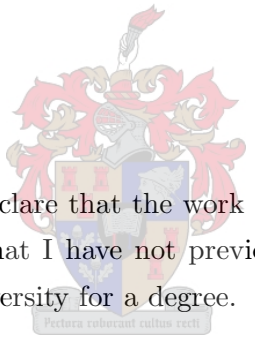
Verification of patient position for proton therapy using Portal X-Rays and Digitally Reconstructed Radiographs

Leendert van der Bijl

Thesis presented in partial fulfilment
of the requirements for the degree of
MSc.Eng
at the University of Stellenbosch

Dr. N. Muller
December 2006

Declaration



I, the undersigned, hereby declare that the work contained in this thesis is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

Signature: ..... Date: .....

Abstract

This thesis investigates the various components required for the development of a patient position verification system to replace the existing system used by the proton facilities of iThemba LABS[1]. The existing system is based on the visual comparison of a portal radiograph (PR) of the patient in the current treatment position and a digitally reconstructed radiograph (DRR) of the patient in the correct treatment position. This system is not only of limited accuracy, but labour intensive and time-consuming. Inaccuracies in patient position are detrimental to the effectiveness of proton therapy, and elongated treatment times add to patient trauma. A new system is needed that is accurate, fast, robust and automatic.

Automatic verification is achieved by using image registration techniques to compare the PR and DRRs. The registration process finds a rigid body transformation which estimates the difference between the current position and the correct position by minimizing the measure which compares the two images. The image registration process therefore consists of four main components: the DRR, the PR, the measure for comparing the two images and the minimization method.

The ray-tracing algorithm by Jacobs was implemented to generate the DRRs, with the option to use X-ray attenuation calibration curves and beam hardening correction curves to generate DRRs that approximate the PRs acquired with iThemba LABS's digital portal radiographic system (DPRS) better.

Investigations were performed mostly on simulated PRs generated from

[1]Laboratory for accelerator based sciences

DRRs, but also on real PRs acquired with iThemba LABS's DPRS.
The use of the Correlation Coefficient (CC) and Mutual Information (MI) similarity measures to compare the two images was investigated.
Similarity curves were constructed using simulated PRs to investigate how the various components of the registration process influence the performance. These included the use of the appropriate XACC and BHCC, the sizes of the DRRs and the PRs, the slice thickness of the CT data, the amount of noise contained by the PR and the focal spot size of the DPRS's X-ray tube.
It was found that the Mutual Information similarity measure used to compare 1024^2 pixel PRs with 256^2 pixel DRRs interpolated to 1024^2 pixels performed the best. It was also found that the CT data with the smallest slice thickness available should be used. If only CT data with thick slices is available, the CT data should be interpolated to have thinner slices.
Five minimization algorithms were implemented and investigated. It was found that the unit vector direction set minimization method can be used to register the simulated PRs robustly and very accurately in a respectable amount of time.
Investigations with limited real PRs showed that the behaviour of the registration process is not significantly different than for simulated PRs.

Pectora roborant cultus recti

Opsomming

Hierdie tesis ondersoek die verskeie komponente benodig vir die ontwikkeling van ‘n pasiëntposisie-verifikasiestelsel wat die stelsel wat huidiglik deur iThemba LABS[2] se proton-fasiliteite gebruik word, kan vervang. Die stelsel wat huidiglik gebruik word is gebaseer op die visiële vergelyking van ‘n portaal X-straalbeeld (PX) van die pasiënt in die huidige posisie en ‘n digitaal-gerekonstrueerde X-straalbeeld (DGX) van die pasiënt in die korrekte posisie. Die huidige stelsel het nie net beperkte akkuraatheid nie, maar is arbeidsintensief en langdurig. Onakkuraatheid in die pasiënt se posisie is baie nadelig vir die effektiwiteit van protonterapie, terwyl verlengde behandelingstye die trauma wat die pasiënt ervaar verhoog. ‘n Nuwe stelsel wat akkuraat, vinnig, robuus en outomaties is word benodig.

Outomatiese verifikasie kan gedoen word deur beeldregistrasie-tegnieke te gebruik om die PX en DGX te vergelyk. Die registrasieproses soek na ‘n rigiede-liggaam-transformasie wat die verskil in die huidige en korrekte posisie benader deur die maatstaf van hoe goed die twee beelde vergelyk, te minimiseer. Die registrasieproses bestaan dus uit vier hoof komponente: die DGX, die PX, die maatstaf om die twee beelde te vergelyk en die minimisasiemetode.

Die “ray-tracing” algoritme van Jacobs is geïmplimenteer om DGX’e te produseer, met die opsie om X-straalattenuasie-kalibrasiekurwes (XAKK’s) en bundelverharding-korreksiekurwes (BVKK’s) te gebruik. Die gebruik van hierdie kurwes produseer DGX’e wat die PX’e (wat deur die digitale portaal X-straalbeeld-stelsel (DPXS) van iThemba LABS geneem word) beter

[2]Laboratory for accelerator based sciences

benader.
Ondersoeke is hoofsaaklik gedoen op gesimuleerde PX'e wat vanaf DGX'e gegenereer is, maar ook op ware PX'e wat deur die DPXS van iThemba LABS geneem is.
Die gebruik van die korrelasie koëffisiënt en "Mutual Information" similariteits-maatstawwe is ondersoek om die twee beelde te vergelyk.
Om te ondersoek hoe sy verskeie komponente die registrasieproses se werkverrigting beïnvloed, is similariteitskurwes met die gesimuleerde PX'e gekonstrueer. Die komponente wat ondersoek is sluit die gebruik van die geskikte XAKK en BVKK, die onderskeie groottes van die DGX'e en die PX'e, die snit dikte van die CT data, die hoeveelheid geraas op die PX'e en die grootte van die fokus-area van die DPXS se X-straalbuis.
Dit is gevind dat die "Mutual Information" similariteits-maatstaf, wanneer dit gebruik is om 1024^2 piksel PX'e met 256^2 piksel DGX'e geïnterpoleer tot 1024^2 piksels te vergelyk, die beste gewerk het. Dit is ook gevind dat die CT data met die dunste beskikbare snitte gebruik moet word. Indien slegs CT data met dik snitte beskikbaar is, moet die CT data geïnterpoleer word om dunner snitte te verkry.
Vyf minimisasie algoritmes is geïmplementeer en ondersoek. Dit is gevind dat die eenheidsvektor-rigtingstel-metode baie robuus en akkuraat is in die registrasie van die gesimuleerde PX'e, en dat registrasie met hierdie metode in 'n bevredigende tyd gedoen kan word.
Ondersoeke gedoen op 'n beperkte aantal ware PX'e het getoon dat die gedrag van die registrasieproses nie wesenlik verskil van die gedrag met gesimuleerde PX'e nie.

Acknowledgements

I would like to thank the following people:

Evan de Kock, my supervisor at iThemba LABS, for his advice and guidance over the past two years.

My academic supervisor, Dr. Neil Muller from the University of Stellenbosch, for his guidance during my research.

Roy Mlambo, for the time he took to help me with taking the portal radiographs for my investigations.

Ithemba LABS, for the use of their facilities and the funding they provided me.

Ms. Elize Fick, for helping me with language corrections of this thesis.

All my family and friends and especially Arabella, for their continuous loving support.

Acronyms

2D Two Dimensional

3D Three Dimensional

BFGS Broyden-Fletcher-Goldfarb-Shanno

BHCC Beam Hardening Correction Curve

CC Correlation Coefficient

CCD Charged Coupled Device

CT Computed Tomography

DPRS Digital Portal Radiographic System

DRR Digitally Reconstructed Radiograph

ESF Edge Spread Function

FRPR Fletcher-Reeves-Polak-Ribiere

FSS Focal Spot Size

LSF Line Spread Function

FWHM Full Width at Half Maximum

MB Mega Bytes

MI Mutual Information

MTF Modulation Transfer Function

PDS Powell's Direction Set

PR Portal Radiograph

PSF Point Spread Function

RBTF Rigid Body Transformation

ROI Region Of Interest

SD Steepest Descent

SNR Signal to Noise Ratio

SPG Stereophotogrammetry

UVDS Unit Vector Direction Set

VOI Volume Of Interest

Voxel Volume Element

XACC X-ray Attenuation Calibration Curve

Contents

Pectora roborant cultus recti

Pectora roborant cultus recti

Pectora roborant cultus recti

List of Tables

List of Figures

Pectora roborant cultus recti

Pectora roborant cultus recti

Chapter 1

Introduction

Cancer is a disease that about 30% of people in industrialized countries suffer from and about half of these people eventually die from the disease. Cancer can be broadly defined as the uncontrolled growth and reproduction of a group of cells, called a tumor. Cancer cells from the primary tumor can metastasize (spread to other parts of the body) and about 30% of patients have metastases at diagnosis. What a cancer cell actually is and how it became that way is still not fully understood [1][2, Chapter 34].

The three main forms of treating cancer are surgery, chemotherapy and radiotherapy. These treatments may be used in isolation or together, depending on the particular cancer in a given patient. The success of treatment is measured in terms of the resulting survival rate. A patient who survives for five years after treatment without further symptoms is said to be cured. The overall 5-year survival rate of all treated cancer sufferers is about 45%. Radiotherapy is a localized form of treatment that is used to treat the primary tumor and is (possibly combined with surgery) responsible for 40% of all cancer cures [1][2, Chapter 34].

While radiotherapy is sometimes used to treat other diseases as well (e.g. see [3]), it is mainly used for cancer treatment. The reason for the relative success of radiotherapy in treating cancer is that cancerous cells are more sensitive to ionising radiation than normal cells within the same organ.

There exist many types of radiation, that includes radiation with photons (X-rays), electrons, neutrons and protons. These types of radiation differ in

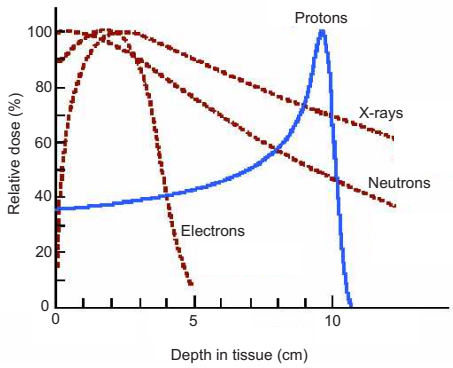


Figure 1.1: *Depth dose curves for different radiation types.*

many aspects, most importantly in the dose distributions they have. Figure 1.1 shows typical relative dose distribution curves for the different radiation types. Each distribution curve shows what percentage of the dose is delivered at a specific depth into the patient tissue. Note the relatively low entrance dose and high dose peak at the end of the range for proton radiation.

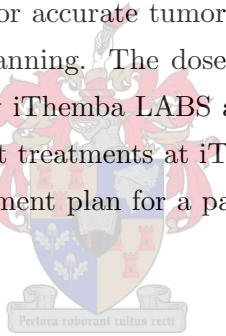
Proton beams are used for radiotherapy mainly because of their physical selectivity resulting from their unique dose distribution as displayed in Figure 1.1, which gives them the ability to conform the dose to the target volume. The use of protons for radiotherapy because of the mentioned dose characteristics was first suggested by Wilson [4] in 1946. Because of their physical selectivety, proton beams are best utilized for eradicating well-delineated lesions close to critical structures. For proton therapy to be effective it is required to have an accurate beam delivery system, precise tumor and critical structure localization, accurate and reproducible patient setup, and accurate 3D treatment planning [1][5][2, Chapter 34].

1.1 Proton beam radiotherapy at iThemba LABS

1.1.1 Current proton therapy facilities

Proton beam therapy commenced at iThemba LABS[1] in September 1993 using a fixed horizontal beam line. A double scattering beam delivery system is used to provide a uniform dose at the treatment position, while field-specific collimators and variable thickness propellers are used to modulate the lateral dimensions and depth extent of the dose peak [5]. The result is a treatment dose that conforms well to the target volume.

The treatment planning system currently used at iThemba LABS is called the *VOXELPLAN* system. The *VOXELPLAN* system uses the *VIRTUOS* (*VIRTU*al Radi*O*therapy *S*imulator) developed by the German Cancer Research Centre in Heidelberg, Germany. This system uses the patient's computed tomography (CT) data for accurate tumor and critical structure delineation and 3D treatment planning. The dose calculations are done by a software system developed by iThemba LABS and is incorporated in the *VOXELPLAN* system [6]. Most treatments at iThemba LABS are given in several fractions, and the treatment plan for a patient therefore consists of several treatment fields.



Current patient positioning system

Patients are currently placed on a motorized treatment chair for treatment. The patient is fixed to the chair by immobilization devices, and can therefore be positioned by moving the chair.
Patients are positioned by an unique patient positioning system based on real-time digital multi-camera stereophotogrammetry (SPG) techniques. Patients are fitted with custom made plastic masks, that carry radiopaque and retro-reflective markers. The positions of the retro-reflective markers that are detected with charge coupled device (CCD) cameras, are calculated and used to move the treatment chair automatically. The patient can be posi-

[1] Laboratory for accelerator based sciences

tioned in this way with the tumor aligned with the beam with an accuracy of about 1mm.
The positioning system was originally designed to treat patients in the sitting position. Although it was later altered to treat patients in the lying position (with only a limited capability of roll and tilt) these limitations, together with the fact that only a horizontal beam is available, restrict the types of lesions that can be treated [5][6].

Current verification of patient position

The patient also wears the plastic mask during the pre-treatment acquisition of its CT data (which is used for treatment planning), so that the position of the markers in relation to the patients anatomy can be established. The masks are always fitted as tight as possible to the patient in order to minimize movement while avoiding patient discomfort. Since there might be small differences in the way the mask is fitted to the patient for the pre-treatment CT scan and for actual treatment, it is necessary to verify that the patient's anatomy is correctly positioned according to the treatment plan before the treatment is started.
Currently, this verification is done by visual comparison of a portal radiograph (PR) taken of the patient in the current treatment position and a digitally reconstructed radiograph (DRR) obtained from the treatment planning system of the patient in the correct treatment position. The PR is obtained by placing an X-ray tube upstream of the collimator in the beam line and taking a radiograph of the patient on an X-ray film placed behind the patient. The DRR is a simulated radiograph and has a view that is similar to the PR. This is a very manual and time-consuming process and needs to be repeated for each of the treatment fields [6].

1.1.2 New proton therapy facilities

iThemba LABS is currently expanding its proton therapy facilities by developing additional proton therapy rooms. The new rooms may either be

fitted with a single fixed horizontal beam line, or with two fixed beam lines at a specific angle.

New patient positioning system

The patient positioning system for each of the new proton therapy rooms will use a commercial robot together with a detachable rigid chair or couch-top for patient support. The system will also make use of SPG techniques for automatic patient positioning. This system will be extremely versatile and allow a wider range of lesions to be treated at a higher throughput rate [5][6].
The new therapy rooms will also each be fitted with a newly designed digital portal radiographic system (DPRS) [7]. This system makes it possible to acquire PRs in digital form, and will thus be much faster than the current system.

1.2 Aim of this thesis

The current verification of patient position which is based on visual inspection by a physician has many shortcomings. Visual inspection is not only of limited accuracy, but labour intensive and time-consuming [8][9]. Inaccuracies in patient position are detrimental to the effectiveness of proton therapy, and elongated treatment times add to patient trauma. A new system is needed that is both accurate and automatic.
The aim of this thesis is to solve the problem of automatically verifying the patient position. The solution to the problem should be such that the verification system that utilizes it be accurate, fast, robust and requires as little human intervention as possible.

1.3 A proposed solution

The problem of verifying the patient position is essentially the problem of calculating the current treatment position. The current treatment position is unknown, because the small error with which the mask is fitted to the patient is unknown.
The deviation of the current treatment position from the correct treatment position can be described by a rigid body transformation, called the error transformation $\mathbf{T}_{error}$. The size of the error of the current treatment position can be quantified from $\mathbf{T}_{error}$.

The patient position can thus be verified as follows:

- If the error calculated from $\mathbf{T}_{error}$ is within acceptable limits, the patient is regarded as being correctly positioned
- If the error is not within acceptable limits, $\mathbf{T}_{error}$ is communicated to the SPG system to adjust the patient position.

Because the mask is reasonably tight fitting the scope for misalignments is small. Therefore the components of $\mathbf{T}_{error}$ are assumed to be smaller than 5mm (for the translations) and 5° (for the rotations).

The transformation $\mathbf{T}_{error}$ can be estimated by a process called image registration. Image registration establishes spatial correspondence between one coordinate space and another coordinate space [10]. The result of image registration is a transformation between the coordinate systems that describes the two spaces. Image registration has been used in various cases (see e.g. [9],[11] and [12]) to estimate the errors in patient setup.

Image registration is done by comparing the PR of the patient in the current treatment position with DRRs generated from the patient's CT data. Each DRR is generated according to a specific rigid body transformation $\mathbf{T}_{DRR}$, that results in the DRR having a particular view of the patient. When the transformation $\mathbf{T}_{DRR}$ is zero, the DRR view will be that of the patient in the correct treatment setup according to the treatment plan. When the trans-

formation $\mathbf{T}_{DRR}$ is a good estimate of $\mathbf{T}_{error}$, the DRR view will resemble the view of the PR very closely.

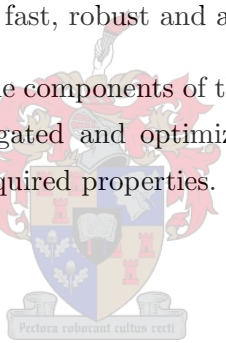
To calculate an estimation of the transformation between the current and correct treatment setup, an objective function P can be formed:

$$P\left(\mathbf{T}_{DRR}\right) = M\left(PR, DRR\left(\mathbf{T}_{DRR}\right)\right), \tag{1.1}$$

where PR is the PR of the patient in the current treatment position, $DRR(\mathbf{T}_{DRR})$ is the DRR according to the transformation $\mathbf{T}_{DRR}$ and M is a measure for how well the two images compare. The estimation of the error transformation $\mathbf{T}_{error}$ can now be calculated by optimizing the objective function P.

The problem of verifying the patient position is therefore solved by optimizing the objective function P, and should be done in such a way that the verification process is accurate, fast, robust and automatic.

In this thesis, the effects that the components of the objective function have on its optimization are investigated and optimized in order to achieve a verification process with the required properties.



1.4 Thesis outline

Background information on the concepts important to the image registration process is discussed in Chapter 2. The generation of DRRs, that is central to the image registration process, is discussed in Chapter 3. The details of the image registration process are discussed in Chapter 4. In Chapter 5 the experiments on various minimization methods and their results are discussed. Chapter 6 gives a conclusion of all the findings of the thesis.

Chapter 2

Background

2.1 Acquiring the PR

As stated in the introduction, the new patient positioning system to be used by iThemba LABS acquires the PRs of the patient in the treatment position with the DPRS developed in [7]. Before the details of this system are discussed, a short introduction to radiography is appropriate.

2.1.1 The radiographic process

Radiography is a diagnostic investigation, where an image is produced of the structures within a patient [2, Chapter 2]. Radiography utilizes electromagnetic waves called X-rays, that have sufficient energy to penetrate through a patient's body. X-rays travel in straight lines and cannot be easily focused by mirrors and lenses like visible light, and therefore only shadow-like images can be produced. X-rays cannot be seen by the human eye, and are detected with photographic film or detector devices like scintillation screens.

The process of acquiring a radiograph is shown in Figure 2.1. X-rays are emitted in all directions from the X-ray tube's anode. All the X-rays are absorbed, except for the radiation in the direction of the patient. The X-rays emitted in the direction of the patient travel through the patient where some are absorbed, some are deflected and some are left to reach the detector. The

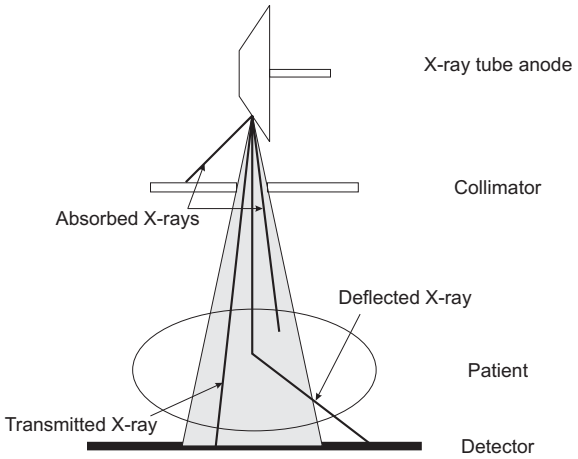


Figure 2.1: *The radiographic process.*

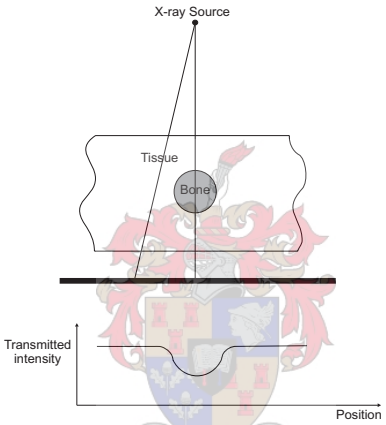


Figure 2.2: *X-rays going through the bone structure are attenuated more than those going only through the tissue.*

X-ray beam is said to be attenuated. Different materials attenuate X-rays differently, e.g. soft tissues attenuate X-rays more than air but less than bone and metal (Figure 2.2).

The intensity level of the radiograph (the transmitted intensity) at a specific position is directly related to the sum of all the attenuation caused by the material in a straight line between that position and the X-ray source. This is called the radiological path-length. As some X-rays travel through different anatomical structures inside the patient than others, they have different radiological path-lengths. This results in different intensity levels on the

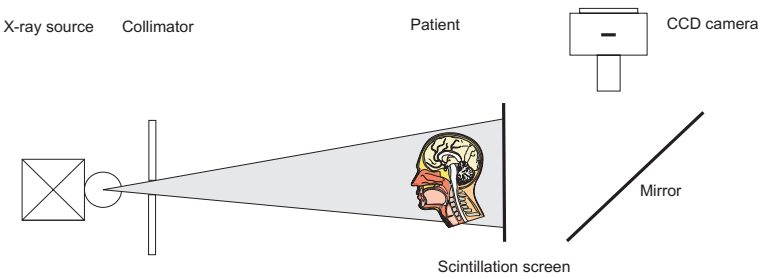


Figure 2.3: *Physical arrangement of the digital portal radiographic system.*

radiograph.

2.1.2 The DPRS

The DPRS to be used by iThemba LABS consists of an X-ray tube for producing the X-rays, a scintillation screen for converting the X-rays to visible light, a protected aluminium front silvered mirror to direct the light and a CCD camera to capture the light image in digital form [7]. A diagram of this system is shown in Figure 2.3.

The mountings of both the X-ray tube and the imaging device will allow both units to be easily moved into and out of the beam line. This makes it possible to take a PR of the patient in the treatment position just before the treatment is started [6].

The main advantage of the new digital system over the old system (that uses X-ray films) is the short time in which an image can be obtained. Other advantages include the optimization of image contrast, reduced patient radiation dose, lower operating costs and more efficient archiving and accessing of the images. The spatial resolution of the new system is, however, not as good as that of the X-ray film system. This is because of the light scatter between the scintillation screen and the CCD camera, and the limited resolution of the CCD camera [7].

It should be noted that the X-ray tube has certain parameters that can be set to achieve a certain X-ray spectrum. This has an effect on the quality of the acquired PR. These parameters and their importance to DRR generation

are discussed in Section 3.2.

2.1.3 The PR

The PR captured by the CCD camera is an intensity image. This intensity image is corrected for the various unwanted effects produced by the CCD camera by doing the appropriate bias field, dark field and flat field subtractions. After these subtractions, the image is also corrected for radial lens distortions.

The internal CCD array of the camera consists of 1024×1024 square pixels, and hence the PR also consists of 1024×1024 square pixels. Physically, each pixel of the PR represents a square area of the scintillation screen. Since the size of the scintillation screen is 300mm×300mm, which is then also the dimensions of the PR, the size of each pixel is 0.29mm^2.
PRs that consists of 512^2- and 256^2 pixels can be produced by the CCD camera by the binning of pixels.

2.2 Acquiring the DRR

A DRR is a computer generated projection image created from the CT data of a patient. A brief discussion of CT data and the coordinate systems involved during DRR generation is now given.

