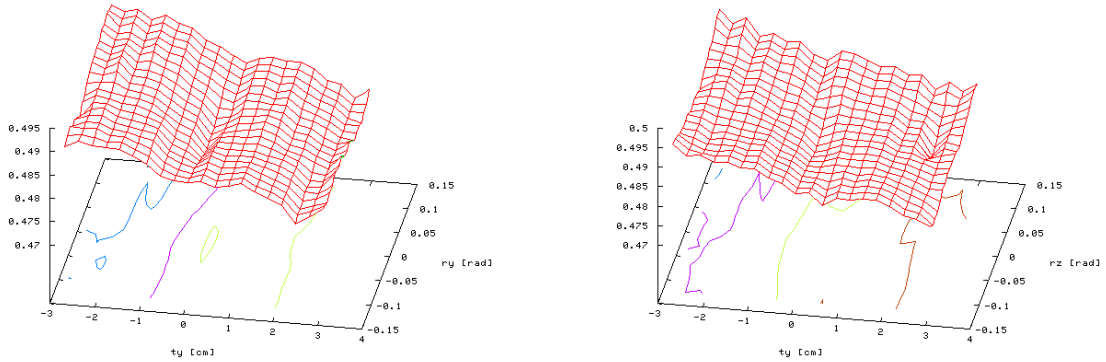
(a) (b)

Figure A.18: Pattern intensity in environment of registration minimum, no ROI, (a) rotations about x and y-axis, (b) rotations about x and z-axis



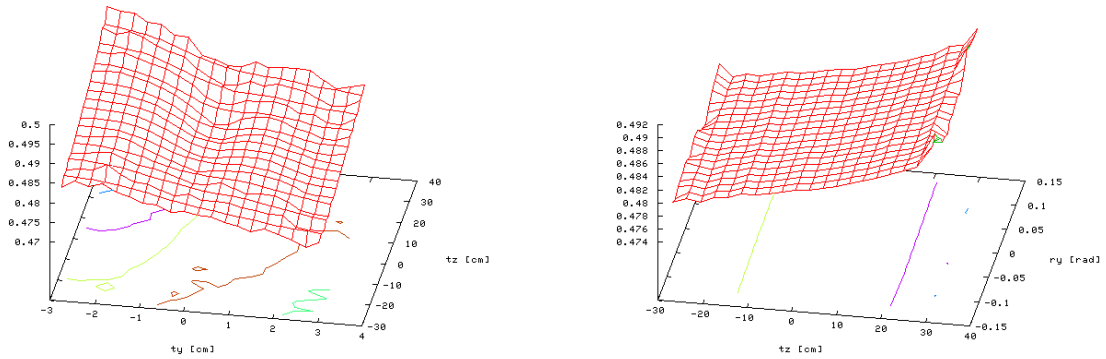
(a) (b)

Figure A.19: Pattern intensity in environment of registration minimum, no ROI, (a) rotations about y and z-axis, (b) translation along x-axis and rotation about z-axis



(a) (b)

Figure A.20: Pattern intensity in environment of registration minimum, no ROI, (a) translation along y-axis and rotation about y-axis, (b) translation along y-axis and rotation about z-axis



(a) (b)

Figure A.21: Pattern intensity in environment of registration minimum, no ROI, (a) translations along y and z-axis, (b) translation along z-axis and rotation about y-axis