2.2.1 CT data

The CT data of a patient is acquired with a CT study during which a CT scanner is used to obtain a 3D image of the patient. The resulting CT data consists of a set of image slices (Figure 2.4). Each slice has a specific thickness, and slices are a specific distance apart. Each slice consists of a large number of pixels and each pixel has an image value assigned to it. A typical CT data set might consist of 50 slices, with each slice consisting of 256×256 pixels. Modern CT scanners are able to create sets of several

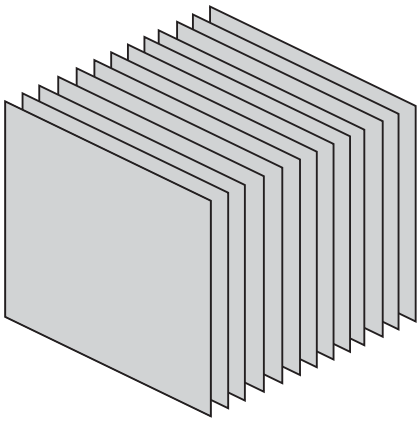


Figure 2.4: *Set of CT image slices.*

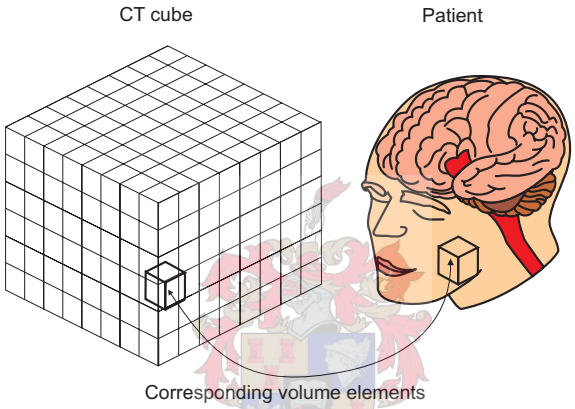


Figure 2.5: *CT cube representing patient anatomy.*

hundred slices of 1024×1024 pixels. Together, the slices of the CT data form a 3D volume called the CT cube that is divided into a large number of small volume elements (voxels).

Each voxel represents a tiny volume of the patients anatomy (Figure 2.5), and the value assigned to the specific voxel (the CT number) represents some physical property of that tiny volume. The CT number is often called the Hounsfield number. With X-ray computed tomography, the CT numbers are a measure of the X-ray attenuation properties of the material represented and given by the Hounsfield number

$$H = 1000 \left(\frac{\bar{\mu}_m - \bar{\mu}_w}{\bar{\mu}_w} \right),$$

where the quantity $\bar{\mu}_m$ is a measure of the X-ray attenuation of the material represented by the voxel. As will be discussed in Section 3.2, the X-ray attenuation coefficient $\bar{\mu}_m$ depends on the specific CT scanner used.
The values of $\bar{\mu}_m$ are calculated by tomographic reconstruction of the projection images produced by the CT scanner. The value of $\bar{\mu}_w$, a measure of the X-ray attenuation of water, is obtained during a calibration process of the CT scanner in which a water phantom is scanned. The values of the CT-numbers typically range from -1000 (for air) to 1500 (for bony structures).

2.2.2 Coordinate systems

The positions of each voxel of the CT data, each pixel of the PR and the X-ray source are required during DRR generation. Each position is known in the coordinate system associated with the corresponding system:

1. The position of each voxel of the CT data is known in the CT scanner coordinate system
2. The position of the X-ray source and the positions of the pixels of the PR is known in the DPRS coordinate system.

The coordinates of these positions need to be transformed to a common coordinate system (called the DRR coordinate system) for use by the DRR generation process. This transformations is shown in Figure 2.6.
In the DRR coordinate system the sides of the CT cube are aligned with the x-, y- and z-axes and the top left corner of the first image slice is located at the origin of the DRR coordinate system.
The coordinates of the X-ray source and the coordinates of the pixels of the PR are also transformed to the DRR coordinate system. This is done in such a way that the direction of the beam central axis (defined as the y-axes of the DPRS) and the position of the treatment room isocenter (defined as the origin of the DPRS) relative to the CT cube agree with the treatment plan for the patient for the desired treatment field. All the transformations are rigid body transformations, which are discussed in Section 2.3.

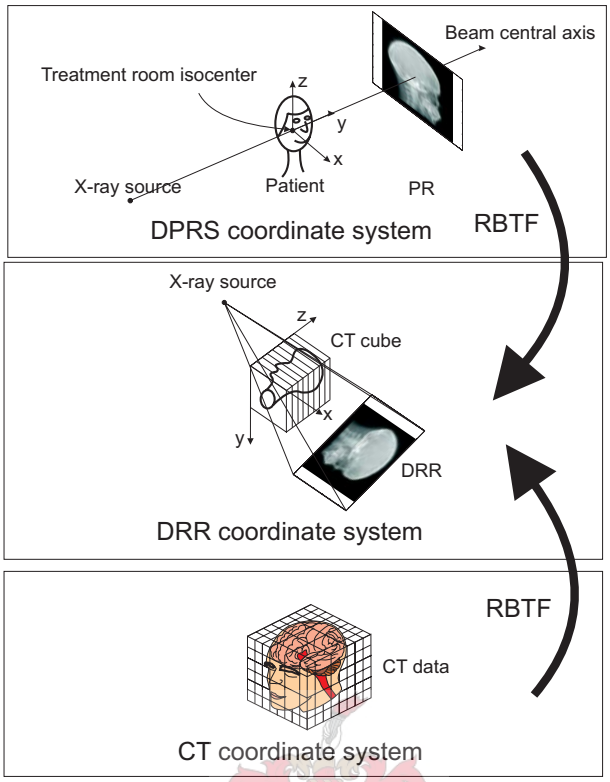


Figure 2.6: *Rigid body transformations between coordinate systems.*

A projection of the CT data from the X-ray source to the pixel positions in the DRR coordinate system will result in an image that has a view of the patient that is very similar to that of the PR.

2.2.3 The DRR

There exist many methods for generating DRRs (see Section 3.7). The most popular method is the ray-tracing algorithm developed by Siddon [13]. This method, discussed in detail in Section 3.1, is also the standard against which all other DRR generation methods are measured. The result of DRR generation is an intensity image that resembles a radiograph.

Since the DRRs will be compared with PRs during the evaluation of the objective function P in (1.1), the aim of DRR generation is to produce an image that is as similar as possible to the PRs obtained with the DPRS.

To generate DRRs that are very similar to PRs, the most prominent physical processes involved in taking a radiograph are modelled and included in the ray-tracing algorithm. Section 3.2 introduces the X-ray attenuation calibration curve (XACC) and Section 3.3 introduces the beam hardening correction curve (BHCC).
While the modelling of these physical processes results in more realistic DRRs, it adds to the complexity of the ray-tracing algorithm. This can have a negative impact on the speed of DRR generation.

With DRR generation there is always a trade-off between speed and quality, and these two properties have an effect on the optimization of the objective function. For each evaluation of the objective function P, a new DRR must be generated according to the transformation $\mathbf{T}_{DRR}$. During the optimization process P could be evaluated many times, depending on the success of the optimization algorithm. It is therefore critical that DRR generation be as fast as possible.
On the other hand, for the transformation $\mathbf{T}_{error}$ calculated by the optimization of P to be a good estimate of the real error in patient position, the DRRs need to be of an acceptable quality.

The physical size of the DRR and the number of pixels for which the DRR should be calculated can be specified in the ray-tracing algorithm. These parameters determine the quality of the DRR (influenced by the **size** of each pixel of the DRR) and speed of DRR generation (influenced by the **number** of pixels that the DRR consists of). The speed and quality of the DRR is also tied to the size of the voxels of the CT cube used by the ray-tracing algorithm.

The size and dimensions of both the DRRs and the CT data can be artificially altered by using interpolation techniques. As explained in the previous paragraph, this will influence the speed of DRR generation and the quality of the DRRs and therefore the optimization of the objective function P. In Section 3.4 these interpolation techniques are discussed in more detail.

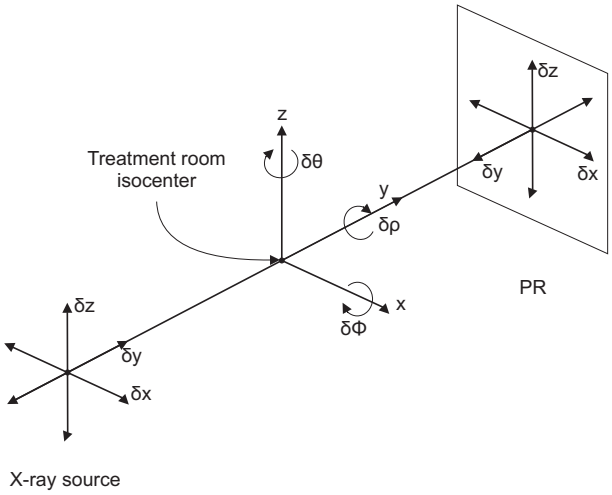


Figure 2.7: *Parameters of* $\mathbf{T}_{DRR}$ *in the DPRS coordinate system.*

2.3 The transformation $\mathbf{T}_{DRR}$

The transformation $\mathbf{T}_{DRR}$ is taken to be a rigid body transformation (RBTF). The RBTF model is the most constrained transformation model for medical imaging [14] and asserts that distances and angles within the spaces remain unchanged. The model therefore assumes that objects are internally immutable and is most appropriate for images that contain bony structures that cannot be deformed (such as the bony skeleton of the head).

A 3D RBTF is defined by three translation parameters and three rotation parameters. Therefore, $\mathbf{T}_{DRR}$ is defined by

$$\mathbf{T}_{DRR} = \{\delta_x, \delta_y, \delta_z, \delta_\phi, \delta_\rho, \delta_\theta\}\,,$$

where δ_x, δ_y and δ_z are translations along the x-, y-, and z-axes of the DPRS coordinate system and δ_ϕ, δ_ρ and δ_θ are rotations around these three axes respectively (see Figure 2.7). It should be noted that this transformation is incorporated when transforming the coordinates of the X-ray source and the PR pixels from the DPRS coordinate system to the DRR coordinate system (see Section 2.2.2). The coordinates of the voxels of the CT cube are not affected by this transformation, hence the X-ray source and DRR pixels have a different position relative to the CT cube for a different $\mathbf{T}_{DRR}$. This

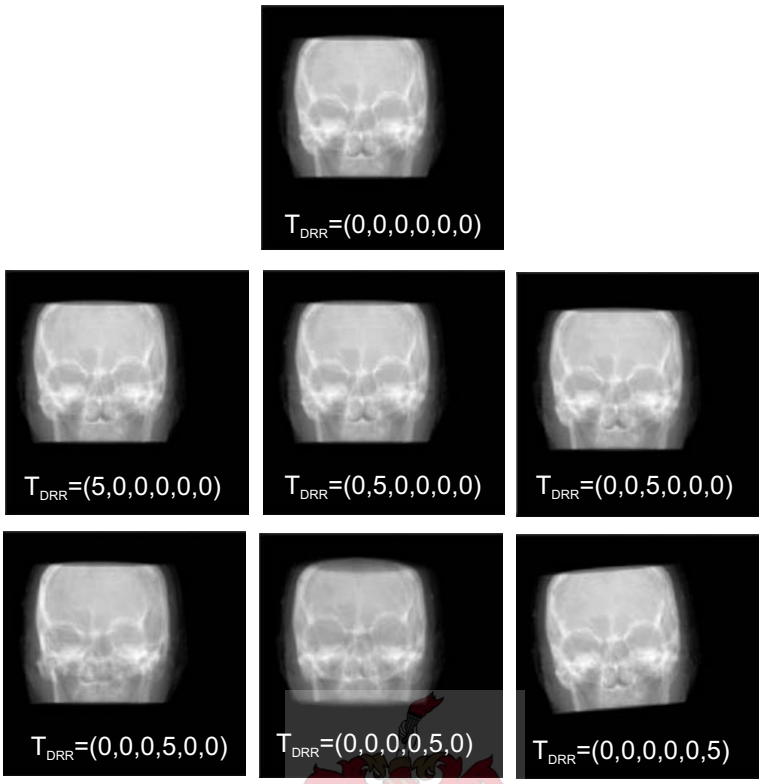


Figure 2.8: *Effects of parameters of* $\mathbf{T}_{DRR}$ *on DRR view.*

results in a DRR with a different view. Figure 2.8 illustrates the effect that changes in the parameters of $\mathbf{T}_{DRR}$ have on the DRR view. Note that the effect that the deviations of 5mm and 5° (which are considered the extreme case) are reasonably small on the DRR view.

The translation parameters δ_x and δ_z result in the X-ray source and PR pixels being shifted horizontally and vertically (respectively) relative to the plane in which the PR pixels reside. These translations are therefore called in-plane, and result in a horizontal and vertical shift in the DRR view.
The translation parameter δ_y results in the X-ray source and PR pixels being shifted closer and further away from the treatment room isocenter. This shift is perpendicular to the PR plane and is therefore called out-of-plane. The out-of-plane translation δ_y results in a magnification of the DRR view.
The rotation parameter δ_ρ rotates the coordinates around the y-axes of the

DPRS. The PR pixels remain in the same plane during this kind of rotation, and therefore it is called an in-plane rotation. The parameters δ_ϕ and δ_θ result in rotation around the x- and z-axes respectively. Since the PR pixels will move out of their original plane, these rotations are called out-of-plane.

While small changes in the in-plane translations and rotation are easily noticeable in the DRR view, this is not the case for the out-of-plane translation and rotations. Detecting small changes in the out-of-plane translation δ_y is particularly difficult, since it only results in a small magnification of the DRR view.

As stated in the Section 1.3, experience with the existing portal verification system used by iThemba LABS has shown that the errors in the placement of the treatment mask are usually much less than the extreme bounds of 5mm for each translation and 5° for each rotation. The parameters of $\mathbf{T}_{DRR}$ will therefore be bounded to -5mm to 5mm for the translation parameters, and -5° to 5° for the rotation parameters.

2.4 The measure M

Selecting the measure M with which the two images are to be compared should be done very carefully, since it has a direct influence on the nature of the objective function P.
As stated in the introduction, the objective is to make the verification process as fast and accurate as possible, with as little human interaction as possible. To achieve these objectives, the objective function P needs to be as smooth as possible with a well defined global optimum so that the optimization process will converge fast to the correct optimum value of P. A measure must be found that yields such an objective function.

Image registration measures can be divided into landmark based, surface based or intensity based measures [10][15]. While landmark based measures operate on corresponding points in the two images and surface based measures operate on corresponding surfaces in the two images, intensity based

measures operate on the intensity values of all the pixels in the two images.

Landmark based and surface based measures both involve a pre-processing stage during which the corresponding features are extracted from the images. This stage often requires user input in the form of segmenting landmarks or other features. Intensity based similarity measures do not require this pre-processing stage, since they operate on all the intensity values of the pixels of the two images, and are therefore fully automatic. Since a fully automatic method is desirable, only intensity based measures are considered.

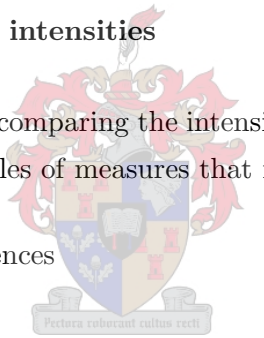
2.4.1 Intensity based similarity measures

Intensity based similarity measures can be divided into three main groups [16], based on how the intensity values are used to calculate it.

Measures based on image intensities

These measures are based on comparing the intensity values of pixels on the two images pair wise. Examples of measures that fall into this group are:

- Sum of Intensity Differences
- Correlation Coefficient.



Measures based on image intensities and spatial information

These measures incorporate spatial information by taking the information of the pixels in the neighbourhood into account. This group includes the following measures:

- Pattern Intensity
- Gradient Correlation
- Gradient Difference
- Sum of Local Normalized Correlation.

Statistically based measures

These measures assume a statistical dependence between the two images and attempt to measure it. They aim to measure how well one image explains another, or to maximize the shared information of the two images. These measures are also called Information Theoretic Measures and Shannon's entropy principle form an important part of these measures [16].
Many of these methods are based on the calculation of the two dimensional histogram of the two images, which is created by counting the number of occurrences of every pair of intensities in the two images [16]. Examples of measures in this group include:

- Entropy of the difference image
- Mutual Information
- Correlation Ratio
- Probability Distribution Estimation.

Studies were performed by iThemba LABS on a number of similarity measures [17]. The way that the similarity measures behaved in the vicinity of the correct position was investigated. The results showed that of the measure considered, the Correlation Coefficient and Mutual Information similarity measures are the most appropriate for this application. These two measures provide a match output that degrades smoothly in the vicinity of the correct position, and do not suffer from local minima. Hence they are suitable for conventional numerical optimization.

2.5 The optimization algorithm

Since the transformation $\mathbf{T}_{DRR}$ has six degrees of freedom, three translational and three rotational, the optimization of the objective function P is not an simple task.
The objective function should either be minimized or maximized to find the

optimum value, depending on how the measure M indicates how well the two images match. Some measures will return a lower value for two images that match better, while others will return a higher value for two images that fit better. In this thesis, the objective function will be optimized by using minimization algorithms. This means that for measures that return higher values for a better match $-M$ will be used, and for measures that return a lower value for a better match M will be used.

There are many types of minimization problems, depending on the nature of the objective function and its variables. Typically, the following cases are considered [18, Chapter 1]:

- In **Continuous minimization** the values of the variables of the objective function are real numbers, while some or all of the variables of the objective function can take on integer values in **Discrete minimization**.
- In **Constrained minimization**, the variables can take on only certain values or ranges of values, while in **Unconstrained minimization** the variables are free to take on any value.
- **Local minimization** only finds the lowest objective function value in the vicinity of where it started looking, while **Global minimization** finds the lowest function value for all possible values of the variables.
- **Stochastic minimization** differs from **Deterministic minimization** in that the underlying model contains uncertainties, i.e. the objective function might not always return the exact same value for a certain value of its variables.

Since the translational and rotational components of the objective function's variable $\mathbf{T}_{DRR}$ can take on any real value, and the objective function will always return the same value for a set of variables, the minimization problem to be dealt with is continuous, unconstrained and deterministic. **Global minimization** (like population based minimization) is very expensive. The studies performed in [17] showed that the Correlation Coefficient and Mutual

information similarity measures can yield objective functions that is convex or that can be well approximated by convex functions in the vicinity of the correct position. Since the local minimum of a convex function is also the global minimum [18, Chapter 1] and the range of possible deviations is limited, this thesis will concentrate on **Local minimization**.

There are two fundamental strategies for solving the unconstrained minimization problem. One is the line search strategy, and the other is the trust region strategy [18, Section 2.2]. The trust region strategy is based on fitting a model to the data (which is accurate within a certain region) and then minimizing the model.
While many methods based on both these strategies exist, all the methods implemented in this thesis are based on the line search strategy. The line search strategy was chosen over the trust region strategy because of its simplicity and the fact that not all of the line search methods require the calculation of the objective function's gradient [18, Section 2.2]. The line search strategy can be briefly described as follows:

- Start at some initial point $\mathbf{p}_0$ in the multi-dimensional space
- Calculate a new point $\mathbf{p}_{new}$ that has a lower objective function value by proceeding a distance of λ along some direction $\mathbf{n}$ in the multi-dimensional space
- Keep calculating new points until one of the stopping criteria is met.

Multi-dimensional minimization algorithms based on the line search strategy differ in the way that the step size λ and the direction $\mathbf{n}$ are calculated. They are broadly divided into algorithms that use only the objective function values and algorithms that use both the objective function values and the first derivative of the objective function for calculating the direction $\mathbf{n}$. Algorithms that use the derivative of the objective function for calculating $\mathbf{n}$ are called gradient based methods.

For this thesis, five line search strategy methods were implemented:

1. The unit vector direction set (UVDS) method
2. Powell's direction set (PDS) method
3. The steepest descent (SD) method (gradient based)
4. The Fletcher-Reeves-Polak-Ribiere (FRPR) conjugate gradient method (gradient based)
5. The Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton method (gradient based)

The UVDS method and SD method were implemented because they are the simplest non-gradient and gradient based minimization algorithms respectively. The other three methods were implemented because they are popular and refined non-gradient and gradient based minimization algorithms.

2.6 Quantifying the error

To decide whether the error in patient position (given by $\mathbf{T}_{error}$ which is approximated by the value of $\mathbf{T}_{DRR}$ when the objective function is at its minimum) is within acceptable limits the error can be quantified in a number of ways, including the following three:

1. Checking whether each of the translation and rotation components are within a specified limit, e.g. if $\mathbf{T}_{error} = (x_{err}, y_{err}, z_{err}, \theta_{err}, \phi_{err}, \rho_{err})$, then
   - $x_{err} <$ x-limit, and
   - $y_{err} <$ y-limit, and
   - $z_{err} <$ z-limit, and
   - $\theta_{err} < \theta$-limit, and
   - $\phi_{err} < \phi$-limit, and
   - $\rho_{err} < \rho$-limit.

2. Combining the three translation components and checking whether this combination is within a specified translation limit and combining the three rotation components and checking whether this combination is within a specified rotation limit. The combination of the various components can be done in a number of ways. One way is a simple summation of the absolute values:

   - $|x_{err}| + |y_{err}| + |z_{err}| <$ translation-limit and
   - $|\theta_{err}| + |\phi_{err}| + |\rho_{err}| <$ rotation-limit

   and another the Euclidian distance measure:

   - $\sqrt{x_{err}^2 + y_{err}^2 + z_{err}^2} <$ translation-limit and
   - $\sqrt{\theta_{err}^2 + \phi_{err}^2 + \rho_{err}^2} <$ rotation-limit.

3. Combining all six components and checking whether this value is within a specified total error limit. Again, one possible way to combine the six components is to do a simple summation of the absolute values:

   - $|x_{err}| + |y_{err}| + |z_{err}| + |\theta_{err}| + |\phi_{err}| + |\rho_{err}| <$ total-limit

   or the Euclidian distance measure can be used:

   - $\sqrt{x_{err}^2 + y_{err}^2 + z_{err}^2 + \theta_{err}^2 + \phi_{err}^2 + \rho_{err}^2} <$ total-limit.

   Since the translation and rotation parameters have different units (mm and degrees), combining them into one measure can introduce bias.

2.7 Performance Measurement

As stated in Section 1.2, the image registration process must be such that the verification system that uses it, is fast, accurate, robust and require as little human intervention as possible. To know at the end of the day if these objectives have been achieved, some way of measuring them is required.

2.7.1 Measuring the speed of registration

The speed of the image registration process is measured by recording the total time in seconds it takes for the minimization algorithm to find the minimum value of the objective function. This time depends on

- **The time it takes for one objective function evaluation,** which depends on the time it takes to generate a DRR and the time it takes M to measure how well the two images match. This time is highly dependent on the software algorithms used for both these tasks, as well as the hardware (e.g. the processor speed, amount of main memory and operating system, see Appendix A) that the software runs on.
- **The number of objective function evaluations required by the minimization algorithm,** which depends only on the minimization algorithm and is therefore hardware independent.

2.7.2 Measuring the accuracy of registration

The accuracy of the image registration process is expressed by the difference between the transformation $\mathbf{T}_{DRR}$ (at the objective function minimum) and the error transformation $\mathbf{T}_{error}$. This difference can be quantified in a number of ways. The one that will be used in this thesis is the Euclidian distance measure:

$$\text{Total error} = \sqrt{(\mathbf{T}_{error} - \mathbf{T}_{DRR})^2}$$

The bias factor that results from the combination of parameters of different units (as mentioned in Section 2.6) is ignored in favour of having a simplistic accuracy measure.

Since $\mathbf{T}_{error}$ is not usually known (it is exactly what needs to be estimated), experiments where $\mathbf{T}_{error}$ is known must be used to calculate the accuracy of the image registration process.

2.7.3 Measuring the robustness of registration

The robustness of the image registration process is measured by the frequency of misregistrations. A misregistration is defined as a case where the image registration process is not able to find an estimate of $\mathbf{T}_{error}$ that falls within an acceptable limit, i.e. total error > accuracy-limit.
If misregistrations occur often, the image registration process is not very robust. To be able to identify a misregistration, it must be possible to calculate total error. Hence the robustness of the image registration process can only be calculated in experiments where $\mathbf{T}_{error}$ is known.

2.7.4 Measuring the amount of human intervention

Human intervention may be needed in some parts of the image registration process. When using feature based image registration, user input in the form of feature selection may be required. Some minimization algorithms may also require guidance by a user. Such forms of human intervention are not desirable in the verification system that is developed.
Any form of user input that is needed by the image registration process should be identified. Any user input should be classified according to:

- The time required for the user to complete the task
- The skill required from the user to complete the task
- The effect that inadequate user input would have on the image registration process

The overall amount of human intervention and the effect that it has on the image registration process can be measured in this way. It should also be noted whether the human intervention needed can be done beforehand, in which case it would not lead to an increase in the time required by the verification process.

Chapter 3

DRR generation

This chapter discusses the details of the method used for generating the DRRs used in this thesis. The aim of this thesis is not to develop a new sophisticated DRR algorithm, but rather to implement an existing algorithm. The existing algorithm is then specialized to generate DRRs that would resemble the PRs acquired with the DPRS. This will allow the image registration process to perform optimally.

Since the ray-tracing method is the traditional method for generating DRRs, it was the method chosen to be implemented. Section 3.1 describes the ray-tracing algorithm. As stated in the Section 2.2.3, the most prominent physical radiographic processes are modelled to create realistic DRRs. Section 3.2 introduces the X-ray attenuation calibration curve (XACC) and Section 3.3 introduces the beam hardening correction curve (BHCC).
The interpolation of CT data and DRRs are explained in Section 3.4. In Section 3.5 the modulated transfer function (MTF) is introduced as a method for measuring the image quality of DRRs. This method is used to measure the influence that the interpolation techniques discussed in Section 3.4 have on the DRR quality.
Section 3.6 introduces various miscellaneous concepts related to DRR generation. A broad overview of some of the other available DRR generation methods is given in Section 3.7.

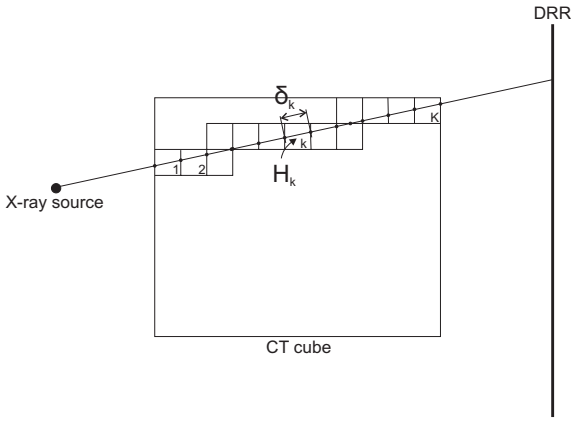


Figure 3.1: *2D representation of radiological path of a ray through the CT cube. The CT number of and the distance travelled through the k'th voxel are also shown.*

3.1 The ray-tracing algorithm

The ray-tracing algorithm is used to calculate the radiological path-length of a ray through the CT cube [13]. For each pixel on the DRR the radiological path-length is calculated from the coordinate that represents the X-ray source to the coordinate that represents that specific pixel. To calculate the radiological path-length of a ray, the ray-tracing algorithm needs to determine which voxels of the CT cube the ray intersects, from the point where it enters the CT cube to the point where it exits. The algorithm also needs to calculate the distance that the ray travelled through each voxel. A 2D representation of a ray travelling through the voxel space is shown in Figure 3.1. The parts of the ray that are outside of the CT cube and their contribution to the radiological path-length are addressed in Section 3.6.1.

The radiological path-length D may be written as the sum:

$$D = \sum_{k=1}^{K} \delta_k C\left(H_k\right), \tag{3.1}$$

where δ_k is the distance travelled through voxel k and H_k is the Hounsfield number of voxel k. C is a function that maps Hounsfield numbers to X-ray attenuation coefficients and is introduced in the following section. K is the

total number of voxels intersected by the ray.

The CT data is stored as a 3D array, and the CT number of each voxel is obtained by indexing the CT data:

$$H_k = H\left(p, q, r\right).$$

For a DRR consisting of 1024×1024 pixels, the radiological path-length needs to be calculated for each of the 1 048 576 pixels. A typical CT data set might consist of 50 slices, with each slice consisting of 256×256 pixels. This means a ray could intersect hundreds of voxels. For each intersected voxel of each ray, the indices p, q and r need to be calculated. The coordinates of the points where the ray enters and exits each voxel also need to be calculated so that the distance inside the voxel can be calculated. Thus a naive approach will have high computational cost.

Siddon [13] suggested an algorithm that considers voxels as the intersection volumes of orthogonal sets of equally spaced parallel planes, instead of independent elements. Instead of calculating the entrance and exit points of each voxel, the algorithm calculates the location where the ray crosses these sets of parallel planes. Since the planes are equally spaced apart, only the location of the first crossing needs to be calculated, the others follow by recursively adding a constant.

Pectora roborant cultus recti

While Siddon provides a fast method for calculating the distance of a ray in a particular voxel, the calculation of the indices of that voxel is still very slow. According to [13] 42% of the total time of the algorithm is spent calculating voxel indices.

Jacobs [19] presented an improvement on Siddon's algorithm called the incremental ray-tracing algorithm. Instead of calculating the indices for each voxel from scratch, the incremental algorithm follows the ray as it crosses the orthogonal sets of parallel planes. Since the ray travels from voxel to voxel, the indices can only be incremented or decremented by one – depending on which plane is crossed. Jacobs claims an improvement factor of 7.5 over the original ray-tracing method.

Jacob's improvement on Siddon's algorithm was implemented for generating the DRRs used in this thesis.

3.2 The X-ray attenuation calibration curve (XACC)

The XACC maps the CT numbers contained in the CT cube's voxels to their corresponding X-ray attenuation coefficients. As will be seen shortly, the X-ray attenuation coefficients depend on the particular X-ray source used.
If the CT numbers are mapped to the X-ray attenuation coefficients that correspond to the DPRS's X-ray tube that is used to acquire the PR's, the radiological path-lengths of the rays that is calculated with (3.1) would be better approximations of the actual radiological path-lengths detected with the DPRS [20]. Therefore, the DRRs would approximate the PRs better.

The calibration curve for a specific CT scanner and X-ray tube pair is constructed by calculating the CT numbers (corresponding to the specific scanner) and X-ray attenuation coefficients (corresponding to the specific X-ray tube) for a large number of tissues.

The CT-numbers of the tissues are calculated using the model presented in [21]. The parameters of the model are fixed by using the measured CT numbers for a small set of tissue substitute materials. The model can then be used to calculate the CT number of any tissue for the specific CT scanner.

The X-ray attenuation coefficients of the tissues are calculated by integrating over the X-ray spectrum:

$$\bar{\mu}_i = \frac{\int \mu_i(E)\,\Omega(E)\,dE}{\int \Omega(E)\,dE}, \tag{3.2}$$

where $\mu_i(E)$ is the attenuation coefficient of tissue i at energy E, which is calculated as in [22], and $\Omega(E)$ is the value of the emitted spectrum at energy E as calculated in [23]. The X-ray spectrum depends on the parameters of the X-ray tube. These parameters include the anode angle, target composition, tube voltage and various filtration parameters. In Figure 3.2 the X-ray spectrum of the DPRS's X-ray tube (for the parameters shown

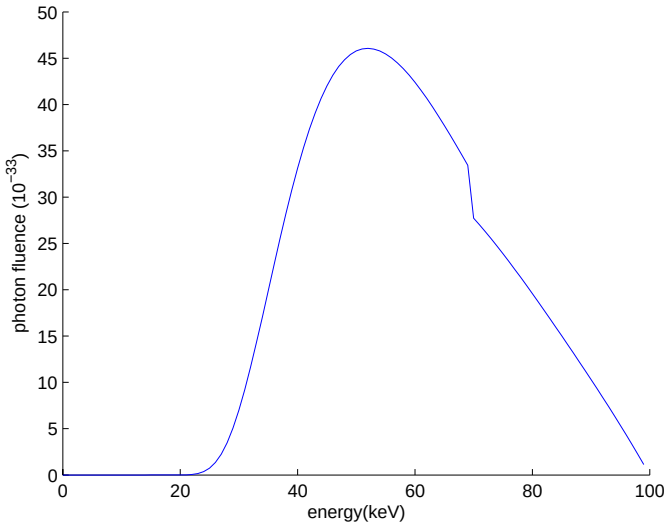


Figure 3.2: *An X-ray spectrum of the DPRS (Tube voltage: 100 kV).*

in Table 3.1) is shown.

Table 3.1: *X-ray tube parameters.*

Rhenium-Tungsten target composition	$5\% - 95\%$
Anode Angle	12°
Tube Voltage	100 kV
Nominal Aluminium Filtration	2.5mm
Oil Filtration	1.5mm

Tissues with low, middle and high CT numbers are used to construct the XACC, with a higher number of tissues from the important soft-tissue region. The XACC is constructed by interpolating these values. An example of such a curve is shown in Figure 3.3. The XACC consists of three different regions [21]. The first region is from Hounsfield numbers -1000 to -200 and represents the air-lung-adipose element mixtures. The second region is from -200 to 200 and represents the soft-tissues. The third region contains the Hounsfield numbers above 200 and represents the harder bone-like materials. The gradient of the XACC is very different in these three regions. Figure 3.4 shows an enlargement of the three different regions.

Figure 3.5 shows two DRRs, the one on the left created without using the

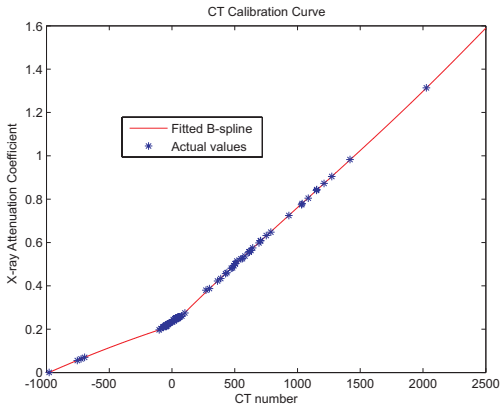


Figure 3.3: *Example of a XACC.*

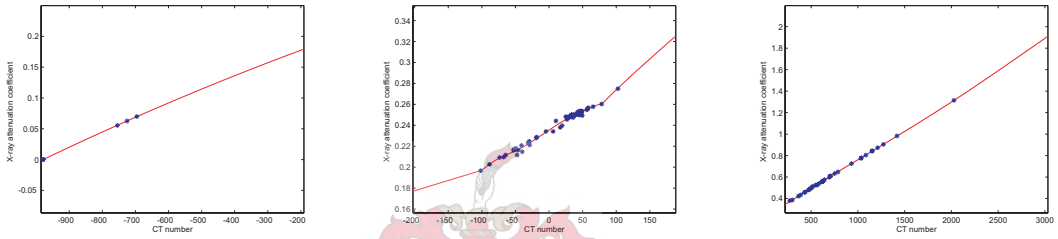


Figure 3.4: *Low, mid and high range CT number mappings*

calibration curve and the one on the right created with the calibration curve. Visually, the DRR generated by using the calibration curve appear to have better image contrast. Unfortunately no real PRs were available to visually compare these images with. In Section 4.2.2 the influence that using the XACC when generating DRRs has on the image registration process is measured.

The use of an XACC to generate DRRs that better resemble specific PRs, is also found in [24], where they created DRRs to resemble mega-voltage radiographs.

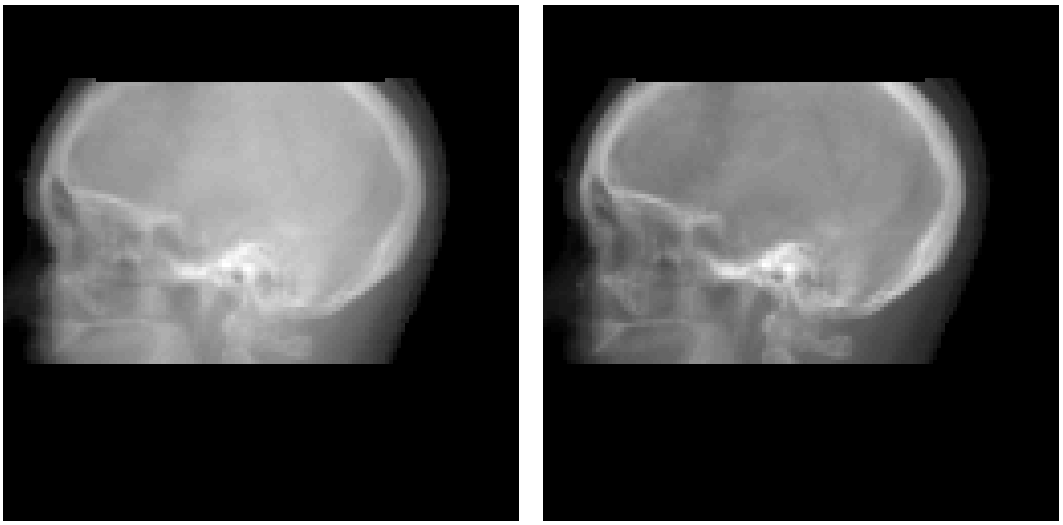


Figure 3.5: *DRR without XACC (left) and DRR with XACC (right).*

3.3 The beam hardening correction curve (BHCC)

In the calculation of the XACC with (3.2) it is assumed that the energy spectrum of the X-ray beam remains constant as it travels through the CT volume. This assumption is in conflict with the physical phenomenon called beam hardening [25]. Because the attenuation by a volume of tissue at a fixed point is greater for X-rays of lower energy, the spectrum of the beam changes as it travels through the CT volume. The spectrum becomes more rapidly depleted of lower energy (or "soft") X-rays compared to the higher energy (or "hard") X-rays and therefore hardens.

It is difficult to take this beam hardening into account during the ray-tracing process, as a new spectrum would need to be calculated continuously, which means that (3.2) needs to be recalculated as well. All these calculations would be impossible to incorporate into the ray-tracing process without having a profound impact on the computational cost.

It is, however, possible to approximate the effect of beam hardening by calculating the effect that beam hardening has on a beam travelling through a volume of water [20]. This is because the attenuation of X-rays by a patient can be roughly approximated by the attenuation of X-rays by a volume of water.

The volume of water is divided into a specified number of layers of the same thickness (δ_x). As the beam travels through the volume of water the spectrum is changed by beam hardening. The spectrum after each layer of water is given by

$$\Omega_n(E) = \Omega_{n-1}(E)\, e^{-\mu_w(E) n \delta_x} \tag{3.3}$$

where Ω_0 is the original emitted spectrum before entering the volume of water and Ω_n is the emitted spectrum after n sections of water. This gives a total distance of water travelled through of $n\delta_x$, where n ranges from 1 to the total number of sections N.

The attenuation coefficient of water at depth $n\delta_x$ with beam hardening accounted for is

$$\tilde{\mu}_n^w = \frac{\int \mu_w(E)\, \Omega_n(E)\, dE}{\int \Omega_n(E)\, dE}. \tag{3.4}$$

The radiological path length is defined as

$$l_n = \sum_{n=0}^{N} \tilde{\mu}_n^w \delta_x. \tag{3.5}$$

A beam hardening correction curve (BHCC) can now be constructed:

$$K(l_n) = \frac{\tilde{\mu}_n^w}{\tilde{\mu}_0^w} \tag{3.6}$$

An example of such a BHCC is shown in Figure 3.6. The summation in (3.1) can now be altered to compensate for beam hardening:

$$D = \sum_{k=1}^{K} \delta_k C(H_k)\, K(d_{k-1}) \tag{3.7}$$

where d_k is the accumulated radiological path length up to the k$^{\text{th}}$ voxel. The index into the BHCC can be computed recursively:

$$d_0 = 0 \tag{3.8}$$

$$d_1 = \delta_1 C(H_1)\, K(\delta_1) \tag{3.9}$$

$$d_k = d_1 + \sum_{m=2}^{k} \delta_m C(H_m)\, K(d_{m-1}) \qquad k \geq 2 \tag{3.10}$$

Note that $D = d_K$.

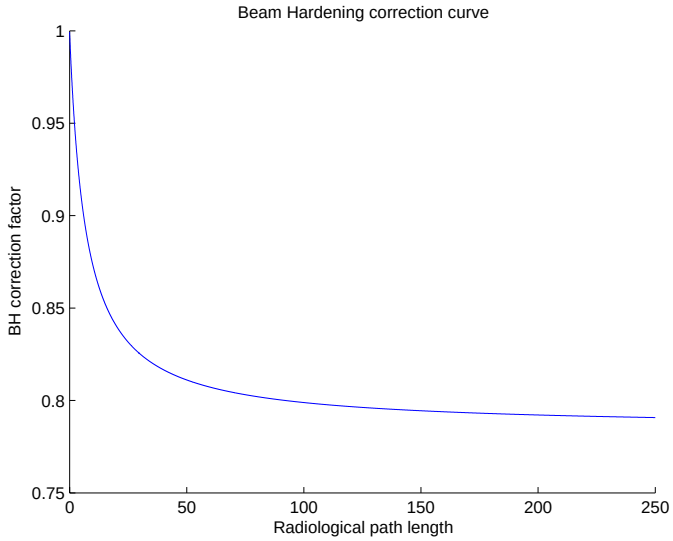


Figure 3.6: *Example of a BHCC.*

Figure 3.7 shows two images. The image on the left is the DRR created without using the BHCC and the image on the right shows the DRR created by incorporating the BHCC.
It should be noted that it is possible to use the XACC without using the BHCC (by setting $K(d) = 1$ for all d in (3.7)), but it is not possible to use the BHCC without using the XACC as well. This is because the XACC is used in calculating the index d_k into the BHCC in (3.10).

By incorporating modelling for beam hardening effects by use of the BHCC in the ray-tracing process, the generated DRRs are a better approximation of the PRs created by the DPRS system.

3.4 Interpolation of the CT data and DRRs

3.4.1 Interpolation of CT data

As stated in Section 2.2.3, the speed of the ray-tracing algorithm depends directly on the number of voxels that each ray intersects as it travels through the CT cube. The number of voxels that a CT cube consists of is inversely

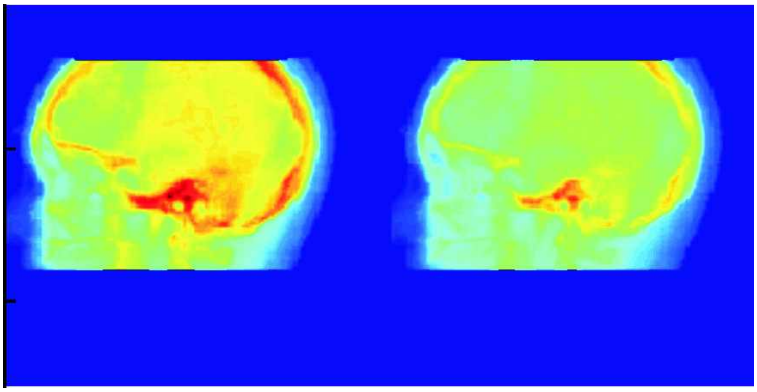


Figure 3.7: *Effect of beam hardening. The figure on the left is the DRR without beam hardening and the one on the right is the DRR with beam hardening.*

proportional to the voxel size. Any patient volume can be represented by a CT cube with larger voxels. To generate a DRR using a CT cube with larger voxels will be faster than generating it from a CT cube representing the same volume with smaller voxels. Generating DRRs from CT data with larger voxels however has the disadvantage that it results in a coarser image that will not resemble the PR as well.

The voxel sizes of a CT cube are tied to the parameters of the CT scanner. Each CT image slice consists of a specified number of pixels (e.g. 256×256). Each CT image slice is also of a specified thickness, and CT image slices are a specified distance apart from each other. These parameters are set bearing in mind the size of the patient volume that must be imaged, as well as the radiation dose that is afflicted on the patient.

After the patient has been scanned, the CT data can be altered by artificially changing the voxel sizes.

Up-sampling

If a patient has been scanned with a large slice thickness and separation, the DRRs created will suffer from slicing artifacts as shown in Figure 3.8(a) where it looks like the image consists of parallel slices. However, the voxel

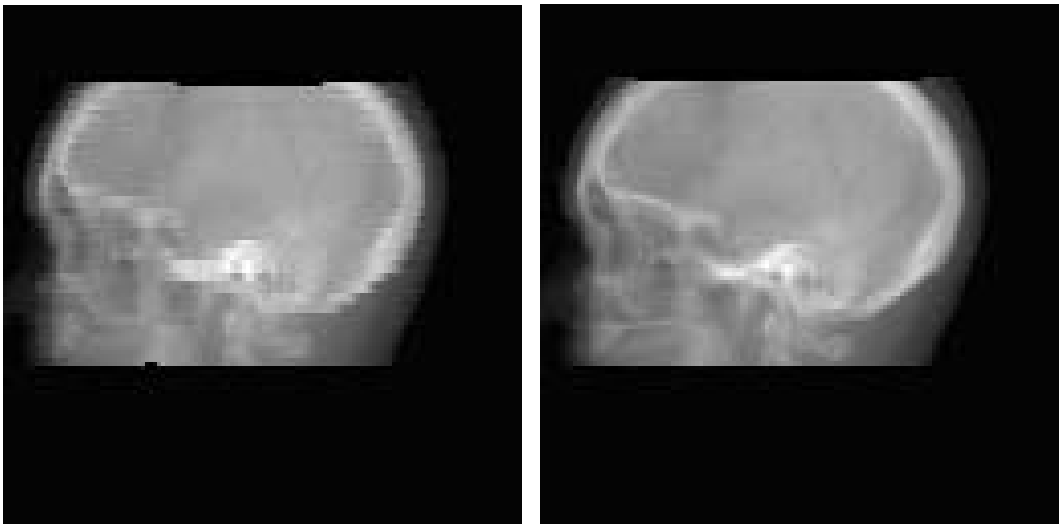


Figure 3.8: *Effect of CT slice thickness on DRR. The DRRs in (a) and (b) have slice thicknesses of 5mm and 1mm respectively*

sizes might be decreased to gain smoother DRR images (Figure 3.8(b)). This process is called up-sampling and is achieved by interpolating the original CT data. The CT data can be interpolated in many different ways (Figure 3.9):

- Interpolating in one dimension across the slices. This kind of interpolation will result in a smaller slice thickness and separation.
- Interpolating each CT slice image. This is a two dimensional interpolation and will result in CT slice images that consists of smaller pixels.
- Three dimensional interpolation of the CT data as a whole. This will yield a CT cube with voxels that are smaller in all three dimensions.

To interpolate a one dimensional function, function values need to be assigned to intermediate positions. The fastest method to assign these values, called nearest neighbour interpolation, is to give the new positions the same function value as their nearest neighbours. A better method is to fit a function through the original values, and to then sample this function at the

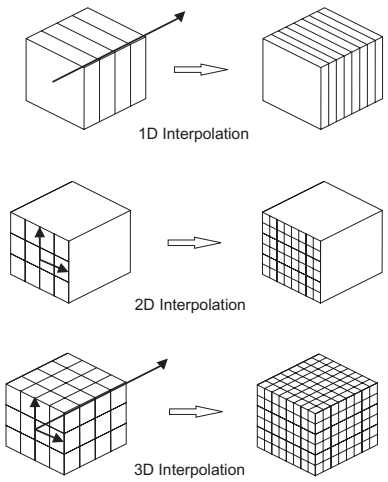


Figure 3.9: *Up-sampling of CT data*

required positions. Linear interpolation and cubic interpolation are such methods, and differ in the type of function that is fitted to the values and the number of points used to fit the function. Linear interpolation fits a polynomial function of the first degree to every two adjacent points, while cubic interpolation fits a polynomial function of the third degree to the function values.

Since cubic interpolation uses more points to fit the function, it is more expensive than linear interpolation but produces a smoother fit [26, Chapter 6]. It should be noted that higher order interpolation can produce over-smoothing effects, while very high order interpolation increases the risk of Runge's phenomenon. Figure 3.10 illustrates the three different interpolation methods in one dimension.

Two dimensional and three dimensional interpolations are done in the same way as one dimensional interpolation. In two dimensions a surface is fitted to the function values. In three dimensions a piece-wise linear or cubic function is fitted to the function values.