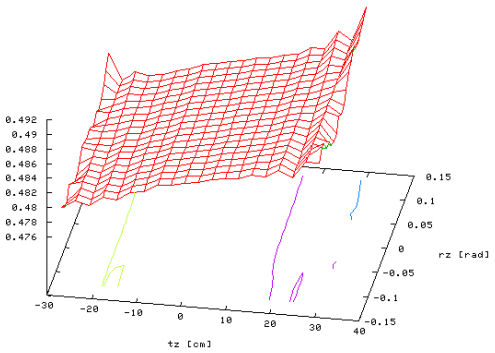
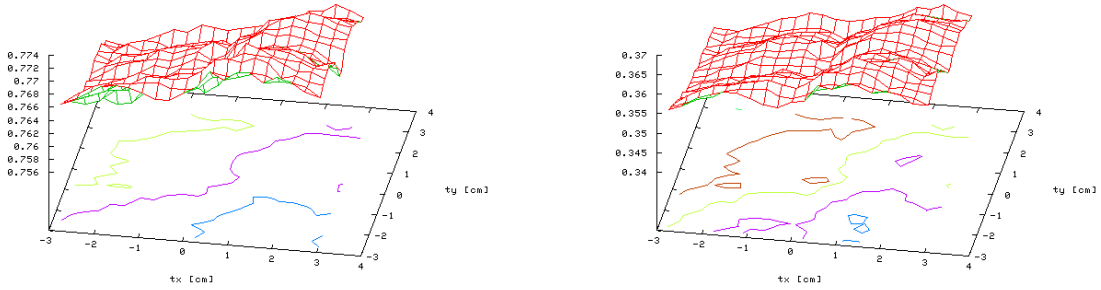
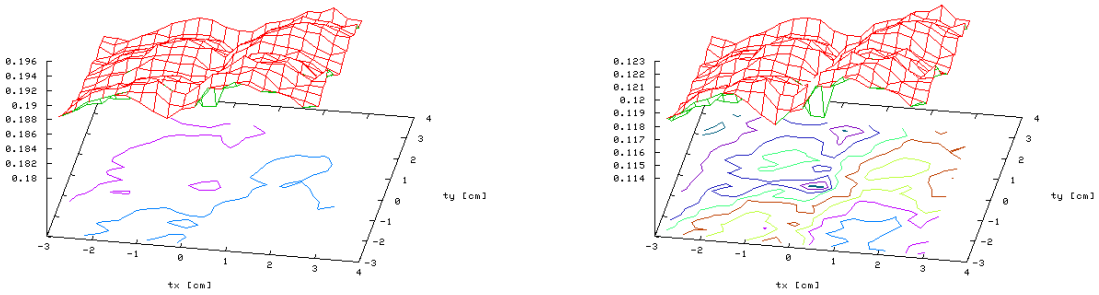


Figure A.22: Pattern intensity in environment of registration minimum, no ROI, translation along z-axis and rotation about z-axis



(a) (b)

Figure A.23: Pattern intensity in environment of registration minimum, no ROI, overall transparency factor $\alpha = 0.01$, (a) $\sigma = 1$ (b) $\sigma = 5$



(a) (b)

Figure A.24: Pattern intensity in environment of registration minimum, no ROI, overall transparency factor $\alpha = 0.01$, (a) $\sigma = 10$ (b) $\sigma = 15$

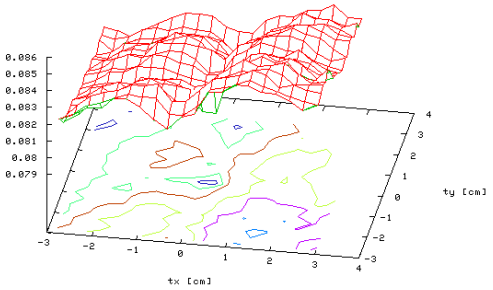


Figure A.25: Pattern intensity in environment of registration minimum, no ROI, overall transparency factor $\alpha = 0.01$, $\sigma = 20$

B Analysis of Optimization

Offset in each dof						Result			
Δt_x [cm]	Δt_y [cm]	Δt_z [cm]	$\Delta\phi$ [deg]	$\Delta\theta$ [deg]	$\Delta\psi$ [deg]	PI/BN	PI/WBN	GD/BN	GD/WBN
1.0	1.0	0	0	0	0	3/2/3/2 ok	7/1/1/1 bad	3/2/5/2 ok	5/2/9/4 bad
1.0	1.0	0	0	0	5	-	-	9/4/2/3 bad	-
0.8	0.8	0	2	2	3	5/2/1/3 bad	-	6/3/2/3 bad	-
-0.7	-0.5	0	0	0	4	-	-	-	-
-0.7	-0.5	0	0	0	2	-	-	-	6/2/4/8 ok
0.7	-0.5	0	0	0	2	4/2/4/4 ok	4/3/1/2 ok	4/2/2/3 ok	5/1/3/2 ok
0.7	-0.5	20	0	0	3	-	6/1/1/5 ok	-	4/5/5/5 ok
-0.5	0.8	-20	0	0	3	-	-	-	-
1.0	0.8	-20	0	0	3	-	-	-	-
0.3	0.9	-20	2	2	4	-	-	-	-
0.4	0.7	-10	1	0.5	3	-	-	-	-
0.2	0.4	-5	0.5	1	2	5/1/2/1 ok	-	4/4/2/2 ok	5/3/2/12 ok