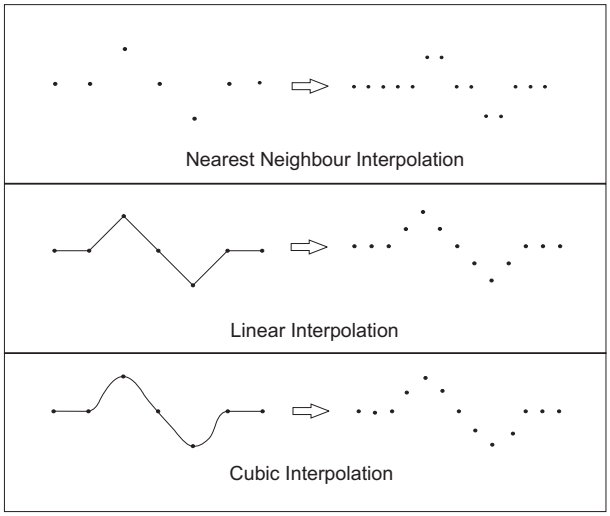


Figure 3.10: *Different interpolation techniques.*

Down-sampling

If the CT cube of a patient consists of very small voxels, the CT data might be down-sampled to increase the voxel sizes and reduce the number of voxels and accordingly the time to generate a DRR.

Down-sampling can also be done by interpolation like up-sampling. For down-sampling the number of positions at which the fitted function will be evaluated are less than the original positions.

Down-sampling can also be achieved by decimation or averaging. During decimation, larger voxels are formed by grouping clusters of smaller voxels (Figure 3.11). Each larger voxel takes on one of the values of the cluster of voxels from which it has been formed. This is equivalent to the nearest neighbour interpolation method.

With down-sampling by averaging, larger voxels are formed from clusters of small voxels in the same way as during the decimation method. With averaging, the new voxel is assigned the average value of the cluster of voxels. This is equivalent to linear interpolation on a suitably chosen grid.

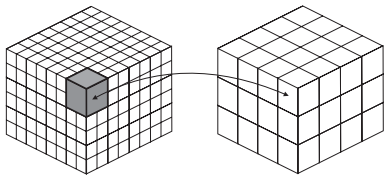


Figure 3.11: *Down-sampling of CT data by grouping clusters of voxels*

3.4.2 Interpolation of DRRs

As stated in Section 2.1.3, the CCD camera of the DPRS can produce PRs consisting of 1024^2-, 512^2- or 256^2 pixels.
The number of pixels that a DRR consists of is equal to the number of rays that must be cast through the CT cube. The fewer pixels a DRR consist of, the less time it would take to generate it.
As will be seen in Section 4.2, when comparing two images, they need to be of the same size. Therefore, when a high quality 1024^2 pixel PR are used during registration, DRRs can either be generated expensively with 1024^2 pixels or be generated much less expensively with 256^2 pixels. The 256^2 pixel DRRs can then be interpolated to 1024^2 pixels to be compared with the same sized PR.
When using a 256^2 pixel PR, DRRs will be generated with 256^2 pixels. Using DRRs consisting of more pixels and then down-sampling them would not make sense, since the more pixels the DRR consists of, the more expensive it is to generate, and the down-sampling process would not increase the image quality of the DRR.

The interpolation of a DRR is exactly the same as the 2D interpolation of a CT image slice as described in the previous section on the up-sampling of CT data. A linear surface is fitted using the nearest four points for linear interpolation, or a cubic surface is fitted using the nearest 16 points for cubic interpolation. These surfaces are then evaluated at the new pixel positions to create a DRR consisting of more pixels.

The effect of DRR interpolation is shown in Figure 3.12. In this case linear

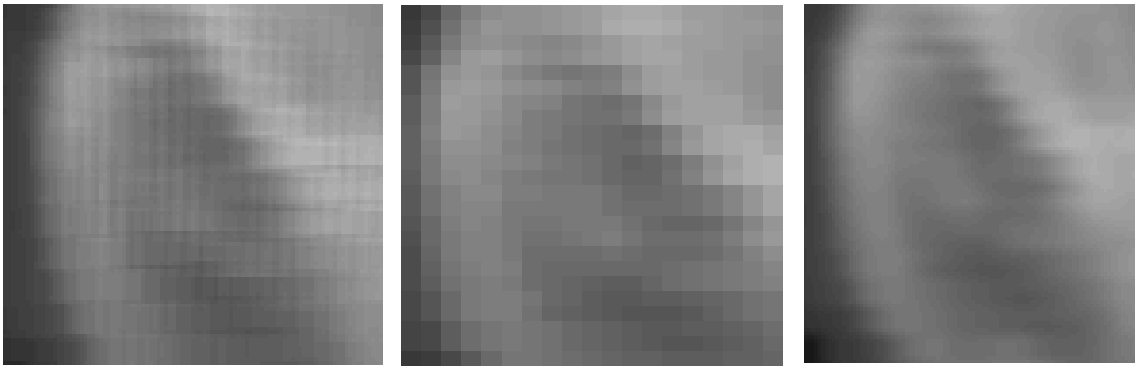


Figure 3.12: *Enlarged pieces of DRRs showing the effect of interpolation*

Table 3.2: *DRR calculation times.*

DRR size	Calculation time (seconds)
1024×1024	14.516
256×256	1.297
256×256 interpolated to 1024×1024	3.735

2D interpolation was used. DRR generation times[1] are given in Table 3.2.

3.5 Image quality

The three basic concepts used to describe the visual quality of an image produced by an imaging system are spatial resolution, contrast and noise [27, Chapter 1].

For this discussion, the ray-tracing algorithm represents the imaging system and the resulting DRR the image. The CT data that is used to create the DRR is part of the imaging system and has a direct influence on the three measures of image quality.

[1]The hardware and software with which these times are calculated are described in Appendix A.

Image resolution

The spatial resolution of an image can be defined as the smallest distance between two objects that can be clearly imaged. This can be calculated by investigating line pairs of ascending frequency and finding the frequency at which the line pairs cannot be discerned anymore. The DRR spatial resolution is determined by the number of pixels that the DRR consists of, as well as the number of pixels that each CT slice consists of and the slice thickness and separation [28]. The size of the CT image slices, the slice thickness and slice separation are set during acquisition, and can afterwards only be altered by up-sampling and down-sampling. The effect (negative or positive) that such alteration of the CT data will have an the DRR spatial resolution needs to be investigated.

Image Contrast

Image contrast is defined as a measure of difference (usually in the image intensity values) between regions in an image, in particular adjacent regions. This measure can be formulated in various ways. One popular form is

$$Contrast = (I_{max} - I_{min})/(I_{max} + I_{min}),$$

where I_{max} is the maximum pixel intensity value in the region and I_{min} the minimum [27, Chapter 1]. The pixel intensity values of a DRR are directly related to the radiological path-length of the rays as calculated in (3.7) and therefore depend on the CT data, the XACC and the BHCC.

Image noise

The noise in a DRR image can be ascribed mainly to the noise that is inherent to the CT data used during DRR generation. The errors induced by the ray-tracing process are limited to numerical round-off errors. These errors occur when calculating the distance that a ray travels through a particular voxel (δ_k in (3.7)) and are very small in comparison to the other terms. To

quantify noise the standard deviation of the image intensity values around their mean value can be calculated for a region in the image that has uniform intensity [27, Chapter 1].

Measuring the image quality

To quantify image quality in terms of these three concepts is very difficult, since they influence each other. Image contrast and spatial resolution are related because image contrast tends to deteriorate as spatial frequency is increased. A measure that relates image contrast to spatial resolution is the modulation transfer function (MTF). The MTF is a popular and widely used measure when it comes to medical imaging. It is not only used for conventional radiography with X-ray films, but also for digital radiography [29],[30],[31]. The quality of DRR imaging systems is difficult to compare against each other, since the quality depends on the CT data used. However, the MTF has been used to measure the effect that the CT data has on the quality of DRRs. In [28] the MTF measured from DRRs is used to evaluate the effect that the number of pixels a CT image slice consists of and slice thickness has on the quality of DRRs. In [32] the MTF of a DRR imaging system is used to determine whether CT data produced with helical CT or axial CT produced DRRs of better quality.

Pectora roborant cultus recti

3.5.1 The modulation transfer function

The various components and stages of an imaging system introduce various forms of unsharpness on the image. This causes the contrast of an image to deteriorate as the frequency increases. To illustrate this, Figure 3.13 shows the input and output of a typical imaging system that introduces unsharpness. The line pair image at the top is the input signal while the line pair image at the bottom is the output of the imaging system. It can be clearly seen that, as the frequency increases, it becomes more difficult to discern the line pairs in the output image as the contrast becomes very low.

An MTF curve gives the relationship between signal or image modulation

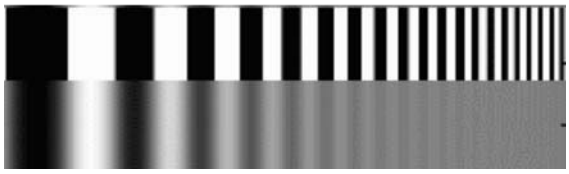


Figure 3.13: *Contrast deteriorates as frequency increases.*

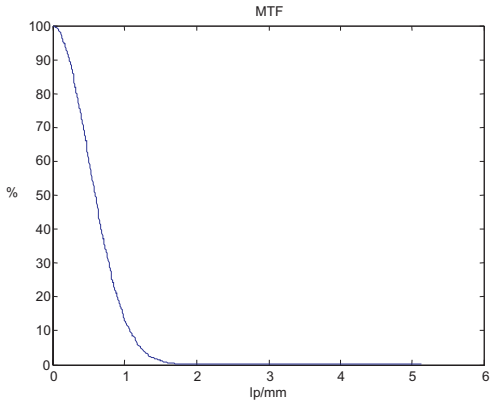


Figure 3.14: *Example of an MTF curve.*

(which constitutes contrast in this case) and frequency. A typical MTF curve of an imaging system that introduces unsharpness is shown in Figure 3.14. The MTF curve of an imaging system can be calculated either by using line pairs, or by using the Fourier Transform.

Pectora roborant cultus recti

Calculating the MTF using line pairs

The MTF can be calculated by generating a DRR of a CT data phantom containing sets of line pairs at designated frequencies and then interpolating these points to form the MTF curve. A DRR of such a phantom (from [33]) is shown in Figure 3.15.

The MTF can be calculated from the DRR generated of the CT data phantom with

$$MTF(f) = \frac{\pi}{\sqrt{2}} \frac{\sqrt{M^2 - N^2}}{|P - Q|} \tag{3.11}$$

where M represents the standard deviation of DRR pixel values within a

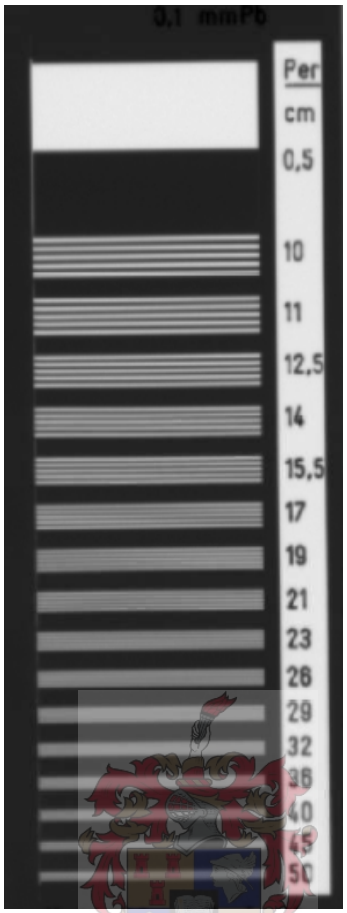


Figure 3.15: *DRR of a CT phantom for MTF calculation.*

set of line pairs and N the standard deviation of an uniform region of the DRR. P represents the mean value of an uniform region that has the highest intensity and Q represents the mean of an uniform region that has the lowest intensity. This formula for calculating the MTF was used in [28] and [32].

Calculating the MTF using the Fourier transform

The Fourier transform can be used to calculate the MTF from the point spread function (PSF), the line spread function (LSF) or the derivative of the edge spread function (ESF) of the imaging system.
The spread functions are characteristic of a particular imaging system and can be obtained by the imaging of a lead plate with a small hole (PSF),

a lead plate with a thin slit (LSF), and the edge of a lead plate (ESF) as illustrated in Figure 3.16. The spread functions are related to each other. The horizontal profile through the PSF and the LSF (obtained with a vertical slit) are equivalent. The ESF are related to the PSF and LSF by the fact that the derivative of the horizontal profile through the ESF is equal to the horizontal profiles of the PSF and the LSF (obtained with a vertical slit) [27, Chapter 6]. Therefore:

$$PF(\text{PSF}) = PF(\text{LSF}) = \frac{d}{dx}PF(\text{ESF})$$

where PF(x) returns the horizontal profile through the given function.
The MTF can be calculated by taking the magnitude of the Fourier transform of any of these three profile functions:

$$\begin{aligned} MTF(f) &= |\mathfrak{F}(PF(\text{PSF}))| \\ &= |\mathfrak{F}(PF(\text{LSF}))| \\ &= \left|\mathfrak{F}(\frac{d}{dx}PF(\text{ESF}))\right| \end{aligned}$$

This is illustrated in Figure 3.16.

3.5.2 Forms of unsharpness

The PSF of an ideal imaging system should be the delta function. This is the case when the X-ray source is infinitely small and the hole in the plate is infinitely small and there are no other factors of unsharpness during the imaging process. The Fourier transform of the delta function is a constant function [34, Appendix G.5]. This means that all frequencies are modulated the same and that there will be no contrast deterioration at high frequencies. In practice, the various components of an imaging system (like the DPRS) all contribute to the unsharpness of the final image.

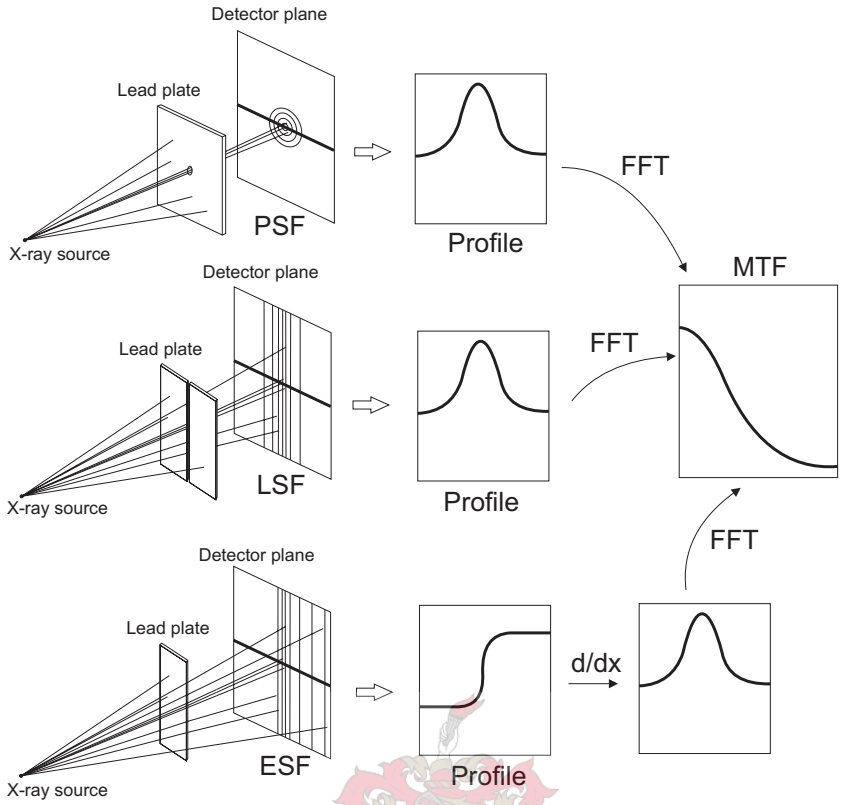


Figure 3.16: *MTF calculation from spread functions using the FFT.*

Geometric unsharpness

Geometric unsharpness is caused by the finite size of the X-ray source's focal spot. In contrast to a point source, a finite source causes a blur around the image of an object called the penumbra. This is illustrated in Figure 3.17. The size of the penumbra is affected by the size of the X-ray source and the object magnification. The object magnification depends on the location of the object in relation to the X-ray source and the imaging plane (Figure 3.18).

Unsharpness due to scattering

As stated in Section 2.1.1, some of the X-rays that travel through the patient are deflected. The effect of this scattering process (shown in Figure 3.19)

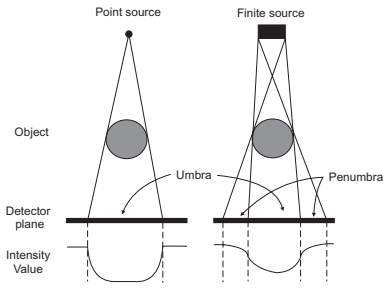


Figure 3.17: *Geometrical unsharpness due to finite source.*

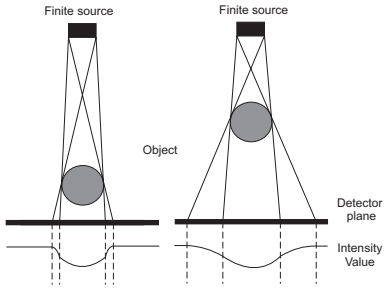


Figure 3.18: *Effect of object magnification on geometrical unsharpness.*

is that the image contains noise and becomes blurry. Grids that absorb scattered rays that do not travel along the direction of the primary beam can be used to reduce the effect of scattering [2, Section 31.1.2].

Pectora roborant cultus recti

Unsharpness due to motion

Motion unsharpness is caused by the movement of the X-ray source, the patient or the detector system during image acquisition. While the X-ray source and detector system are usually held by rigid supporting structures and assumed stationary, patient movement can be a major cause of unsharpness. Patient immobilization techniques are used to minimize patient movement during image acquisition. By using short exposure times during X-ray acquisition, the effect of patient movement can also be limited. However, short exposure times can have a negative impact on image contrast.

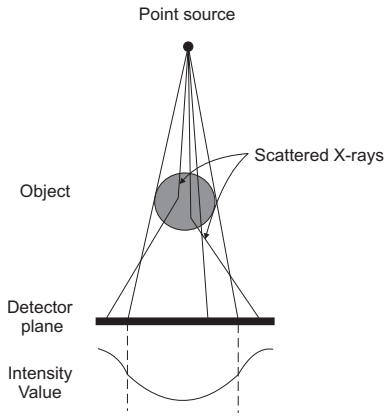


Figure 3.19: *Effect of scattering*

Detector unsharpness

Detector unsharpness is caused by the various components that the particular detector consists of. In the DPRS the diffusion of light in the scintillation screen and the distortion of the reflecting mirrors and CCD camera lens add to the unsharpness of the image [7].

Digitization unsharpness

Small errors are induced by the CCD camera when the light image is converted to electronic signals which are digitized and stored as binary numbers. This is because a finite number of bits are used to store the intensity value of each pixel.

3.5.3 Using the MTF to measure the effect of interpolation on the CT data and DRRs

As stated earlier, it is difficult to measure the absolute quality of the DRR generation process because of its dependency on the CT data. However, the MTF can be used to measure the effect on the quality of the DRRs when the voxel sizes of the CT cube and the pixel sizes of the DRR are artificially altered as described in Section 3.4.

The following is true for the DRR imaging process in terms of the various forms of unsharpness of a digital radiographic imaging system:

- The X-ray source from which each ray starts is a perfect point source. Therefore there is no geometrical unsharpness.
- The ray-tracing algorithm only models the primary fluence and no diverging rays, which eliminates the unsharpness factor due to scattering.
- The object is represented by the CT cube, which is stationary together with the X-ray source and image plane. There is therefore no unsharpness due to movement.
- There is no detector phase during DRR generation. Each pixel is directly calculated with the ray-tracing algorithm.
- Since the ray-tracing is already done in digital form, there is no additional digitization necessary and therefore the errors induced by the ray-tracing process are limited to numerical round-off errors as explained in Section 3.5. There might, however, be errors due to aliasing which is caused by under-sampling. This will be discussed shortly.

Since none of the usual forms of unsharpness occurs during DRR generation (if it is assumed that no under-sampling occurs) and the numerical round-off errors introduced by the ray-tracing process are very small, the imaging process will be very close to ideal – the PSF will be a delta function resulting in a constant MTF. This makes measuring the effect that alteration of the voxel or pixel sizes has on the MTF a futile exercise. To aid the measurement, geometrical unsharpness is introduced by modelling a finite X-ray source.

Modelling of a finite X-ray source

To simulate geometrical unsharpness due to the finite focal spot size (FSS) of the DPRS's X-ray tube when generating a DRR, it is possible to add

blurring to each ray that is cast from the X-ray source to the imaging plane. This results in each pixel of the DRR being blurred in relation to the FSS. This method is computationally very expensive. An alternative method is to convolve the DRR (generated without geometrical unsharpness) with a Gaussian kernel function with a full width at half maximum (FWHM) that depends on the FSS of the DPRS's X-ray tube [33]. The FWHM of the kernel also depends on the DPRS setup (the source-object and object-detector distances), which affects the magnification of the FSS as shown in Figure 3.20. The effective size of the focal spot w depends on the position of the point x in the object. It is assumed that the effective size corresponding to any point x inside the object can be approximated by the effective size corresponding to the treatment iso-center which is on the central beam axis. The effective size can therefore be calculated by

$$w = \frac{OD}{SO} \times FSS, \tag{3.12}$$

where SO and OD are the source-object and object-detector distances respectively and FSS the focal spot size.

A typical setup for the DPRS used by iThemba LABS has values of 2000mm for SO and 500mm for OD. The X-ray tube of this DPRS has two settings for the FSS, either 1.5mm or 0.8mm. This results in effective focal spot sizes of 0.375mm or 0.2mm, which are also the FWHMs of the Gaussian kernels with which the DRRs are to be convolved. The standard deviation of the Gaussian kernel with a FWHM of w is calculated by

$$\sigma = \frac{w}{2.3548}$$

and the DRR with modelling for the finite source incorporated is given by the convolution

$$DRR_{FS} = DRR \otimes G(\sigma)$$

where $G(\sigma)$ is a Gaussian function with standard deviation σ.

By adding geometrical unsharpness to the DRR imaging process, the PSF will no longer be a delta function and the MTF will also no longer be a constant function. This makes it possible to measure the effect that the

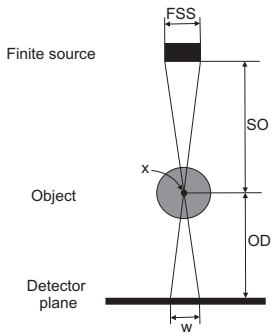


Figure 3.20: *Magnification of finite source.*

alteration of the voxel and pixel sizes has on the MTF.

Calculating the MTF using DRRs

To calculate the MTF of the DRR generation process it was decided to measure the ESF of the DRR generation process. This is done by constructing artificial CT cubes that each contains a thin plate and generating DRRs from them.

Each of these CT cubes is constructed with specific voxel sizes. All the voxels of each CT cube are given the CT number of air, except for the slice in the middle of each CT cube. This middle slice is constructed to contain the edge of the plate. This is achieved by dividing the slice along a vertical line and setting the left side of the slice to the CT number of air and the right side to the highest available CT number. The right side of the middle slice represents the plate. The position of the vertical line where the slice is divided represents the edge of the plate, and can be set to different positions.

The setup for DRR generation is done so that the central beam axis is perpendicular to the slices and enters the CT cube in the center of the first slice. Since the MTF was found to be spatially dependent, the edge of the thin plate is placed at two different positions. The edge is placed either in the center of the middle slice (co-incident with the central beam axis) or on the outskirt (near the edge) of the middle slice (at an angle with the central beam axis). A top view of the setup for DRR generation for both the edge

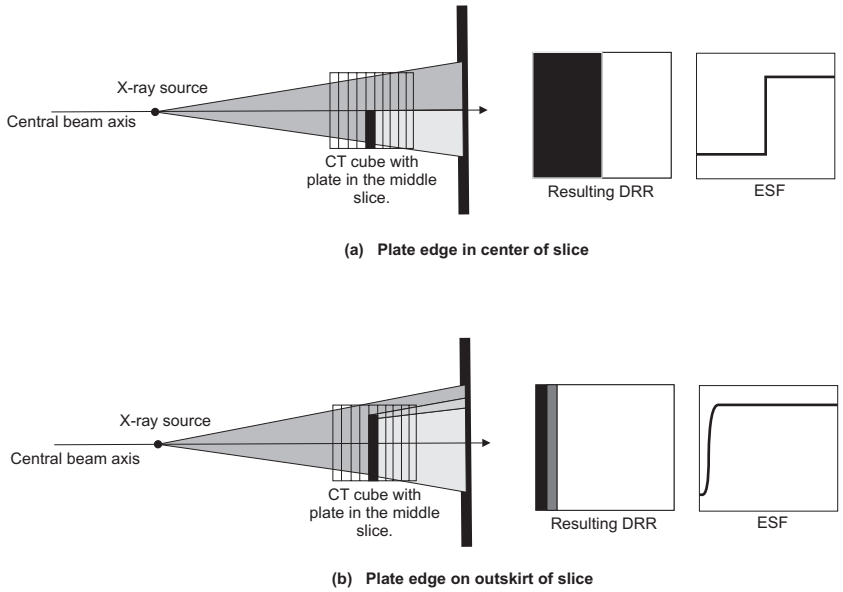


Figure 3.21: *Top view of DRR generation setup for MTF calculation.*

in the center and the edge on the outskirt of the middle slice is shown in Figure 3.21. The figure also shows the resulting DRR for each setup, and that the ESF can be obtained from the horizontal profile through the center of the DRR. As stated in Section 3.5.1 the MTF can be calculated by taking the magnitude of the Fourier transform of the derivative of this ESF.