Table B.1: Convergence tests for pattern intensity (PI) and gradient difference (GD) in combination with best neighbor (BN) and weighted best neighbor (WBN) optimization. The initial refinement vector $\Delta = (1cm, 1cm, 10cm, 2°, 2°, 2°)$ has been used. The number of cycles in each of the four cycles of the main optimization loop has been counted for each experiment. By visual comparison between resulting DRR and radiograph, the quality has been judged. A dash signals divergence. In each pass, optimization is started at the pre-registered point in search space, displaced by $(\Delta t_x, \Delta t_y, \Delta t_z, \Delta\phi, \Delta\theta, \Delta\psi)$

Offset in each dof						Result			
Δt_x [cm]	Δt_y [cm]	Δt_z [cm]	$\Delta\phi$ [deg]	$\Delta\theta$ [deg]	$\Delta\psi$ [deg]	PI/BN	PI/WBN	GD/BN	GD/WBN
0.8	0.8	0	2	2	3	-	-	8/1/8/2 bad	6/1/2/2 ok
-0.6	0.3	0	0.5	-1	-2	3/2/2/2 ok	2/2/2/2 ok	2/2/4/1 ok	2/2/2/4 ok
0.4	-0.7	0	-0.8	-0.3	2	6/2/1/3 bad	3/1/3/1 bad	-	3/3/2/6 ok
0.9	0.4	5	1.3	-0.7	2.3	4/4/2/3 ok	3/1/3/4 ok	4/2/3/6 ok	3/1/2/4 ok
-0.9	0.8	-5	-1.4	0.3	-2.2	-	-	-	-
0.9	0.8	-5	-1.4	0.3	-2.2	4/3/1/3 bad	6/2/1/1 bad	4/4/2/2 bad	-
0.2	-0.3	-10	-0.7	0.9	2.1	2/3/2/4 ok	3/1/2/1 ok	7/1/6/2 bad	3/1/3/3 ok
-0.3	0.5	15	0	0	2.8	4/1/4/4 ok	5/2/1/1 ok	4/2/3/4 ok	5/5/3/7 ok
0.4	-0.3	-20	0.5	0.5	3	-	3/1/2/4 bad	-	-
0.4	-0.3	-15	0.5	0.5	3	4/1/2/1	4/2/6/1 bad	4/1/1/1 bad	-
0.4	-0.3	-10	0.5	0.5	3	-	-	-	-

Table B.2: Convergence tests for pattern intensity (PI) and gradient difference (GD) in combination with best neighbor (BN) and weighted best neighbor (WBN) optimization. The initial refinement vector $\Delta = (1cm, 1cm, 10cm, 4°, 4°, 4°)$ has been used. The number of cycles in each of the four cycles of the main optimization loop has been counted for each experiment. By visual comparison between resulting DRR and radiograph, the quality has been judged. A dash signals divergence. In each pass, optimization is started at the pre-registered point in search space, displaced by $(\Delta t_x, \Delta t_y, \Delta t_z, \Delta\phi, \Delta\theta, \Delta\psi)$

C Glossary

AAA *see* ABDOMINAL AORTIC ANEURYSM

Abdominal Aortic Aneurysm Leak in the inner wall of the aorta which results in a blood-streamed lump located in the abdomen

AR *see* AUGMENTED REALITY

Augmented Reality. A technique that uses virtual objects to enhance the user's perception of the real world.

CANP Computer Aided Navigation and Planning

CCD Charged Coupled Device

CT *see* COMPUTED TOMOGRAPHY

Computed Tomography Process of creating cross-sectional or tomographic images from projections (line integrals) of the patient at multiple angles using X-rays. The 3D model is reconstructed by iterative algorithms.

CTA *see* COMPUTER TOMOGRAPHY ANGIOGRAPHY

Computer Tomography Angiography Computer Tomography in combination with contrast agents in order to visualize blood vessel lumen

DLT Direct Linear Transformation

DICOM Digital Imaging and Communications in Medicine

DRR *see* DIGITALLY RECONSTRUCTED RADIOGRAPH

Digitally Reconstructed Radiograph An artificial X-ray image computed from CT data via direct volume rendering and a perspective or orthographic projection

GPU Graphical Processing Unit

GUI Graphical User Interface

TAA *see* THORACIC AORTIC ANEURYSM

TAAA *see* THORACOABDOMINAL AORTIC ANEURYSM

Thoracic Aortic Aneurysm Leak in the inner wall of the aorta which results in a blood-streamed lump located in the thorax