It should be noted that with the edge on the outskirt of the slice, the angle of the rays causes the edge to be spread over a larger part of the DRR, which results in a different ESF and MTF. This is the reason for the spatial dependency of the MTF. The extent of this spread not only depends on the position of the edge in relation to the central beam axis, but also on the thickness of the plate and therefore the size of the voxels. It should be remembered that when the voxel size of the CT data is changed, the number of voxels is increased or decreased, but that the physical size of the CT cube remains constant.

Sampling frequency

When generating a DRR, the pixels for which the ray-sums are calculated are only a sample of all the possible rays that can be cast from the X-ray source to the image plane. According to the sampling theorem, a signal can be described completely by uniformly spaced samples at a frequency of twice the highest frequency of the original signal [34, Section 2.8]. This frequency is called the Nyquist frequency. When a signal is sampled below the Nyquist frequency a phenomenon called aliasing corrupts the sampled signal [35, Section 2.2.4].

The sampling frequency of the DRR generation process for each spatial dimension is calculated by dividing the number of pixels in that dimension by the physical length of that dimension [35, Section 2.2.4], e.g. if a DRR has a size of 300mm×300mm and consists of 512×512 pixels, the sampling frequency would be

$$f_s = 512/300 = 1.7 \text{ samples/mm}$$

Since all the DRRs generated in this thesis are two dimensional square images, the sampling frequency is the same for both dimensions.

When a DRR is generated from a CT cube that contains line pairs in only one of its slices, and the slices are perpendicular to the central beam axis, then the highest frequency of line pairs are achieved when each alternating voxel has a different value. For CT slices of 512×512 pixels of size 1mm×1mm the highest frequency is 0.5 lp/mm (line-pairs per millimetre), and the Nyquist frequency 1 samples/mm.

When generating DRRs to measure the ESF (which is used to calculate the MTF) as described in the previous section, care should be taken that the sampling frequency of the DRR generation process do not go below the Nyquist frequency of 1 samples/mm to avoid aliasing.

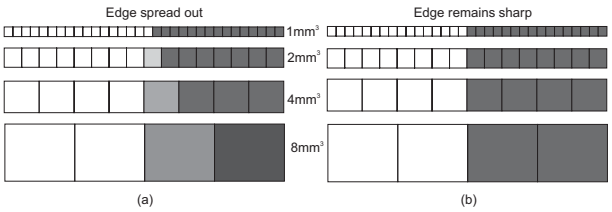


Figure 3.22: *One dimensional view of down-sampling by averaging*

Down-sampling of CT data

To measure the effect that down-sampling of the CT data has on the MTF, a CT cube is constructed in the way described above with 512^3 voxels of size $1mm^3$. CT cubes with 256^3 voxels of size $2mm^3$, 128^3 voxels of size $4mm^3$ and 64^3 voxels of size $8mm^3$ are then created by down-sampling the original CT data. The down-sampling is done using averaging as described in Section 3.4.1, and Figure 3.22 shows a one dimensional illustration of the effect on the edge of the plate. The figure shows how $2mm^3$-, $4mm^3$- and $8mm^3$ cubical voxels are obtained. When the plate ends on an uneven voxel, the edge is spread out over three voxels (Figure 3.22(a)). If the edge falls on a even voxel, the edge remains sharp (Figure 3.22(b)).

Experiments were conducted for the case where the edge of the plate is in the center of the middle slice (like Figure 3.21(a)) and where the edge of the plate is on the outskirt of the middle slice (Figure 3.21(b)). Experiments were also conducted for the case when down-sampling causes the edge to be spread out over multiple voxels and for the case where the edge stays sharp. Thus, four experiments were conducted to investigate the effect of down-sampling of the CT cube has on the image quality of DRRs:

- **Experiment 1**: Plate edge in the center of middle slice – the edge remaining sharp
- **Experiment 2**: Plate edge in the center of middle slice – the edge spread out
- **Experiment 3**: Plate edge on the outskirt of middle slice – the edge remaining sharp

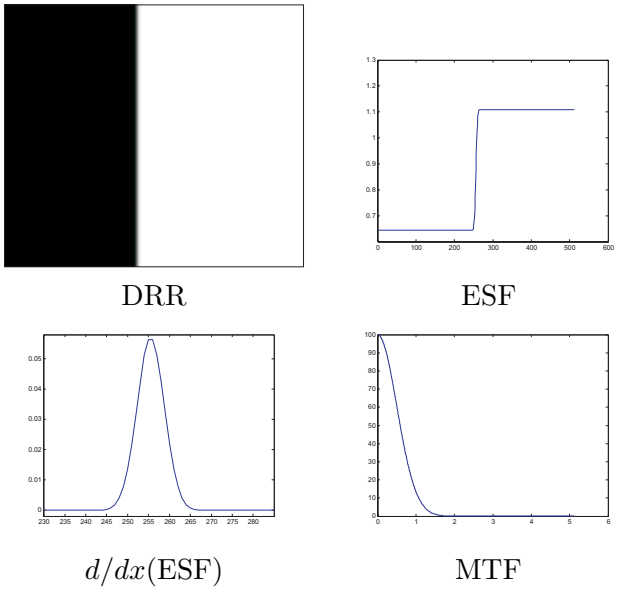


Figure 3.23: *Calculation of the MTF for the CT cube down-sampled to 4mm^3 with the edge positioned to remain sharp.*

- **Experiment 4**: Plate edge on the outskirt of middle slice – the edge spread out

In all the experiments above, modelling for the finite source is added. Since the edge of the plate only comprises a small area of the DRR, DRRs of size 50mm^2 are created and positioned so that the edge falls in the center of the DRR image. The DRRs are created with 512^2 pixels which result in a sampling frequency of 10.24 samples/mm. This sampling frequency is sufficient to avoid any aliasing.

Figure 3.23 shows a typical DRR, the ESF taken from the horizontal profile of the DRR, the derivative of the ESF and the MTF calculated using the Fourier transform. The DRR of this figure is created from the CT cube down-sampled to 4mm^3 voxels with the plate in the center of the middle slice and the plate positioned so as to cause a sharp edge.

Figure 3.24 shows the calculation of the MTF for the same CT cube, but with the edge positioned to cause an edge that is spread out. As can be seen

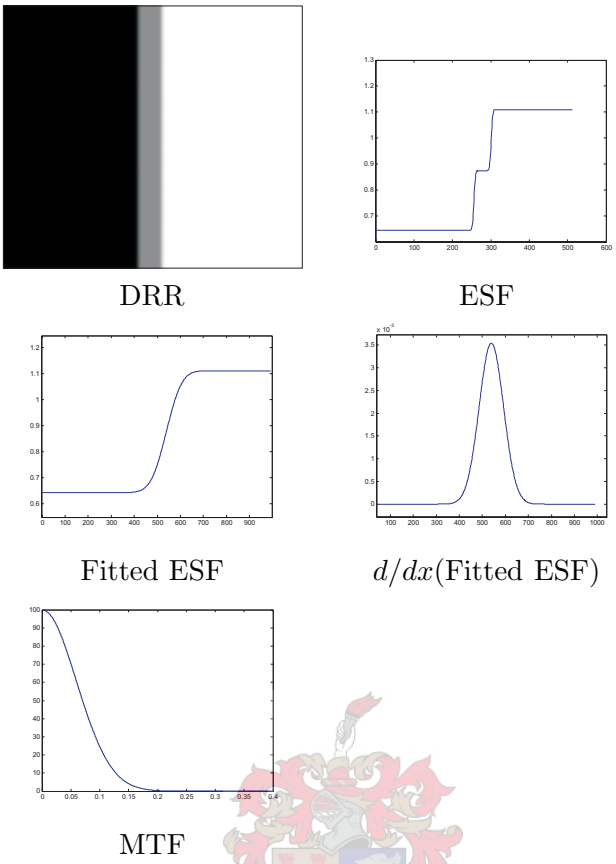


Figure 3.24: *Calculation of the MTF for the CT cube down-sampled to 4mm^3 with the edge spread out.*

the ESF contains a step. These kinds of steps are characteristic of all the experiments with the edge positioned so that it is spread out. Calculating the Fourier transform from functions that contain steps like these does not yield usable results, since they contain ringing due to the steps. This problem is overcome by approximating the ESF with a smooth Gaussian kernel function as shown in the figure. The MTF is then calculated from the fitted ESF.

When the MTF curve of Figure 3.24 is compared with the one in Figure 3.23, it is clear that the MTF values for the spread out edge is much lower than the MTF values for the sharp edge. This is a manifestation of the decline in image quality that the down-sampling incurs.

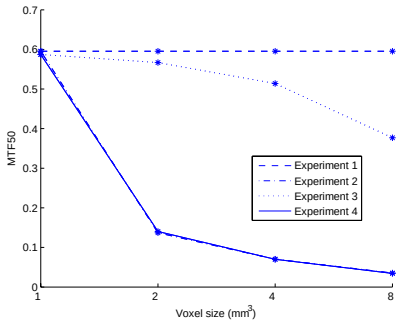


Figure 3.25: *Results of down-sampling experiments*

To be able to compare the results of the experiments conducted, the MTF50 values are taken from the MTF curves of the four experiments. These are simply the frequencies where the MTF curves are at 50% of their maximum value. These MTF values are commonly used to measure the quality of imaging systems, and were also used in [28] to analyse the effect that the CT parameters have on DRR image quality.

Figure 3.25 shows the MTF50 values of the four down-sampling experiments conducted. The MTF50 for Experiment 1 has a constant value of 0.5954 lp/mm for the original $1mm^3$ voxel CT data and the down-sampled $2mm^3$-, $4mm^3$- and $8mm^3$ voxel CT data. This is partly because the edge remains sharp as shown in Figure 3.22(b). It is also partly due to the fact that the increase in voxel size has no effect on the ESF because the edge of the plate is co-incident with the central beam axis (Figure 3.26(a)).
The MTF50 for Experiment 2 shows a sharp decline from the original $1mm^3$ voxel CT data to the down-sampled $2mm^3$ voxel CT data. This is because the edge is spread out for the down-sampled CT data (Figure 3.22(a)). The further decline in the MTF50 value from the $2mm^3$ voxels to the $8mm^3$ voxels is because of the increased voxel size and the fact that the position of the edge moved away from the central beam axis because of the down-sampling (Figure 3.22(a)). The effect that an increase in voxel size has on the spreading of the edge is shown in Figure 3.26(b).

The MTF50 for Experiment 3 shows a gradual but steady decline. Because the edge stays sharp in this experiment, no radical decrease in the MTF50

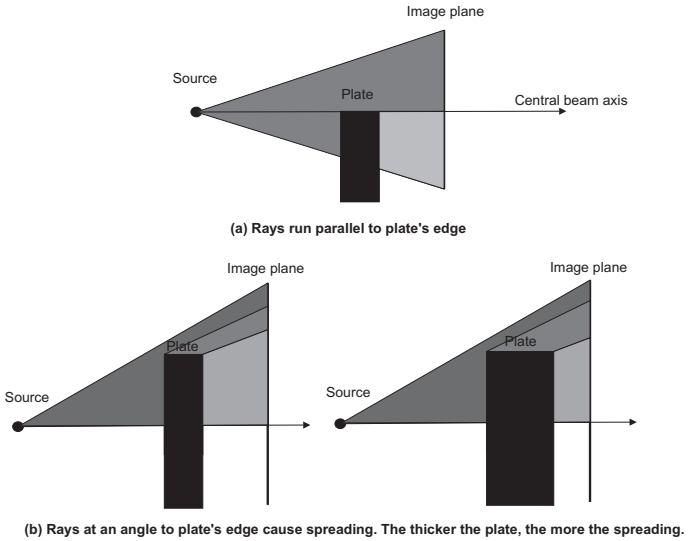


Figure 3.26: *Effect of the position of the plate's edge and the voxel size on the spreading of the edge.*

value can be seen between the $1mm^3$- and $2mm^3$ voxels. It can be seen that the MTF50 value for the original $1mm^3$ voxels is slightly lower for this experiment (where the edge is on the outskirt) than for the edge in the center (Experiment 1). This is because of the spread caused by the angle of the rays with the central beam axis. The further decline in the MTF50 can be solely ascribed to the spreading caused by the increase in voxel size.
The MTF50 for Experiment 4 is very much the same as the MTF50 for Experiment 2. This is because the edge is spread out for both these experiments, which overpowers the effect that moving the plate to the outskirt of the middle slice would have on the spreading.

It is concluded from these experiments that down-sampling of the CT data leads to a loss in image quality. This is with the exception of the case of Experiment 1, which is limited to the position of the DRR that co-incides with the central beam axis. The further away from the central beam axis, the worse is the effect of down-sampling on the image quality. If any edge is spread out it has a severe negative impact on the image quality, and even if the edge stays sharp, the increase in voxel size has a negative impact on the image quality.

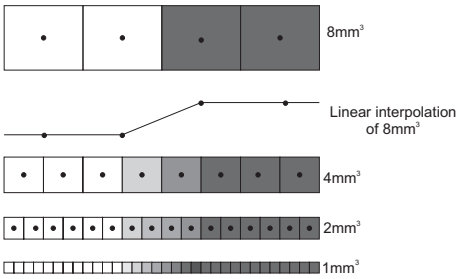


Figure 3.27: *One dimensional view of up-sampling by linear interpolation*

Up-sampling of CT data

To measure the effect that up-sampling of the CT data has on the MTF, a CT cube is constructed in the same way as for the down-sampling experiments but with 64^3 voxels of size $8mm^3$. CT cubes of 128^3 $4mm^3$ voxels, 256^3 $2mm^3$ voxels and 512^3 $8mm^3$ voxels are then created by up-sampling the original CT cube. The up-sampling is done by linear interpolation as described in Section 3.4.1 and Figure 3.27 gives a one dimensional illustration of the effect on the edge of the plate. The figure shows how $4mm^3$-, $2mm^3$- and $1mm^3$ cubical voxels are obtained. As can be seen from this figure, any up-sampling causes the edge to be spread out over a distance that is equal to the width of one of the original $8mm^3$ voxels. The smaller the voxels the CT cube is up-sampled to, the smoother the edge becomes.

Two experiments were conducted to measure the effect of up-sampling on the MTF:

- **Experiment 1**: Plate edge in the center of middle slice
- **Experiment 2**: Plate edge on the outskirt of middle slice

These experiments were conducted in the same way as those for the down-sampling of the CT cube. The setup for Experiment 1 is as shown in Figure 3.21(a) and the setup for Experiment 2 is as shown in Figure 3.21(b).
Modelling for the finite X-ray source is added and the DRRs are once again of size $50mm^2$ with 512^2 pixels and positioned so that the edges fall in the center of the DRRs.

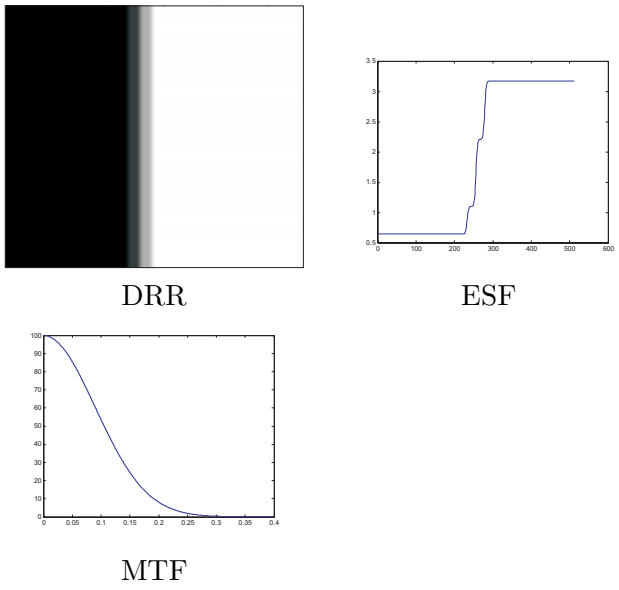


Figure 3.28: *Calculation of the MTF for the CT cube up-sampled to 4mm^3.*

Figure 3.28 shows a DRR created from the CT cube up-sampled to 4mm^3 voxels and the ESF taken from the horizontal profile of the DRR. Two steps are seen on the ESF. They are caused by the two voxels of intermediate value which is the result of the up-sampling process (Figure 3.27). The ESF is once again approximated with a Gaussian kernel function as was done for the down-sampling experiments, and the MTF shown was calculated using the Fourier transform. The DRR was created with the plate in the center of the middle slice.

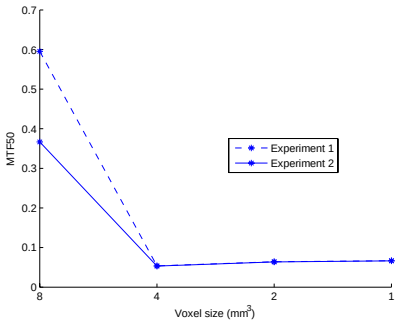


Figure 3.29: *Results of up-sampling experiments*

Figure 3.29 shows the MTF50 values for the two up-sampling experiments conducted. It can be clearly seen that MTF50 value for the original $8mm^3$ voxels with the edge on the outskirt (Experiment 2) is again lower than that for the edge in the center (Experiment 1). This was also noticed for Experiment 2 and 4 of the down-sampling experiments and is caused by the angle between the rays and the plate's edge (Figure 3.26(b)). For the up-sampling experiments the effect is more profound because the original voxels are of a much larger size ($8mm^3$ compared to the down-sampling's original voxel size of $1mm^3$).
The sharp decline in the MTF50 of both experiments for the $4mm^3$ voxel up-sampled CT cube is because of the spreading of the edge as shown in Figure 3.27.
After the initial sharp decline in the MTF50 for $4mm^3$ voxels, both experiments show a slight increase in the MTF50 for $2mm^3$- and $1mm^3$ voxels over $4mm^3$ voxels. This can be ascribed to the thinning of the plate which has the effect explained in Figure 3.26(b).

It is concluded that up-sampling of the CT cube also has a negative impact on the image quality. It should however be noted that the negative impact of up-sampling does not escalate as in the case of down-sampling, e.g. up-sampling to a quarter of the original voxel size is not worse than up-sampling to a half of the original voxel size.
Although up-sampling has a negative impact on image quality in terms of spatial resolution, the improvement that it makes in terms of image smoothness when working with CT data that has thick slices, is very useful (Figure 3.8).

Interpolation of DRRs

When measuring the effect that interpolation of a DRR has on the image quality of that DRR, it is very important that no under-sampling occurs. Incorrect values for the MTF might be recorded if aliasing occurs because of under-sampling. The sampling rate is tied to the DRR generation parameters that specify the physical size of the DRR, and the number of pixels that

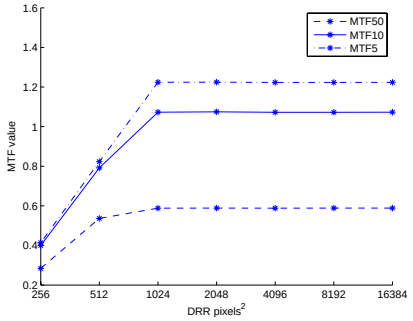


Figure 3.30: *MTF values for DRRs with different amounts of pixels (Experiment with the edge in the center of the middle slice)*

needs to be calculated. Since the DRRs need to be of the same size as the scintillation screen of the DPRS, the physical size is restricted to 300mm^2. To find out how many pixels need to be calculated for a high enough sampling rate, an experiment was conducted that investigates the behaviour of the MTF values for DRRs calculated with different amounts of pixels. The experiment was set-up as in Figure 3.21(a) with the plate edge in the center of the middle slice. The MTF curves for DRRs of 256^2- to 15360^2 pixels were calculated and the MTF50, MTF10 and MTF5 values were recorded.

Figure 3.30 shows the result of the experiment. The MTF values of the DRRs calculated with 1024^2 pixels and above have the correct value, while those with less pixels suffer from incorrect values because of under-sampling. To calculate DRRs with more than 1024^2 pixels are very expensive, therefore the experiment were adjusted to allow more spreading of the edge. This was done by placing the plate's edge on the outskirt of the middle slice as in Figure 3.21(b) and by using a thicker, 20mm plate. This was done to further aid the spreading of the edge as described in Figure 3.26. The MTF values were once again measured for the new experiment setup.

Figure 3.31 shows the result of the new experiment. It can be seen that the MTF values are stable for DRRs calculated with 512^2 pixels and above.

To measure the effect that DRR interpolation has on the image quality of the DRRs as measured by the MTF, three DRRs were calculated with the

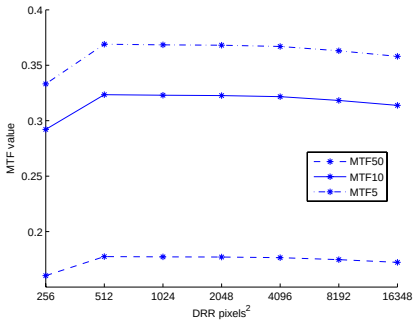


Figure 3.31: *MTF values for DRRs with different amounts of pixels (Experiment with 20mm plate and edge on the outskirt of the middle slice)*

thick plate setup as described above. The DRRs were created with 512^2-, 1024^2- and 2048^2 pixels, and the MTF curves were calculated for each DRR. As gathered from the previous experiment, the MTF50 values of these DRRs were very close to 0.177 lp/mm.
Using the interpolation techniques described in Section 3.4.2, the following DRRs were created:

- the 512^2 pixel DRR were up-sampled to a 1024^2 pixel DRR and a 2048^2 pixel DRR,
- the 1024^2 pixel DRR were down-sampled to a 512^2 pixel DRR and up-sampled to a 2048^2 pixel DRR,
- the 2048^2 pixel DRR were down-sampled to a 1024^2 pixel DRR and a 512^2 pixel DRR.

The MTF curves were calculated for these DRRs and the MTF50 values were recorded. Figure 3.32 and Table 3.3 show the MTF50 values. The figure shows that the MTF50 values remain virtually unchanged. The up-sampling of the 512^2 pixel DRR to 1024^2- and 2048^2 pixels is the only DRR interpolation that had a noticeable but small negative effect on the image quality. The down-sampling of DRRs showed no noticeable negative effect on the MTF values.

Table 3.4 shows a summary of all the interpolation results.