Thoracoabdominal Aortic Aneurysm Leak in the inner wall of the aorta which results in a blood-streamed lump located in both, abdomen and thorax

Type-B-Dissection Two leaks in the inner aortic wall creating two distinguishable and seperated lumen

ROI Region of Interest

dof degrees of freedom

SVD Singular Value Decomposition

DSA *see* DIGITAL SUBTRACTION ANGIOGRAPHY

Angiography X-ray based technique, where also blood vessel lumen is visualized by usage of contrast agents

Digital Subtraction Angiography Techniques which calculates a difference image from a contrasted and an uncontrasted radiograph in order to visualize only the blood vessel lumen. See also ANGIOGRAPHY

Bibliography

[1] *Medcyclopaedia*. `http://www.amershamhealth.com/medcyclopaedia/index.asp`, 2003.

[2] Y. I. ABDEL-AZIZ and H. M. KARARA, *Direct Linear Transformation from Comparator Coordinates into Object Space Coordinates in Close-Range Photogrammetry*, Symposium on Close Range Photogrammetry, (1971), pp. 1–18.

[3] M. A. AUDETTE, F. P. FERRIE, and T. M. PETERS, *An Algorithmic Overview of Surface Registration Techniques for Medical Imaging*, Medical Image Analysis, 4 (1999), pp. 201–217.

[4] R. BELL, P. R. TAYLOR, M. AUKETT, T. SABHARWAL, and J. REIDY, *Results of Urgent and Emergency Thoracic Procedures treated by Endoluminal Repair*, European Journal of Endovascular Surgery, 25 (2003), pp. 527–531.

[5] P. BERGERON, T. DE CHAUMARAY, J. GAY, and V. DOUILLEZ, *Endovascular treatment of thoracic aortic aneurysms*, Journal of Cardiovascular Surgery, 44 (2003), pp. 349–361.

[6] J. BLANKENSTEIJN, *Imaging Techniques for Endovascular Repair of Abdominal Aortic Aneurysms*, Medica Mundi, 44 (2002).

[7] I. BRONSTEIN, K. SEMENDJAJEW, G. MUSIOL, and H. MÜHLIG, *Taschenbuch der Mathematik*, Verlag Harri Deutsch, 1999. 4., überarbeitete und erweiterte Auflage der Neubearbeitung.

[8] L. G. BROWN and T. BOULT, *Registration of Planar Film Radiographs with Computed Tomography*, IEEE Workshop on Mathematical Methods in Biomedical Imaging, (1996).

[9] B. CABRAL, N. CAM, and J. FORAN, *Accelerated Volume Rendering and Tomographic Reconstruction Using Texture Mapping Hardware*, in ACM Symposium on Volume Visualization, 1994.

[10] F. CRIADO, N. CLARK, and M. BARNATAN, *Stent graft repair in the aortic arch and descending thoracic aorta: a 4-year experience*, Journal of Vascular Surgery, 36 (2002), pp. 1121–1128.

[11] B. DREBIN, L. CARPENTER, and P. HANRAHAN, *Volume Rendering*, in Computer Graphics Conference Series, SIGGRAPH, 1988, pp. 65–74.

[12] K. ENGEL, *OpenQVis Volume Rendering Library*. `http://openqvis.sourceforge.net`.

[13] K. ENGEL, *Strategien und Algorithmen zur Interaktiven Volumenvisualisierung in Digitalen Dokumenten*, PhD thesis, Universität Stuttgart, 2002.

[14] K. ENGEL and T. ERTL, *Interactive High-Quality Volume Rendering with Flexible Consumer Graphics Hardware*, EUROGRAPHICS, (2002).

[15] K. ENGEL, M. KRAUS, and T. ERTL, *High-Quality Pre-Integrated Volume Rendering Using Hardware-Accelerated Pixel Shading*, in Siggraph/Eurographics Workshop on Graphics Hardware, 2001.

[16] O. ETZMUSS, *Indeed Visual Concepts*, Personal Communications. October 2003.

[17] O. FAUGERAS, *Three-Dimensional Computer Vision: A geometric Viewpoint*, MIT Press, Cambridge, MA, 1995.

[18] W. FERGUSON, *The Photographic Researches of Ferdinand Hurter and Vero C. Driffield*, Dobbs Ferry, NY Morgan and Morgan, 1974.