DRR size	**MTF50(lp/mm)**
512	0.17749
1024	0.1772
2048	0.17745
512 up-sampled to 1024	0.17263
512 up-sampled to 2048	0.17286
1024 up-sampled to 2048	0.17608
1024 down-sampled to 512	0.17601
2048 down-sampled to 512	0.17739
2048 down-sampled to 1024	0.17722

Table 3.3: *Effect of DRR interpolation on MTF50 values.*

Type of Interpolation	**Effect on MTF**	**Practical Use**	**Resulting DRR image**
CT data down-sampling	Large negative	Decrease DRR generation time	Coarser
CT data up-sampling	Negative	Reduces slicing effect on DRRs	Smoother
DRR down-sampling	Little effect	Increasing DRR pixel size	Coarser
DRR up-sampling	Small negative	Decreasing DRR pixel size	Smoother

Table 3.4: *Results of Interpolation of CT data and DRRs*

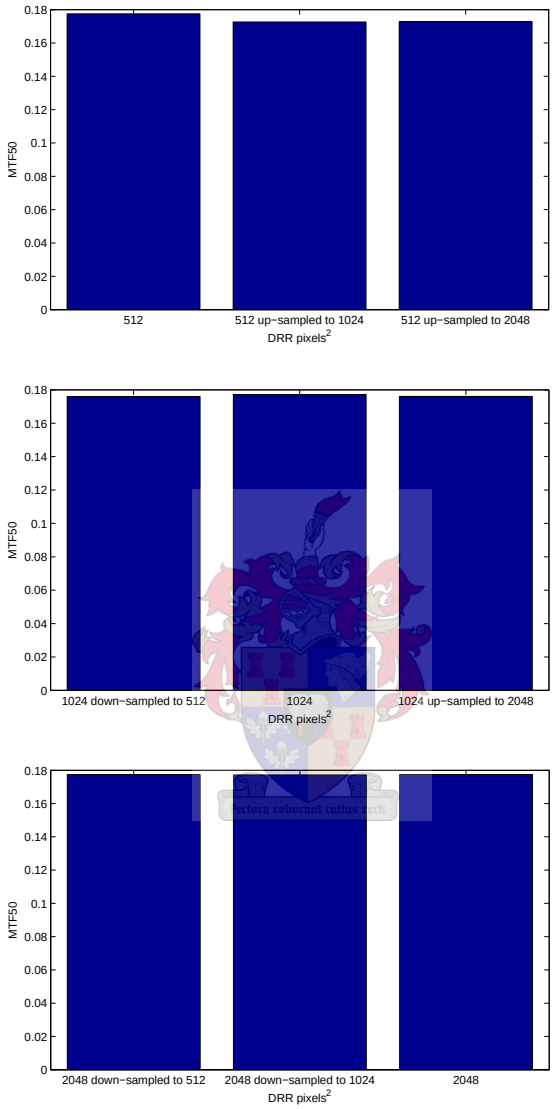


Figure 3.32: *Effect of DRR interpolation on MTF50 values.*

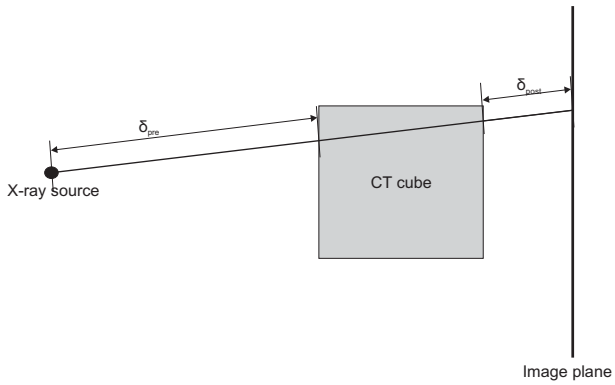


Figure 3.33: *Calculation of the distances of air attenuation to include*

3.6 More on DRR generation

This section introduces various miscellaneous concepts that is important during DRR generation.

3.6.1 Adding attenuation of air

When a radiograph is taken, the X-rays travel through a piece of air before they enter the patient, and through another piece of air between the patient and the detector system. These two pieces of air attenuates the X-rays and although it has a small effect, it is incorrect to ignore it.

To include the attenuation of the X-rays by the pieces of air before and after the patient, the distances between the X-ray source and the CT cube, and the CT cube and the image plane are calculated (Figure 3.33). The radiological path-length D of (3.7) can now be initialized to include the attenuation of the air between the X-ray source and CT cube by setting

$$d_0 = \delta_{pre} C(H_A),$$

where δ_{pre} is the distance between the X-ray source and the CT cube and H_A is the CT number of air.

The attenuation of the piece of air between the CT cube and the image

plane can be included by calculating the total radiological path-length as

$$D = d_K + \delta_{post} C(H_A) K(d_K),$$

where d_K is the attenuation after travelling through all K voxels and δ_{post} is the distance between the CT cube and the image plane.

3.6.2 Using volumes of interest

The CT data of a patient can be accompanied by volume of interest (VOI) data. The VOI data contains the segmentation of a particular volume in the CT cube of the patient and can contain a number of sub-VOIs. Each sub-VOI is represented by a number of contour polygons, each contour corresponding to a specific CT image slice of the CT data.

Usually the patient volume (along with other important volumes such as the optic nerves, spinal cord and target volume) is segmented from the CT cube and available as a VOI. This VOI can be used during DRR generation to exclude any unwanted structures (such as the CT scanner couch) that might be contained in the CT data. This is easily accomplished by generating a mask from the VOI for each CT image slice and using the mask to give the pixels outside the VOI a specific identification CT number. Because the material outside the patient volume should be air, the XACC is adjusted to map the identification CT number to the X-ray attenuation coefficient of air. The pixels outside the VOI is not set directly to the CT number of air because they are used during the check for incomplete CT data as will be discussed in Section 3.6.4.

Care must be taken not to choose the identification CT number equal to a number that can occur in the CT cube. As the CT numbers for patients range from -1000 to 1500 the identification CT number was chosen to be 9999. Figure 3.34 shows how a mask is generated from the contour of the patient VOI and applied to the corresponding CT image slice. This is done for each slice of the CT data. If a particular slice has no corresponding VOI contour polygon, the whole slice is set to 9999.

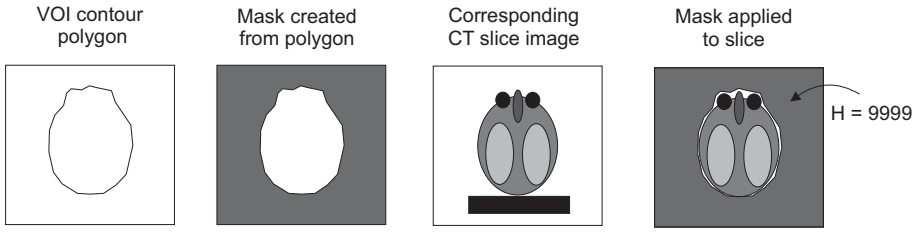


Figure 3.34: *Applying mask from VOI on CT slice*

3.6.3 Using regions of interest

The PR of the patient that is acquired with the DPRS might also contain unwanted structures like patient immobilization devices and parts of the chair or couch used during treatment. These unwanted structures can be removed by using a user supplied region of interest (ROI).
The user can select the patient ROI from the PR using polygon based drawing tools. The DRRs that are to be compared with the PR are then only calculated for the pixels that lie within the given ROI.
In addition to removing unwanted structures, using a ROI also reduces DRR generation time. This is because large areas of a PR might not contain interesting information that will aid the comparison with the DRR, and therefore it is not necessary to calculate the pixels of the DRR for these regions.

Figure 3.35(a) shows a PR taken from a skull phantom. The wooden plate and stub used to support the skull can be clearly seen in the PR. In (b) a polygon is drawn around the skull on the PR to create a ROI that excludes any unwanted structures. A DRR generated without using the ROI is shown in (c). The wooden plate that supports the skull can be seen in the DRR. Note that the CT data used to generate the DRR was not accompanied by VOI data. The wooden plate and stub (although artificial) are the kinds of unwanted structures that can be removed from the CT data by using VOIs as described in the previous section. In (d) the DRR generated by using the ROI is shown.
By only calculating the pixels for the DRR that resides inside the ROI, the number of rays that the radiological path-length needs to be calculated for is reduced from 1048576 to 486759. The time needed to calculate the DRR

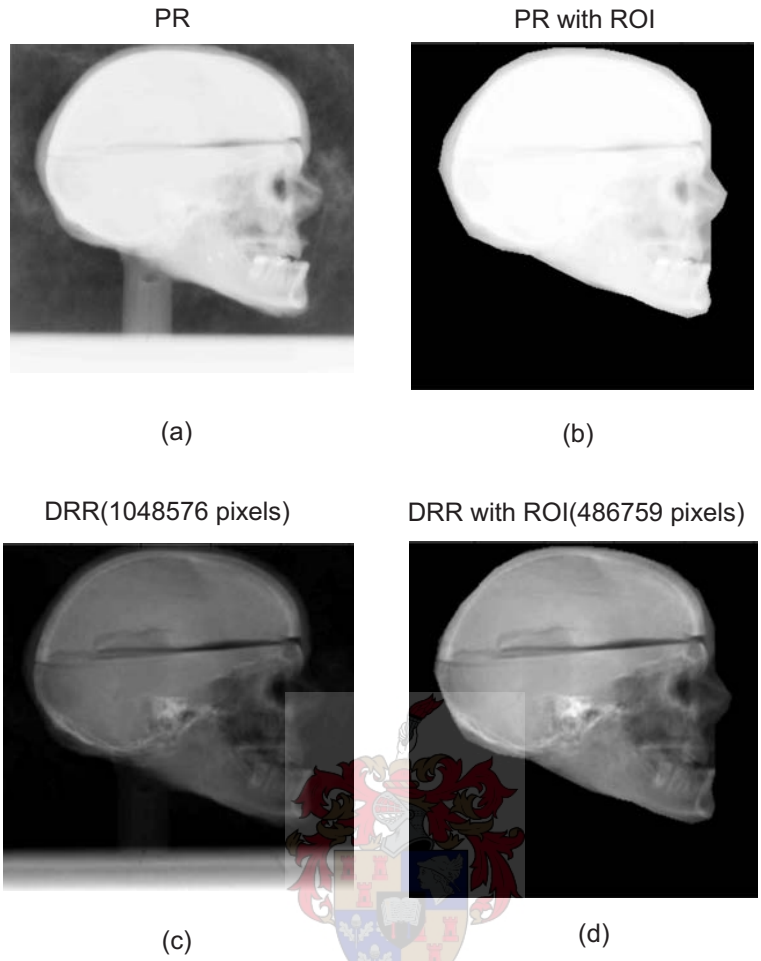


Figure 3.35: *DRR generation using a user supplied ROI*

is hence reduced by 53.58% by using the ROI.

3.6.4 Check for incomplete CT data

Care must be taken when generating DRRs from certain views of the CT data. It must be remembered that the CT data represents only the imaged part of the patient. It can therefore happen that rays exit the CT cube at a point in space where there is supposed to be patient anatomy, but the patient anatomy was not included in the CT scan. This occurrence is illustrated in Figure 3.36. The CT data is therefore incomplete for such particular rays. The radiological path-length for these rays are incorrect,

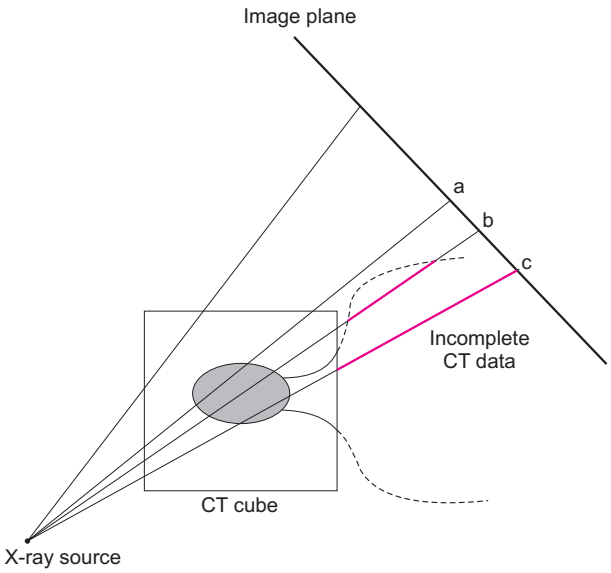


Figure 3.36: *Incomplete CT data*

and the corresponding DRR pixels will not correlate with the pixels on the PR at the same position. They should therefore not be included in any comparisons.

During DRR generation, the patient VOI is used to check for incomplete CT data. An output mask is generated which indicates which DRR pixels are valid. If a pixel is included in the output mask, the corresponding DRR pixel is valid. For a pixel to be valid, the corresponding ray must adhere to the following two rules:

1. The first voxel that the ray travels through in the CT cube must be outside the patient volume (it should have the CT number 9999)

2. The last voxel that the ray travels through in the CT cube must also be outside the patient volume (it should also have the CT number 9999).

From Figure 3.36 it can be seen that ray (a) is valid and adheres to the two rules. Ray (c) is invalid and is identified as being invalid because it does not satisfy the second rule. Ray (b) is also invalid but there is not enough information available to identify it as being invalid.

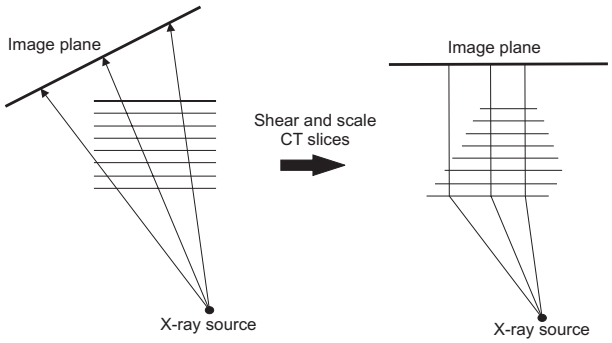


Figure 3.37: *Shearing and scaling of CT slices*

3.7 Other DRR generation methods

As stated in the beginning of this chapter, the ray-tracing method is the traditional method for generating DRRs. Because of the long computation times required by ray-tracing, many other methods for generating DRRs less expensively have been developed.

In [36] Larose presents a method that uses an intermediate data representation called the **Transgraph**. The basic idea is to pre-calculate a set of DRRs from certain viewpoints, and then arbitrary DRRs can be created by interpolating these existing DRRs. The accuracy of this method is however limited to the density of the viewpoints for which DRRs are pre-calculated.

Lacroute et.al. [37] introduced a method called **shear warp factorization**. This method transforms the CT data into a sheared object space so that all the rays runs parallel to each other as illustrated in Figure 3.37. An intermediate image is generated by summing the voxels along the projection axis. The DRR is then generated by warping the intermediate image onto the image plane.

Voxel-projection or **splatting** based DRR generation methods calculates the DRR by accumulating the image plane projections for each voxel of the CT cube. The projection of a voxel on the image plane is called its footprint (Figure 3.38). Many different footprint approximating functions have been suggested, such as cone, Gaussian, sync and bilinear functions.

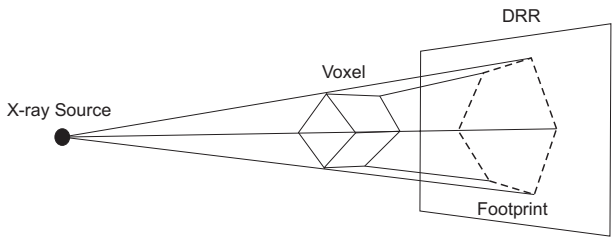


Figure 3.38: *Voxel footprint*

In [38] smooth basis functions are used to approximate the footprint.

In [39] and [11] **light fields** are used to generate DRRs. In this method, rays are parameterized by their intersection with two arbitrary planes. The shape between these two planes is called a light slab which contains infinitely many rays. Any image can be created by determining which rays are involved and associating them with their corresponding pixels. The light slab is constructed by pre-calculating a large number of rays. Missing rays are then calculated by interpolation.

In [40] the CT cube is transformed to a **cylindrical harmonic** representation. A DRR is then created by projecting each of these harmonics from the chosen projection point to the imaging plane and taking the superposition of all the projections.

A technique for **hardware based** DRR generation using consumer-grade computer graphics hardware is introduced in [36]. This allows for rapid real-time DRR generation.

Chapter 4

Image Registration

This chapter discusses the details of the image registration process. For development and testing of the registration process it was necessary to use simulated PRs because real PRs were not yet readily available from the DPRS to be used by iThemba LABS. The generation of these simulated PRs is discussed in Section 4.1. As stated in the Introduction Chapter, the image registration process entails the minimization of the objective function P. The various components that the objective function consists of play a very important part in the way it behaves. Section 4.2 investigates the effect of these components in order to select them to be beneficial to the objective function's behaviour. In Section 4.3 the minimization of the objective function is discussed and the minimization algorithms considered are introduced.

4.1 Generating simulated PRs

Real PRs of the patient in the treatment position can be simulated by generating DRRs from the patient's CT data. Inaccuracies in patient setup are simulated by an error transformation as was done when forming the objective function in (1.1):

$$PR_{simulated} = DRR(\mathbf{T}_{simulated}), \tag{4.1}$$

where

$$\mathbf{T}_{simulated} = \{\delta_x, \delta_y, \delta_z, \delta_\phi, \delta_\rho, \delta_\theta\}\,.$$

This has the added advantage that the error in patient setup $\mathbf{T}_{simulated}$ is known exactly and can be used to quantify the accuracy of the registration process.

The DRRs used to simulate PRs need to be as similar as possible to the real PRs. To achieve this, the primary physical aspects of the DPRS are modelled and incorporated into the DRR generation process. These aspects are:

- Using the appropriate XACC that depends on the spectrum of the DPRS's X-ray tube
- Using the appropriate BHCC to compensate for beam hardening
- Modelling of the finite focal spot size (FSS) of the DPRS's X-ray tube
- Modelling of the detector system's modulation transfer function
- Modelling of noise introduced by the DPRS

The first two aspects are incorporated in the DRR generation process as described in Sections 3.2 and 3.3 on DRR generation.
While the incorporation the first two aspects yields an image of better quality by increasing image contrast, the latter three aspects worsen the image quality of the DRR by blurring the image and adding noise to the image. The modelling of a finite X-ray source is done as described in Section 3.5.3.

4.1.1 Modelling the detector system's MTF

Modelling of the detector system's modulation transfer function is incorporated by use of the DPRS's MTF. This was measured with the experimental setup illustrated in Figure 4.1. In this experiment the ESF was obtained taking a radiograph with a plate placed against the scintillation screen of

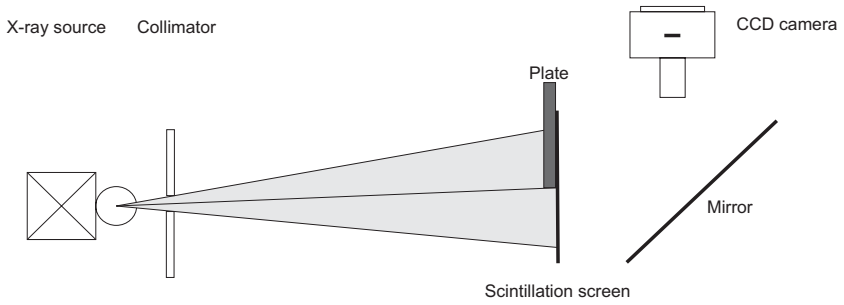


Figure 4.1: *Measuring the ESF of the DPRS.*

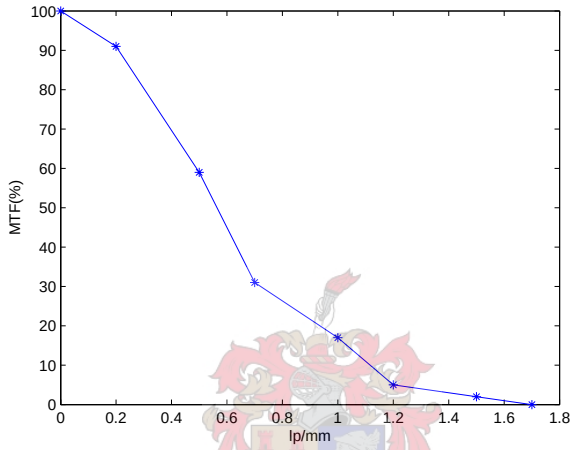


Figure 4.2: *Measured MTF of the DPRS.*

the DPRS. The MTF is calculated from the ESF by use of the Fourier transform as illustrated in Figure 3.16 and discussed in the corresponding section. The plate was positioned so that its vertical edge was in the center of the scintillation screen and co-incident with the central beam axis. Because the object-detector distance is very small for the setup with the plate against the scintillation screen, the effective size of the focal spot of the X-ray tube becomes very small according to (3.12). For this experiment geometrical unsharpness is therefore negligible, and the MTF that was calculated from the ESF is mainly due to the blurring by the detector system. The measured MTF of the DPRS is shown in Figure 4.2.

To incorporate the detector system's MTF, the DRR needs to be convolved with a kernel that approximates this MTF [33]. It was decided to approxi-

mate the MTF of the detector system with a double Gaussian kernel so that both the steep fall-off characteristic of the first part of the MTF curve as well as the slow decay of the latter part of the MTF curve would be well approximated. The double Gaussian kernel is given by

$$K(\sigma_1, \sigma_2, \alpha) = \alpha G(\sigma_1) + (1 - \alpha) G(\sigma_2). \tag{4.2}$$

where $G(\sigma_1)$ and $G(\sigma_2)$ are two Gaussian functions with different standard deviations and α an appropriate weight.

A DRR are created with the setup as in Figure 3.21(a), but with the plate against the imaging plane to simulate the setup for the calculation of the MTF of the DPRS in Figure 4.1. The MTF of the DRR can be calculated by taking the magnitude of the Fourier transform of the derivative of the horizontal profile through the center of the DRR after it is convolved with the double Gaussian kernel

$$MTF_{DRR} = \left| \mathfrak{F} \left(\frac{d}{dx} PF \left(DRR \otimes K \left(\sigma_1, \sigma_2, \alpha \right) \right) \right) \right|. \tag{4.3}$$

The parameters σ_1, σ_2 and α of the double Gaussian kernel is calculated by minimizing the least-square error between the measured MTF of the DPRS and MTF_{DRR}. Thus the values were calculated as:

$$\sigma_1 = 2.7545$$
$$\sigma_2 = 4.1465$$
$$\alpha = 0.4820$$

The detector system's MTF can now be incorporated by the following convolution:

$$DRR_{DS} = DRR \otimes K(\sigma_1, \sigma_2, \alpha).$$

The MTF of DRR_{DS} calculated with the parameters of K as above is shown (together with the MTF measured from the DPRS) in Figure 4.3, from which it can be verified that the MTF of DRR_{DS} corresponds to the MTF of the DPRS.

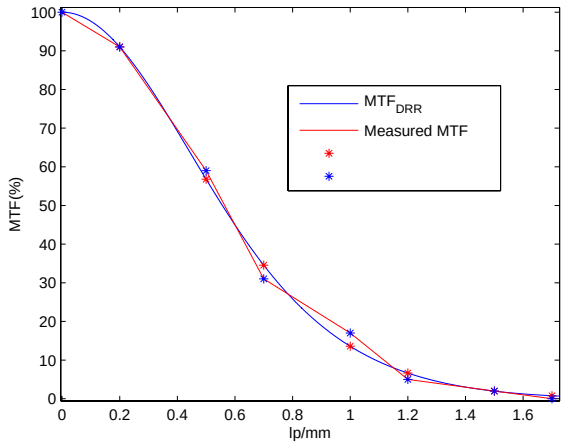


Figure 4.3: *MTF of DRR_{DS} and Measured MTF of the DPRS.*

4.1.2 Modelling of DPRS noise

The primary sources of noise in the DPRS are X-ray quantum noise, CCD chip read-out noise and digitization noise [7]. The noise produced by these sources are assumed to be spatially invariant and uncorrelated to the PR pixel values, and therefore the noise can be modelled by Gaussian noise [35, Section 5.2]. Gaussian (also called normal) noise can be added to the DRR by adding a random number to each pixel. This number is generated by a random number generator that produces normally distributed random numbers. The extent of the noise is determined by the mean and standard deviation of this normal distribution. It is important to carefully set the standard deviation (σ_{noise}) of the normal distribution to simulate the noise introduced by the DPRS. The mean of the normal distribution is set to zero so that the mean pixel value of the DRR will not be influenced. The value for σ_{noise} was calculated by matching the signal-to-noise ratio (SNR) of the noise to be added with that of a flat field image obtained from the DPRS. To calculate σ_{noise} in this way, a number of flat field images were taken with the DPRS. The intensity values of the flat field images ranged from 47664 to 65535. The average values for the mean and standard deviation of the pixel

intensity values were measured as 52048.4 and 1496.6. The SNR is given by

$$SNR_{flat} = 20\log_{10}\left(\frac{M}{STD}\right) \text{ dB}, \tag{4.4}$$

where M and STD is the mean and standard deviation. With the measured values of M and STD given above the average SNR of the flat-field images of the DPRS was calculated to be 30.827 dB.

A flat field image was also obtained by generating a DRR from a CT cube with all the voxels having the CT number of air. The M and STD of the pixel values of this DRR were calculated to be 0.9878 and 1.907×10^{-5}, which results in a SNR_{flat} of 94.286 dB. This illustrates the negligible noise the ray-tracing process introduces.