[19] M. FEUERSTEIN, *Design of a Planning Tool for Augmented Reality Supported Placement and Tracking for a Telemanipulator in Heart Surgery*, Master's thesis, Technical University of Munich, December 2003.

[20] J. FOLEY, A. VAN DAM, S. FEINER, and J. HUGHES, *Computer Graphics - Principles and Practice*, Addison-Wesley Publishing Company, 1990.

[21] R. GÖCKE, *Schnelle Verfahren des Volume Rendering für die 2D/3D-Registrierung*, Master's thesis, Universität Rostock, 1997.

[22] R. GÖCKE, J. WEESE, and H. SCHUMANN, *Fast Volume Rendering Methods for Voxel-based 2D/3D Registration - A Comparative Study*. `http://citeseer.nj.nec.com/233235.html`, 1999.

[23] J. GOLZARIAN, *Imaging after endovascular repair of abdominal aortic aneurysms*, Abdominal Imaging, 28 (2002).

[24] M. GRABENWOGER, T. FLECK, M. CZERNY, D. H. M. EHRLICH, M. SCHODER, J. LAMMER, and E. WOLNER, *Endovascular stent graft placement in patients with acute thoracic aortic syndromes*, European Journal of Cardiothorac Surgery, 23 (2003), pp. 788–793.

[25] M. GROHER, *Development of a Planning and Navigation Tool for Endoscopic Treatment of Aortic Aneurysms - Computer Supported Implantation of a Stent Graft*, December 2003.

[26] A. GUÉZIEC, P. KAZANZIDES, B. WILLIAMSON, and R. TAYLOR, *Anatomy-Based Registration of CT-Scan and Intraoperative X-Ray Images for Guiding a Surgical Robot*, in IEEE, 1998.

[27] S. HALL, *Endovascular repair of abdominal aortic aneurysms*, AORN Journal, 77 (2003), pp. 631–642.

[28] K. HAMMER, *Physikalische Formeln und Tabellen*, J. Lindauer Verlag (Schaefer), 1988, ch. A 5.

[29] C. HAMPTON, T. PERSONS, C. WYATT, and Y. ZHANG, *Survey of Image Segmentation*. `http://www.citeseer.nj.nec.com/hampton98survey.html`.

[30] R. HARTLEY and A. ZISSERMAN, *Multiple View Geometry in computer vision*, Cambridge University Press, 2000.

[31] H. IMAMURA, N. IDA, N. SUGIMOTO, S. EIHO, S. ICHI URAYAMA, K. UENO, and K. INOUE, *Registration of Preoperative CTA and Intraoperative Fluoroscopic images for Assisting Aortic Stent Grafting*, in MICCAI unknown, sep 2002, pp. 477–484.

[32] L. JOLLIFFE, *Principal Component Analysis*, Springer Verlag, New York, 1986.

[33] S. JONIĆ, P. THÉVENAZ, and M. UNSER, *Multiresolution-Based Registration of a Volume to a Set of its Projections*, in Proceedings of the SPIE International Symposium on Medical Imaging: Image Processing (MI'03), San Diego CA, USA, February 2003, pp. 1049–1052.

[34] J. KNISS, G. KINDLMANN, and C. HANSEN, *Interactive Volume Rendering Using Multi-Dimensional Transfer Functions and Direct Manipulation Widgets*, Visualization, (2001).

[35] J. KNISS, G. KINDLMANN, and C. HANSEN, *Multi-Dimensional Transfer Functions for Interactive Volume Rendering*, IEEE Transactions on Visualization and Computer Graphics, (2002).

[36] J. KNISS, P. MCCORMICK, J. AHRENS, J. PAINTER, A. KEAHEY, and C. HANSEN, *Interactive Texture-Based Volume Rendering for Large Data Sets*, IEEE Computer Graphics and Applications, 21 (2001).

[37] J. KNISS, S. PREMOZE, C. HANSEN, and D. EBERT, *Interactive Translucent Volume Rendering and Procedural Modeling*, IEEE Visualization, (2002).

[38] J. KNISS, S. PREMOZE, M. IKITS, A. LEFOHN, C. HANSEN, and E. PRAUN, *Gaussian Transfer Functions for Multi-Field Volume Visualization*, IEEE Visualization, (2003).