The value of σ_{noise} is calculated as 0.0284 by solving for STD in (4.4) with $M = 0.9878$ and $SNR_{flat} = 30.827$ dB. After the noise is added to the DRR the values for M and STD are 0.9881 (which shows that the mean pixel value is not significantly influenced) and 0.0283 respectively. This results in a SNR_{flat} of 30.860 dB, which matches the SNR of the DPRS.

Another value for σ_{noise} can be calculated by taking a certain percentage of the average PR pixel value. From experience with the portal verification system currently in use by iThemba LABS that uses X-ray films, the noise levels of the radiographs should not exceed 5% of the average pixel value. The calculation of the average pixel value involves the construction of a water phantom the size of a human head. The human head consists largely of the brain and other soft tissues whose interaction with X-rays are approximated well by water. The human head can therefore be roughly approximated by a CT cube whose voxels are set to the CT number of water. The idea is that the pixel values of the DRR generated from this CT cube will be representative of the average pixel values of a PR taken of a human head. A value for σ_{noise} can therefore be calculated by taking the 5% value of the mean pixel values of the DRR calculated with the water phantom.

The mean value of the DRR pixels generated with a 200mm^3 water phantom CT cube was measured to be 41.0780. The corresponding value of σ_{noise} is 2.0539 which is 5% of 41.0780.

This value of σ_{noise} calculated with the water phantom is significantly higher than the one calculated by matching the SNR of DPRS flat field images. The latter is considered to be the more realistic value for σ_{noise}, while the value calculated with the water phantom experiment should be considered an upper bound on the noise.

By incorporating the modelling of the aspects of the DPRS as discussed above, a simulated PR can be generated by

$$PR_{simulated} = \left(DRR_{XACC/BHCC}(\mathbf{T}_{simulated}) \otimes K_{FS} \otimes K_{DS}\right) + N(\sigma_{noise}), \tag{4.5}$$

where $DRR_{XACC/BHCC}(\mathbf{T}_{simulated})$ is a DRR generated with the appropriate XACC and BHCC and simulated setup error transformation $\mathbf{T}_{simulated}$, K_{FS} is the convolution kernel for modelling the finite source, K_{DS} is the convolution kernel for modelling the detector system's MTF and $N(\sigma_{noise})$ is Gaussian noise with a standard deviation of σ_{noise}.

4.1.3 Generating simulated PRs of different sizes

As mentioned in Section 2.1.3, the CCD camera of the DPRS produces PRs consisting of 1024^2 pixels. PRs of 512^2- and 256^2 pixels can be produced by the CCD camera by the binning of pixels.

To simulate the way the CCD camera produces PRs of different sizes, the DRRs generated for simulating PRs are always generated at 1024^2 pixels. Reference images of 512^2- and 256^2 pixels are then created by down-sampling the 1024^2 pixel DRR by an averaging process. This process, illustrated in Figure 4.4, is analogous to the CT data down-sampling process described in Section 3.4.1 and simulates the internal binning process of the CCD camera. Each pixel of the down-sampled DRR is given the average value of the cluster of pixels from which it is created.

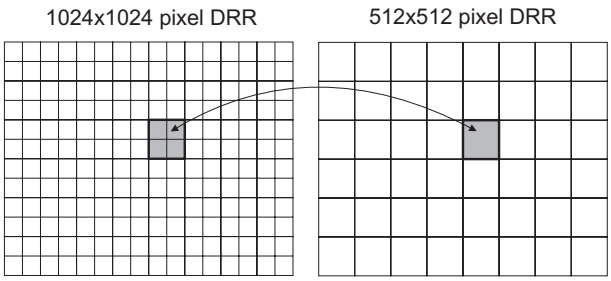


Figure 4.4: *DRR down-sampling by averaging*

4.2 Investigating the objective function P

The investigation of the objective function P that was defined in (1.1) as

$$P\left(\mathbf{T}_{DRR}\right) = M\left(PR, DRR\left(\mathbf{T}_{DRR}\right)\right),$$

plays a very important part in the solving of the image registration problem. In the first place, it allows the investigation of the effect that the various components have. In the second place it allows the evaluation of the qualities of the objective function, which is invaluable in choosing a minimization algorithm for the objective function. E.g. if the objective function suffers from many local minima, local optimization techniques will not suffice and more sophisticated global optimization techniques like simulated annealing or evolutionary based optimization should be considered.

Since the objective function has six degrees of freedom, corresponding to the three translation and three rotation parameters of the transformation $\mathbf{T}_{DRR}$, it is not possible to visualize it practically. For this reason, the investigations are done by evaluating the objective function along various one dimensional paths through the six dimensional space. Similarity curves are constructed by recording the values of the measure M for the values of $\mathbf{T}_{DRR}$ along these one dimensional paths. The similarity curves can be used to visualize and investigate the objective function. As stated in the Section 2.5, some measures return a higher value for two images that compare better and some a lower value. A desirable similarity curve is one which is smooth and increases (or decreases) to an optimum value (which is the value at

which the images compare best), and then decreases (or increases) back again.

The objective function has three main components:

- The measure M
- The portal radiograph PR
- The digitally reconstructed radiograph DRR.

As stated in the introduction, the only similarity measures that will be considered are the Mutual Information (MI) and Correlation Coefficient (CC) intensity based similarity measures.

During the investigations, simulated PRs (whose creation is described in Section 4.1) will be used for PR. Four different simulated PRs will be used with different parameters for the X-ray tube focal spot size (FSS) and for the standard deviation (σ_{noise}) of the added Gaussian noise in (4.5):

1. $PR1$: FSS = 1.5mm, $\sigma_{noise} = 0.0282$
2. $PR2$: FSS = 0.8mm, $\sigma_{noise} = 0.0282$
3. $PR3$: FSS = 1.5mm, $\sigma_{noise} = 2.0539$
4. $PR4$: FSS = 0.8mm, $\sigma_{noise} = 2.0539$

Simulated PRs of different sizes are also created as described in Section 4.1.3 resulting in $PR1_{1024}$, $PR1_{512}$ and $PR1_{256}$, and the same for $PR2$ to $PR4$. This resulted in a total of 12 different images for PR. Figure 4.5 shows the original DRR from which the simulated PRs were created, together with $PR1_{1024}$, $PR2_{1024}$, $PR3_{1024}$ and $PR4_{1024}$.
The simulated PRs were created from the CT data of a real patient, and with a view specified by one of the treatment fields for the patient. A summary of the CT data used during DRR generation is given in Table 4.1. For the investigations, no inaccuracies in patient setup are simulated, i.e. $\mathbf{T}_{simulated}$

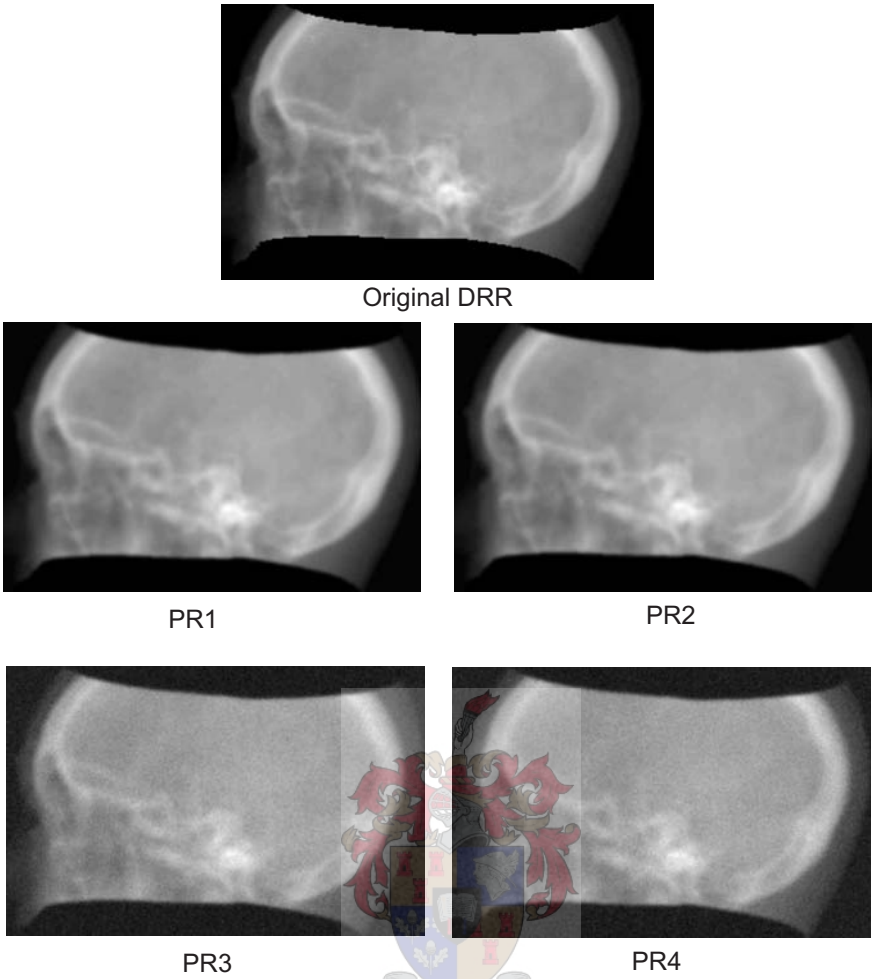


Figure 4.5: *Original DRR together with* $PR1_{1024}$, $PR2_{1024}$, $PR3_{1024}$ *and* $PR4_{1024}$.

in (4.5) is set to zero. Hence the objective function is expected to have a minimum value when the transformation $\mathbf{T}_{DRR}$ is zero:

$$\begin{aligned} \delta_x = \delta_y = \delta_z &= 0\text{mm} \\ \delta_\theta = \delta_\phi = \delta_\rho &= 0^\circ. \end{aligned}$$

The DRRs used for DRR in the investigations are all created from the same CT data that the simulated PRs are created from. The components of DRR generation that have an effect on the objective function are investigated. These components are:

Table 4.1: *Summary of CT data.*

Field-of-view size	260mm×260mm×136mm
CT slice dimensions	256×256
CT slice pixel size	1.015625mm^2
Number of slices	68
Slice separation	2mm
Slice thickness	2mm

- The parameters of the transformation $\mathbf{T}_{DRR}$ for which DRR is calculated, i.e. the translation parameters δ_x, δ_y and δ_z, and the rotation parameters δ_θ, δ_ϕ and δ_ρ.
- Whether the appropriate XACC and BHCC are used.
- The number of pixels that the DRR is generated for.
- The interpolation of the DRR.
- The slice separation and thickness of the CT data.

For each investigation a number of similarity curves are constructed and compared with one another. For each similarity curve of a particular investigation, one of the 12 simulated PR images needs to be selected for PR, and the parameters for the generation of DRR need to be selected. In order to isolate the influence that the component under investigation has on the objective function, the other components need to be selected in a way that will allow the objective function to perform in the best possible way and be kept constant during the investigation.
The following is the setting for components which will allow the objective function to perform at its optimum:

- PR selected as $PR2_{1024}$. This is because $PR2_{1024}$ has:
  - an FSS of 8mm and is therefore blurred less than $PR1_{1024}$
  - added Gaussian noise with a standard deviation of 0.0282 and is therefore less noisy than $PR3_{1024}$ or $PR4_{1024}$.

  - 1024^2 pixels, which results in better resolution than $PR2_{512}$ and $PR2_{256}$.

- DRR is calculated with
  - the parameters of $\mathbf{T}_{DRR}$ set to form the one dimensional path through the objective function
  - the appropriate XACC and BHCC, since the simulated PR are generated with them
  - 1024^2 pixels, which is equal to the size of $PR2_{1024}$
  - the same CT data that $PR2_{1024}$ is created from

As stated earlier, a desirable similarity curve is one which increases (or decreases) to an optimum value and the decreases (or increases) back again. During the investigations, the three main properties of a similarity curve that needs to be looked at are:

1. The smoothness of the curve. If the curve is very noisy, the objective function suffers from local minima which make the task of minimizing the objective function very difficult.

2. The position of the optimum value. If the position of the optimum is very close to the correct value, the accuracy of the objective function is very high.

3. The slope of the similarity curve near the optimum value. The steeper the slope of the curve, the faster the minimization process. This is especially the case for gradient based minimization algorithms.

4.2.1 The parameters of $\mathbf{T}_{DRR}$

The aim of this investigation is to find out how each of the six parameters of the transformation $\mathbf{T}_{DRR}$ influences the objective function. This is done by varying only one parameter of $\mathbf{T}_{DRR}$ at a time. In this way the values of M can be recorded along a one dimensional path and be easily visualized

Table 4.2: *Values of the components not under investigation.*

PR used	**XACC used**	**BHCC used**	**DRR size**
$PR2_{1024}$	yes	yes	1024^2

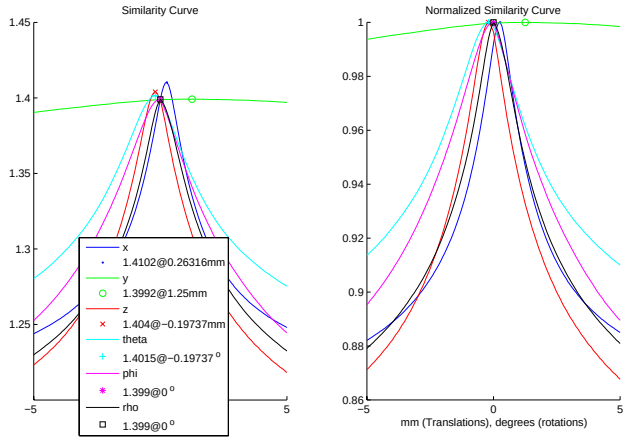


Figure 4.6: *Mutual Information similarity curves.*

as a similarity curve.

For the investigation of a particular parameter, say the translation δ_x, the parameter is varied from -5mm (or -5° for a rotation parameter) to 5mm (or 5°) in 150 steps. This is according to the limitations set in Section 2.3 and results in a step size of 0.066mm (or 0.066°), which limits the accuracy of the similarity curve to the same size. The other translation and rotation parameters are kept at 0mm (or 0°).

The components not under investigation are set to their optimum values as shown in Table 4.2.

The results of this investigation are shown in Figures 4.6 and 4.7 for the MI and CC similarity measures respectively.

The left hand side of Figure 4.6 shows the MI similarity curves for all the translation and rotation parameters of $\mathbf{T}_{DRR}$. The legend lists the maximum value for each curve together with the value of the parameter at which this maximum value occurred, e.g. the maximum value for the curve investigating the influence of δ_x are 1.4102 and occurred at $\delta_x = 0.26316$mm.

The maximum values for the curves are not all the same, nor are the posi-

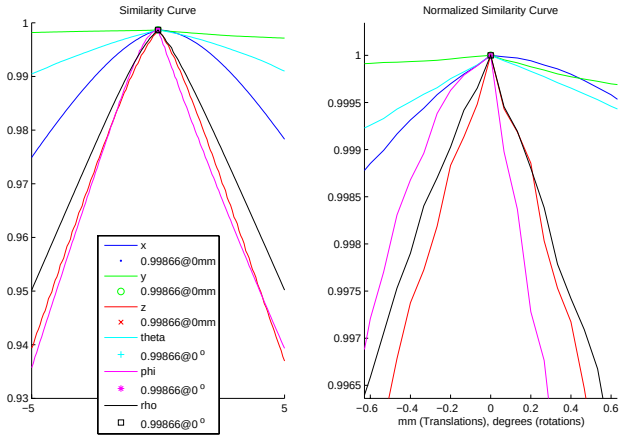


Figure 4.7: *Correlation Coefficient similarity curves.*

tions of the maximum values. Since the correct values for all the parameters are 0mm and 0°, the curves for δ_{phi} and δ_{rho} have their maximum values at the correct position.

The curve for the translation parameter δ_y is the least accurate with a maximum value at 1.25mm. As explained in Section 2.3 this is not surprising, since detecting small changes in the out-of-plane translation δ_y is very difficult.

The right hand side shows the same curves, but normalized to have maximum values of 1 so that the slope of the different curves are easier to compare to one another. The slopes of of the in-plane translations and rotation δ_x, δ_z and δ_{rho} are steeper than those of the out-of-plane rotations δ_θ and δ_{phi}, while the out-of-plane translation δ_y hardly a very small slope.

Figure 4.7 shows the CC similarity curves for the parameters of $\mathbf{T}_{DRR}$ (the right hand side is a zoomed-in part of the normalized curves). As can be seen from the legend, all the parameters reach the same maximum value at the correct position of 0mm and 0°. The CC measure is therefore more accurate than the MI measure in detecting small changes in each of the parameters of $\mathbf{T}_{DRR}$.

The slope of the CC curves are all very different, ranging from δ_y which (as was the case for MI) has a very small slope, to δ_z which has a very steep slope.

Table 4.3: *Values of the components not under investigation.*

PR used	**DRR size**	**Parameter varied**
$PR2_{1024}$	1024^2	δ_x

From this investigation it is gathered that the objective function is more accurate for the CC similarity measure than for the MI similarity measure. For MI, the out-of-plane translation parameter δ_y is much less accurate than the other parameters of $\mathbf{T}_{DRR}$.
While this investigation only changed a single parameter of $\mathbf{T}_{DRR}$ at a time, similarity curves where more than one parameter of $\mathbf{T}_{DRR}$ are changed at a time did not show any situations where the behaviour differs significantly.

4.2.2 The use of the XACC and the BHCC

The aim of this investigation is to find out how the use of the XACC and BHCC for DRR influences the objective function. The path for which the similarity curves are calculated, is constructed by varying only δ_x while keeping the other parameters of $\mathbf{T}_{DRR}$ at 0mm and 0°.
Three curves are constructed by comparing $PR2_{1024}$ with DRR represented by three different DRRs. One with the DRR generated by not using the XACC or the BHCC, one with the DRR generated by using the XACC but not the BHCC, and one with the DRR generated by using both the XACC and the BHCC.
The components not under investigation are set to their optimum values as shown in Table 4.3.

Figure 4.8 shows the MI curves for this investigation, with the curves on the right hand side normalized to have a maximum value of one. The curve created by not using the XACC is less smooth and has a slope that is much less steep than the curves created when using the XACC. The curves created by using the BHCC and those not using the BHCC are virtually identical.

Figure 4.9 shows the CC curves. For CC, there is not such a big difference between the curves using the XACC and the curves that don't. The curves

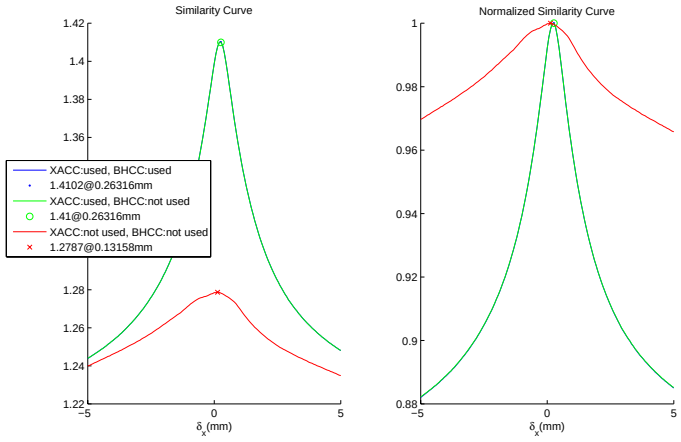


Figure 4.8: *Mutual Information similarity curves.*

for using the BHCC and those not using the BHCC are once again virtually identical.

From the curves above, it might be concluded that using the BHCC does not make any difference. It should be remembered that the BHCC depends on the X-ray tube parameters, and that the tube voltage of the X-ray tube can be adjusted. For all the DRRs generated for these investigations, a mid-range X-ray tube voltage of 75kV is used. Figure 4.10 shows the MI curves created by using DRRs generated according to a tube voltage of 130kV (the X-ray tube's upper limit). It can be seen (on the curves on the left hand side, which is a zoomed-in view) that the difference between the two curves created by using and not using the BHCC are bigger for the 130kV pair than for the 75kV pair. From this investigation it is seen that when using the MI similarity measure, the XACC increases the accuracy and the steepness of slope of the objective function. Using the BHCC has a much less profound impact on the objective function.

When using the CC similarity measure, the use of the XACC and BHCC has very little impact on the objective function.

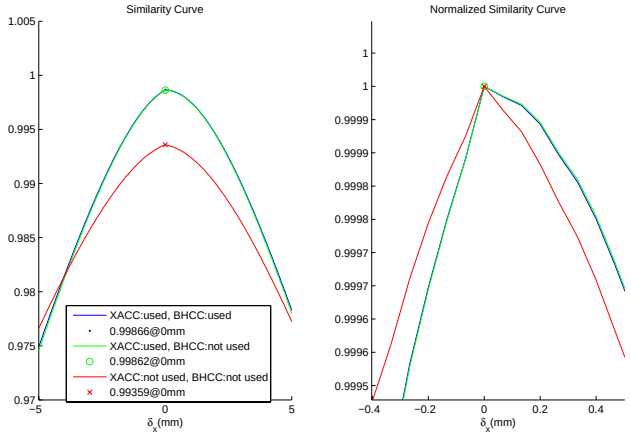


Figure 4.9: *Correlation Coefficient similarity curves.*

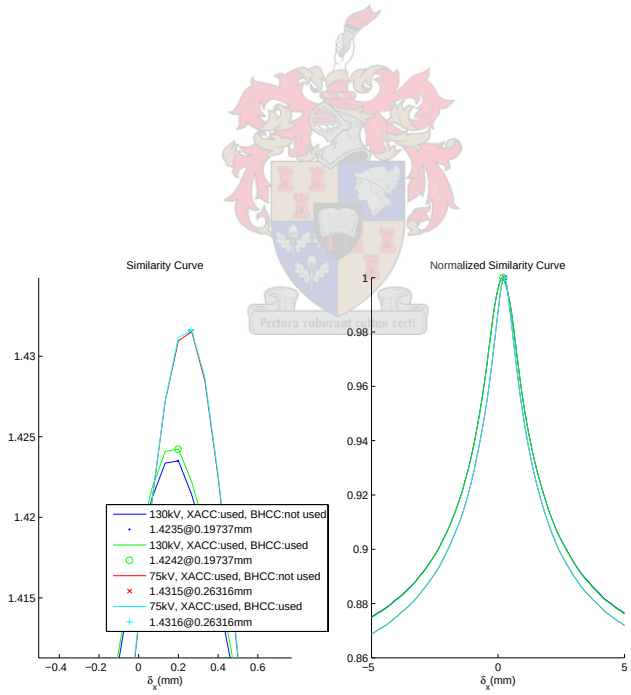


Figure 4.10: *Mutual Information similarity curves.*

Table 4.4: *DRR and PR sizes, and the DRR generation time (seconds).*

DRR size	**PR used**	**DRR time**
256^2	$PR2_{256}$	1.44
512^2	$PR2_{512}$	4.62
1024^2	$PR2_{1024}$	17.24
256^2 interpolated to 512^2	$PR2_{512}$	2.06
256^2 interpolated to 1024^2	$PR2_{1024}$	3.48
512^2 interpolated to 1024^2	$PR2_{1024}$	6.68

Table 4.5: *Values of the components not under investigation.*

PR used	**XACC used**	**XACC used**	**Parameter varied**
$PR2$	yes	yes	δ_x

Since the use of the XACC and the BHCC during DRR generation comes at virtually no extra computational cost and may have a positive impact, albeit small, they should always be used.

4.2.3 The number of pixels and the interpolation of DRRs

The aim of this investigation is to find out how the number of pixels which DRR consists of, influences the objective function. The DRR representing DRR can either be created with the same number of pixels as the PR it is to be compared with, or be interpolated to be compared with a PR that consist of more pixels. Hence, six similarity curves are constructed for this investigation by comparing DRR of different sizes with PR. The sizes of DRR which are investigated, together with the PR with which they are to be compared with, and the time it takes[1] to generate DRR are shown in Table 4.4.

The components not under investigation are set to their optimum values as shown in Table 4.5.

[1]The hardware and software with which these times are calculated are stated in Appendix A.