[39] KONRAD-ZUSE-ZENTRUM FÜR INFORMATIONSTECHNIK BERLIN (ZIB) and INDEED - VISUAL CONCEPTS GMBH, *amira 3.0 - User's Guide and Reference Manual*, TGS Template Graphics Software, Inc., Berlin, Germany, and USA, 2002.

[40] W. KRÜGER, *The Application of Transport Theory to the Visualization of 3D Scalar Data Fields*, in Proceedings of IEEE Visualization, A. Kaufmann, ed., IEEE Computer Society Press, 1990, pp. 273–280.

[41] P. LACROUTE, *VolPack Volume Rendering Library*. `http://www-graphics.stanford.edu/papers/author.html#PhilippeLacroute`.

[42] P. LACROUTE, *Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation*, PhD thesis, Stanford University, 1995.

[43] P. LACROUTE, *Real-Time Volume Rendering on Shared Memory Multiprocessors Using the Shear-Warp Factorization*. `http://www-graphics.stanford.edu/papers/author.html#PhilippeLacroute`, 1995.

[44] P. LACROUTE and M. LEVOY, *Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation*, in SIGGRAPH, 1994, pp. 451–458.

[45] A. LAMOTHE, *Black Art of 3D Game Programming*, Waite Group Press, 1995.

[46] J. LANDRÉ, *Programming with Intel IPP and Intel OpenCV under GNU Linux: a beginners tutorial*. `http://sourceforge.net/projects/opencvlibrary/`, 2003. 28. Oct 2003.

[47] L. LEMIEUX, R. JAGOE, D. FISH, N. KITCHEN, and D. THOMAS, *A Patient-to-Computed-Tomography Image Registration Method Based on Digitally Reconstructed Radiographs*, Med. Phys., 21 (1994), pp. 1749–1760.

[48] M. LEVOY, *Display of Surfaces from Volume Data*, IEEE Computer Graphics and Applications, 8 (1988).

[49] W. LORENSEN and H. CLINE, *Marching Cubes: A High Resolution 3D Surface Construction Algorithm*, in Computer Graphics Conference Series, SIGGRAPH, 1987, pp. 163–169.

[50] J. MAINTZ and M. VIERGEVER, *A Survey of Medical Image Registration*, Medical Image Analysis, 2 (1997), pp. 1–36.

[51] J. MATSUMURA, D. BREWSTER, M. MAKAROUN, and D. NAFTEL, *A multicenter controlled clinical trial of open versus endovascular treatment of abdominal aortic aneurysm*, Journal of Vascular Surgery, 37 (2003), pp. 262–271.

[52] N. MAX, *Optical Models for Direct Volume Rendering*, IEEE Transactions on Visualization and Computer Graphics, 1 (1995), pp. 99–108.

[53] M. MENARD, D. CHEW, R. CHAN, M. CONTE, M. DONALDSON, J. MANNICK, A. WHITTEMORE, and M. BELKIN, *Outcome in patients at high risk after open surgical repair of abdominal aortic aneurysm*, Journal of Vascular Surgery, 37 (2003), pp. 285–292.

[54] H. MURASE and S. NAYAR, *Visual Learning and Recognition of 3D Objects from Apperance*, Int. Journal of Computer Vision, (1995), pp. 5–24.

[55] NATIONAL ELECTRIC MANUFACTURERS ASSOCIATION, *Digital Imaging and Communications in Medicine (DICOM)*. `http://medical.nema.org/dicom/2003.html`, 2003. 22. Oct 2003.

[56] K. OREND, T. KOTSIS, R. SCHARRER-PALMER, X. KAPFER, F. LIEWALD, J. GORICH, and L. SUNDER-PLASSMANN, *Endovascular Repair of Aortic Rupture due to Trauma and Aneurysm*, European Journal of Vascular and Endovascular Surgery, 23 (2002), pp. 61–67.

[57] K. OREND, R. SCHARRER-PALMER, X. KAPFER, T. KOTSIS, J. GORICH, and L. SUNDER-PLASSMANN, *Endovascular treatment in diseases of the descending thoracic aorta: 6-year results of a single center*, Journal of Vascular Surgery, 37 (2003), pp. 91–99.

[58] G. P. PENNEY, J. WEESE, J. A. LITTLE, P. DESMEDT, D. L. G. HILL, and D. J. HAWKES, *A Comparison of Similarity Measures for Use in 2D-3D Medical Image Registration*, IEEE Trans. Med.Imag., 17 (1998), pp. 586–595.

[59] T. PORTER and T. DUFF, *Compositing Digital Images*, in Proceedings of SIGGRAPH, vol. 18, SIGGRAPH, 1984, pp. 253–259.

[60] J. T. POWELL and R. M. GREENHALGH, *Small Abdominal Aortic Aneurysms*, The New England Journal of Medicine, 348 (2003), pp. 1895–1901.

[61] W. PRESS, S. TEUKOLSKY, W. VETTERLING, and B. FLANNERY, *Numerical Recipes in C — The Art of Scientific Computing, Second Edition*, Cambridge University Press, 1994.

[62] C. REZK-SALAMA, *Volume Rendering Techniques for General Purpose Graphics Hardware*, PhD thesis, Universität Erlangen-Nürnberg, 2001.

[63] M. ROTH, *Intraoperative fluoroskopiebasierte Patientenlageerkennung zur präzisen Unterstützung chirurgischer Eingriffe*, Master's thesis, TU München, 2000.

[64] M. ROTH, *Technical University of Munich*, Personal Communications. November 2003.

[65] S. RUDIN, D. BEDNAREK, and W. WONG, *Accurate Characterization of Image Intensifier Distortion*, Medical Physics, 18 (1991), pp. 1145–1151.

[66] D. RUECKERT, *Image Processing and Analysis - A Practical Approach*, Oxford University Press, 1999, ch. 13.

[67] H. SCHINZ, W. BAENSCH, W. FROMMHOLD, R. GLAUNER, E. UEHLINGER, and J. WELLAUER, *Lehrbuch der Röntgendiagnostik, Allgemeine Grundlagen und Methoden, 6. Auflage*, 1965, ch. 1.

[68] B. SCHUELER and X. HU, *Correction of Image Intensifier Distortion for Three-dimensional X-ray angiography*, SPIE: Medical Imaging, 2432 (1995), pp. 272–279.

[69] D. SOIMU, C. BADEA, and N. PALLIKARAKIS, *A Novel Approach for Distortion Correction for X-Ray Image Intensifiers*, Computerized Medical Imaging and Graphics, 27 (2003), pp. 79–85.

[70] G. TEITELBAUM, W. BRADLEY, and B. KLEIN, *MR imaging artifacts, ferrromagnetism, and magnetic torque of intravascular fitlers, stents, and coils*, Radiology, 166 (1998), pp. 657–664.

[71] J. TRAUB, *Design of an Intra-operational Augmented Reality Enhanced Port Placement and Tracking of Telemanipulatorarms in Heart Surgery*, Master's thesis, Technical University of Munich, November 2003.

[72] R. TSAI, *A Versatile Camera Calibration Technique for High Accuracy 3D Machine Vision Metrology using Off-the-Shelf TV Cameras and Lenses*, Computer Science 51342, IBM, 1 1985.

[73] M. TUCERYAN, Y. GENC, and N. NAVAB, *Single point active alignment method (SPAAM) for optical see-through HMD calibration for augmented reality*, Proceedings of the IEEE and ACM International Symposium on Augmented Reality, (2000), pp. 149–158.

[74] A. WATT and F. POLICARPO, *Games - Animation and Advanced Real-Time Rendering*, vol. 2, Addison-Wesley, 2003.

[75] J. WEESE, R. GÖCKE, G. P. PENNEY, P. DESMEDT, T. BUZUG, and H. SCHUMANN, *A fast voxel-based 2D/3D registration algorithm using a volume rendering method based on the shear warp factorization*, in SPIE International Symposium on Medical Imaging, San Diego, USA, 1999.

[76] W. WEIN, *Implementation and Validation of Intensity Based 3D-2D Registration Algorithm for Radiation Therapy*, December 2003.

[77] R. WESTERMANN and T. ERTL, *Efficiently Using Graphics Hardware in Volume Rendering Applications*, tech. rep., Computer Graphics Group, Universität Erlangen-Nürnberg, Germany, 1998.

[78] L. WESTOVER, *Footprint Evaluation for Volume Rendering*, Computer Graphics, 24 (1990).

[79] J. WILHELMS and A. V. GELDER, *Octrees for Faster Iso-surface Generation*, in Transactions on Graphics, ACM, 1992, pp. 201–227.

[80] R. WILLSON, *Tsai Camera Calibration Software*. `http://www-2.cs.cmu.edu/afs/cs.cmu.edu/user/rgw/www/TsaiCode.html`, October 1995.