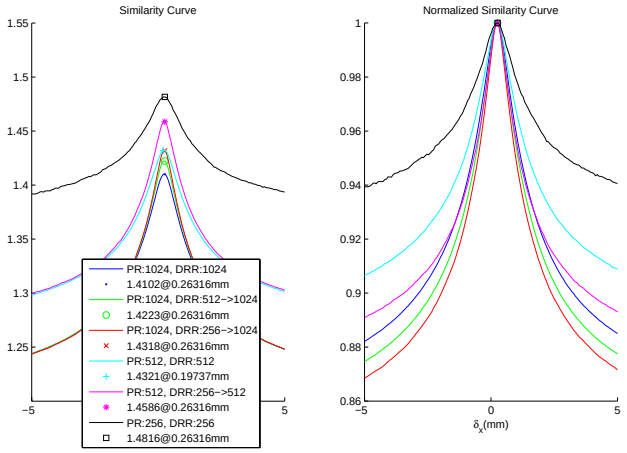


Figure 4.11: *Mutual Information similarity curves.*

Figure 4.11 shows the MI curves for this investigation. The maximum values for all the curves have a position of either 0.20mm or 0.26mm, which means they are all of the same accuracy. From the right hand side of Figure 4.11, it is seen that the curves created by comparing $PR2_{1024}$ with a DRR either generating directly with 1024^2 pixels or interpolated to 1024^2 pixels, showed the steepest slope. It is surprising to see that the curves created by interpolating the 1024^2 pixel DRR shows steeper slopes than the curves for the original 1024^2 pixel DRR. This can be ascribed to the smoothing effect that interpolation has on the images. The curve created by comparing the 256^2 pixel DRR with $PR2_{256}$ shows the smallest slope, and also becomes noisy further away from 0mm.

Figure 4.12 shows the CC curves for this investigation. The zoomed-in view of the normalized curves on the right hand side shows that the CC curves created by using the DRR with the most pixels were the most accurate, with the curve created by comparing $PR2_{1024}$ with the 1024^2 pixel DRR having its maximum value at the correct position of 0mm. This is more accurate than any of the MI curves. All the other CC curves, however, suffer from local minima.

The results of this investigation are very important. It shows that when using the MI similarity measure, the same accuracy can be achieved with

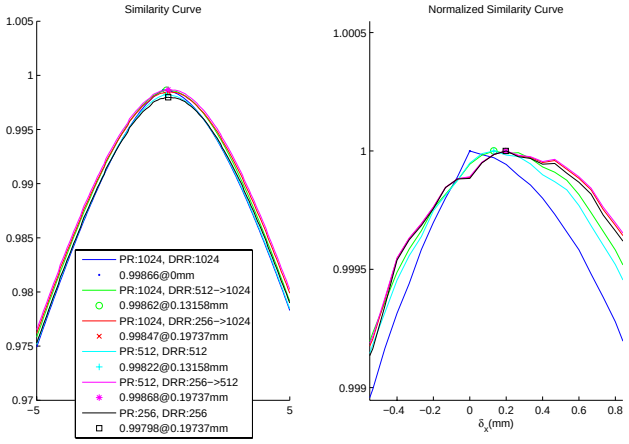


Figure 4.12: *Correlation Coefficient similarity curves.*

a 256^2 pixel DRR interpolated to 1024^2 pixels than when using an original 1024^2 pixel DRR. It also shows that the curves produced with the interpolated DRR have a steeper slope. When considered that the interpolated DRR can be generated in 3.48 seconds compared to the 17.24 seconds of the original 1024^2 pixel DRR, it is clear that interpolated DRRs can be used to reduce the evaluation time of the objective function, and therefore the verification process. The fact that the interpolation of the DRRs does not effect the accuracy of similarity curves very much is in line with the results of the experiments conducted in Section 3.5.3, which showed that the interpolation of DRRs does not have a profound impact on the DRR image quality.

The results of the investigation regarding CC is also very important. On the one hand, the curve created with the 1024^2 pixel DRR is the most accurate of all the curves, and more accurate than any of the MI curves. On the other hand, the curves not created with the 1024^2 pixel DRR all suffer from local minima. Therefore, the CC measure can only be used for the objective function together with the very computationally expensive 1024^2 pixel DRRs.

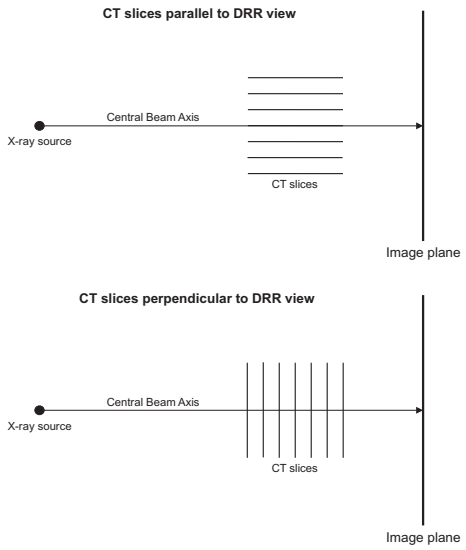


Figure 4.13: *Direction of CT slices relative to central beam axis or DRR view*

4.2.4 The slice separation and thickness of the CT data

The aim of this investigation is to find out how the slice separation and thickness of the CT data from which DRR is generated, influence the objective function. The slice thickness and separation of the CT slices are usually not of the same size as the size of the pixels of the CT slices, and therefore the voxels of the CT cube are not cubical. Hence, the effect that the slice separation and thickness have on DRR depends on the particular view of DRR with relation to the CT slices. The two extreme cases are when the DRR view (which is defined by the direction of the central beam axis) is parallel with the CT slices, and when the DRR view is perpendicular with the CT slices. These two cases are illustrated in Figure 4.13.

When the DRR view is parallel to the CT slices, the effect of differences in the CT slice separation and thickness on the image quality of the DRR is much more pronounced than when the DRR view is perpendicular to the slices.

The CT slice separation can be altered by interpolation of the CT data as explained in Section 3.4.1. To investigate the effect that altering the CT

Table 4.6: *Values of the components not under investigation.*

PR used	**DRR size**	**XACC used**	**XACC used**	**Parameter varied**
$PR2_{1024}$	1024^2	yes	yes	δ_x

Table 4.7: *DRR generation times.*

Slice separation and thickness	**Nr. of slices**	**DRR time (sec) parallel view**	**DRR time (sec) perpendicular view**
1mm	132	2.86	2.83
2mm	68	1.91	1.50
5mm	28	1.30	0.69
10mm	14	1.17	0.43

slice separation has on the objective function, a CT cube is created with $1mm^3$ cubical voxels and then down-sampled to obtain CT cubes with slice separations of 1mm, 2mm, 5mm and 10mm. DRRs created from these CT cubes from a view that is parallel to the CT cube's slices, and from a view that is perpendicular to the CT cube's slices, are compared with PR. Such DRRs are shown Figure 4.14 (view parallel to slices) and Figure 4.15 (view perpendicular to slices). The components not under investigation are set to their optimum values as shown in Table 4.6. The view of the DRR relative to the CT slices not only has an effect on the image quality of the DRR, but also on the speed of DRR generation. Table 4.7 shows the time needed for DRR generation from CT data sets consisting of different numbers of slices[2].

[2]The hardware and software with which these times are calculated are stated in Appendix A.

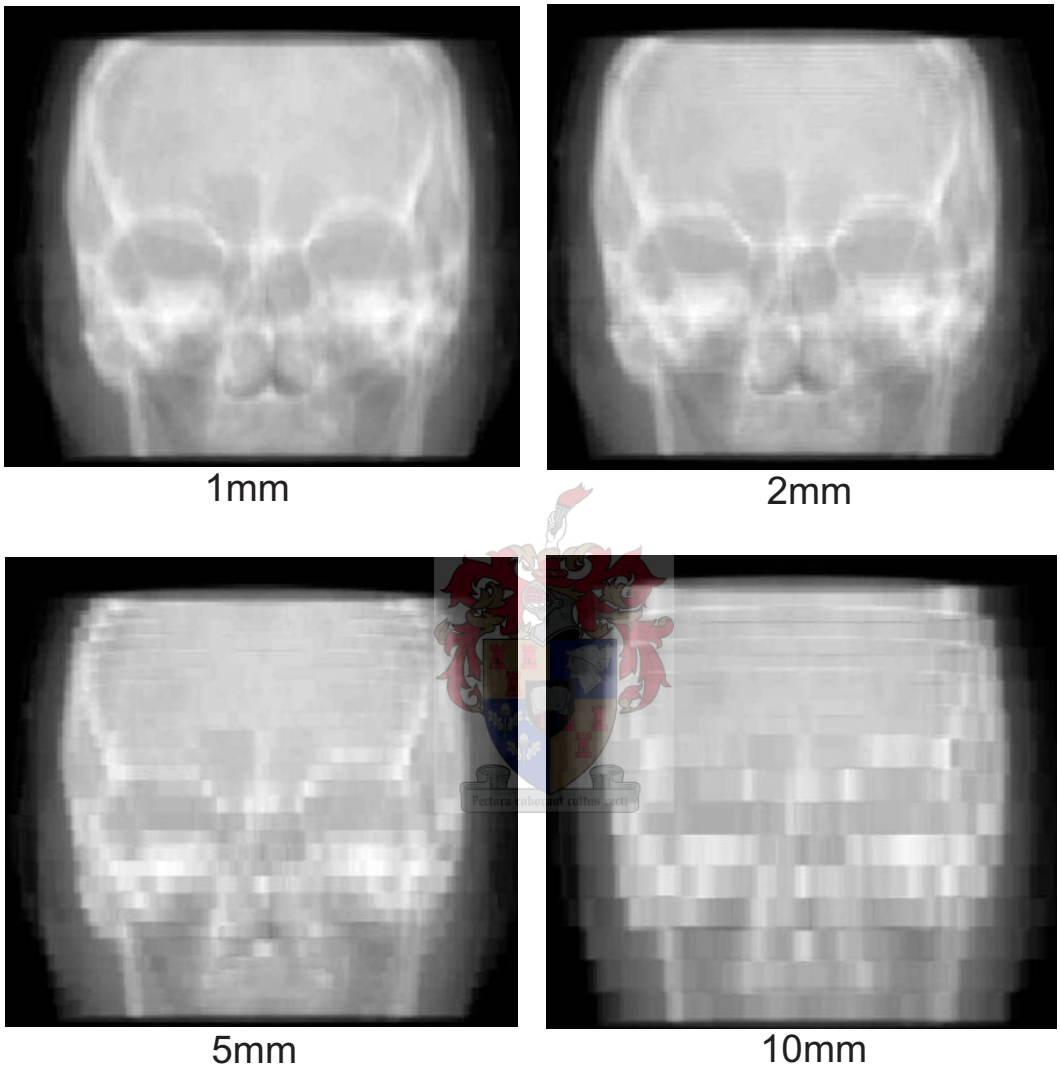


Figure 4.14: *DRR generated from CT data with 1mm, 2mm, 5mm and 10mm slices. DRRs generated with a view* ***parallel*** *to the CT image slices*

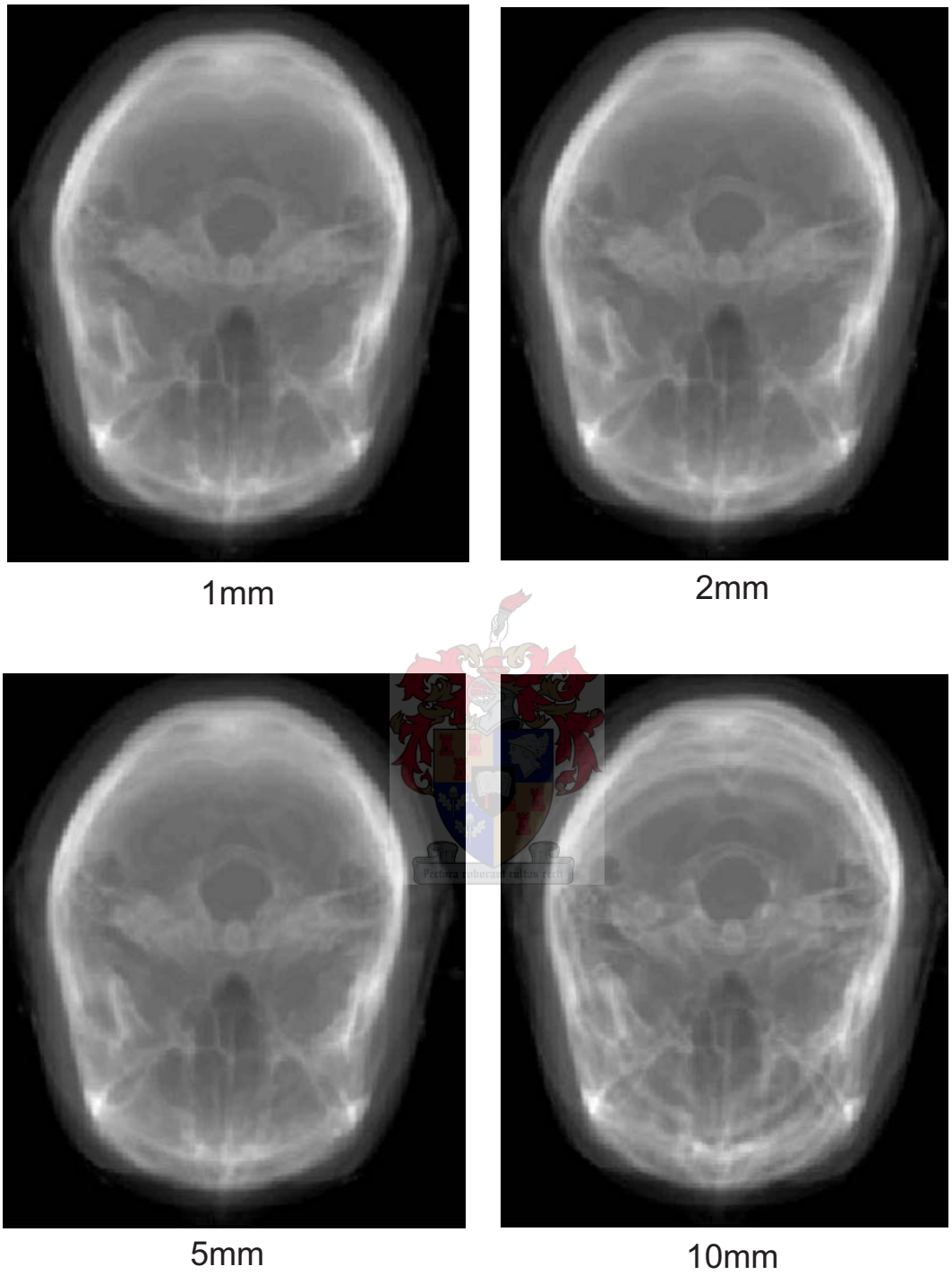


Figure 4.15: *DRR generated from CT data with 1mm, 2mm, 5mm and 10mm slices. DRRs generated with a view* ***perpendicular*** *to the CT image slices*

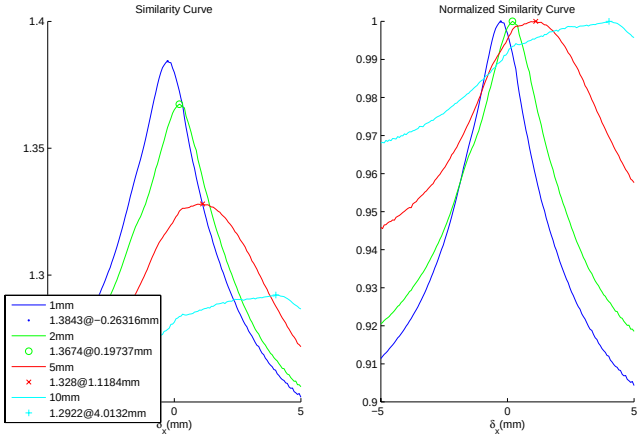


Figure 4.16: *Mutual Information similarity curves.*

Figure 4.16 shows the result of the investigation with the DRR view parallel to the CT slices. It can be clearly seen that the accuracy of the curves deteriorate significantly as the slice separation is increased from 1mm to 10mm. From the normalized curves on the right hand side, it is seen that the steepness of the slopes of the curves also becomes smaller as the slice separation is increased.

Figure 4.17 shows the result of the investigation with the DRR view perpendicular to the CT slices. These curves do not suffer from inaccuracies as the curves for the parallel view do. From the normalized curves on the right hand side, it is seen that the steepness of the slope is only significantly smaller for the extreme 10mm slice separation case.

It is gathered from this investigation that it is very important that the DRRs are always generated from CT data with the smallest available slice thickness and separation. Although the effect is less serious when the DRRs are generated with a view that is close to perpendicular to the slices, it should be remembered that a patient treatment plan usually consists of multiple treatment fields, some of which might require a DRR view that would suffer from using CT data with thick slices.
This investigation also shows that down-sampling the CT data to increase the speed of the DRR generation process is not a good idea because of the

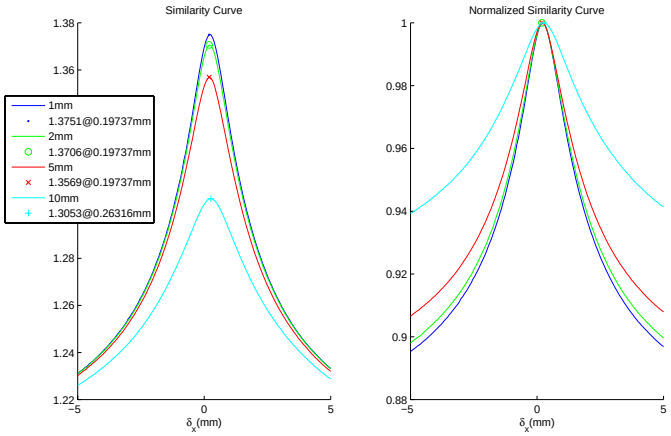


Figure 4.17: *Mutual Information similarity curves.*

severe decrease in accuracy. This is not surprising, since the experiments conducted in Section 3.5.3 showed that the down-sampling of CT data causes a decrease in DRR image quality.

When only CT data with thick slices are available, the CT data can be up-sampled as discussed in Section 3.4.1. To investigate what the effect on the objective function would be, the CT cube of the down-sample investigation which consists of 5mm slices was up-sampled to obtain CT cubes consisting of 3mm and 1mm slices. The similarity curves obtained by using these cubes are shown in Figure 4.18.

By up-sampling the CT data the accuracy of the similarity curves are improved. It can be seen that the accuracy of the up-sampled cube with 1mm slices are comparable with the accuracy of the original cube with 1mm slices in Figure 4.16.
The right hand side of Figure 4.18 shows that the slopes of the curves created with the up-sampled CT cubes are also steeper than the one for the cube with the 5mm slices.
It is gathered from this investigation that when the available CT data consists of thick slices, the CT cube should be up-sampled to a smaller slice thickness.

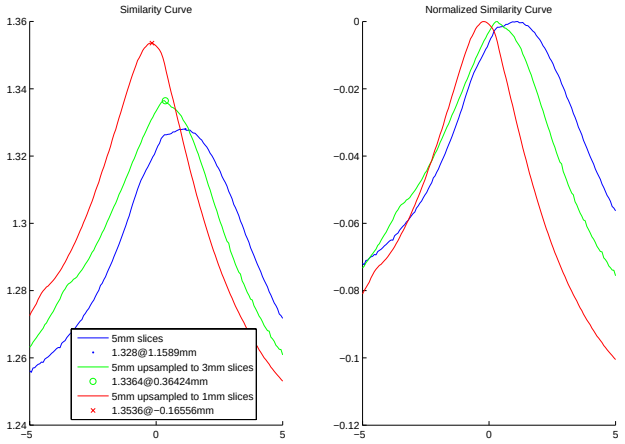


Figure 4.18: *Mutual Information similarity curves.*

Table 4.8: *Values of the components not under investigation.*

DRR size	**XACC used**	**XACC used**	**Parameter varied**
1024^2	yes	yes	δ_x

4.2.5 The effects of the PR

The aim of this investigation is to find out how the use of different simulated PRs for PR influence the objective function. As stated at the beginning of this section, there are four different simulated PRs. The 1024^2 pixel versions of these simulated PRs ($PR1_{1024}$, $PR2_{1024}$, $PR3_{1024}$ and $PR4_{1024}$) will be used to create the similarity curves. The components not under investigation are set to their optimum values as shown in Table 4.8.

Figure 4.19 shows the MI curves for the investigation. The difference between the curves created using $PR1_{1024}$ and $PR2_{1024}$, and the difference between the curves created using $PR3_{1024}$ and $PR4_{1024}$ is very small. The difference between the PRs with different noise levels is more pronounced, though the influence on the accuracy of the curves is still small. However, from the figure on the right hand side can be seen that the impact of the noise on the steepness of the slopes of the curves is very large.

Figure 4.20 shows the CC curves for the investigation. The same behaviour

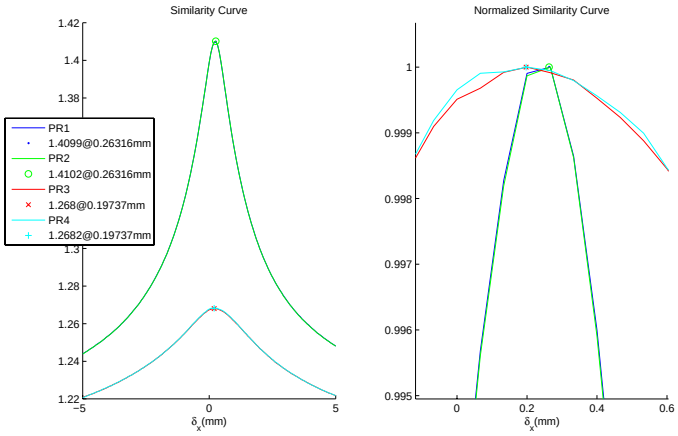


Figure 4.19: *Mutual Information similarity curves.*

as with the MI curves is noticed. The different FSS values do not have any noticeable influence, and the different noise levels do not have any impact on the accuracy of the curves. Unlike for the MI curves, the slopes of the CC curves are not significantly affected by different noise levels.

These investigations shows that the noise levels of PR have an influence on the objective function when using the MI measure, but not so much when using the CC measure. The use of different FSS values has, however, virtually no impact on the objective function. It should be remembered that the noise levels of $PR1_{1024}$ and $PR2_{1024}$ are considered normal, while the noise levels of $PR3_{1024}$ and $PR4_{1024}$ are an extreme case as stated in Section 4.1.2.

4.2.6 Conclusions from the investigations

From the investigations conducted it is clear that the components of the objective function can be selected in two ways to achieve a smooth, accurate objective function. The one is to generate the DRR with 256^2 pixels (using the XACC and the BHCC) and interpolate it to 1024^2 pixels and compare it with a 1024^2 pixel PR by using the MI similarity measure. As can be seen in Figure 4.21 all the parameters of $\mathbf{T}_{DRR}$ are within an accuracy of 0.3mm, except for the out-of-plane translation δ_y. With this objective function, the

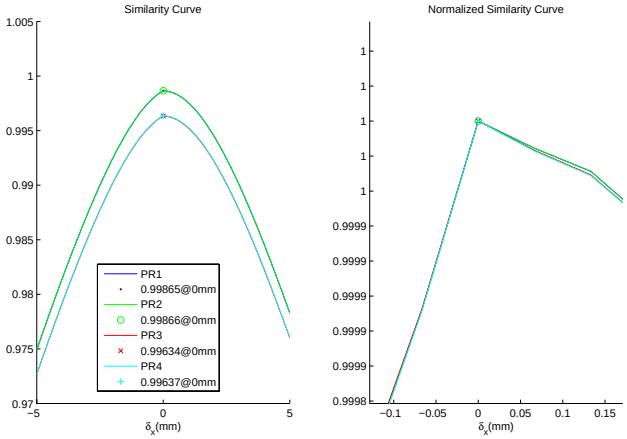


Figure 4.20: *Correlation Coefficient similarity curves.*

minimization algorithm might therefore not be able to find the correct value for δ_y. However, the evaluation of this objective function is very fast because of the use of interpolated DRRs.

The second way is to generate the DRR (using the XACC and the BHCC) with 1024^2 pixels and comparing it with a 1024^2 pixel PR by using the CC similarity measure. As can be seen from Figure 4.22, this method yields an objective function which is accurate for all six parameters of $\mathbf{T}_{DRR}$. The downside of this method is the time it takes to evaluate the objective function. The evaluation of the objective function for this method takes almost five times longer than for the first method.

A summary of the components of the two objective functions and their respective function evaluation times are given in the Table 4.9. The DRRs of both methods are created using the XACC and BHCC. The function evaluation times include the DRR generation time, the time it takes to interpolate the DRR (if necessary) and the time it takes for the measure to compare the two images. The simulated PRs are created beforehand, and their computation time can therefore be omitted.

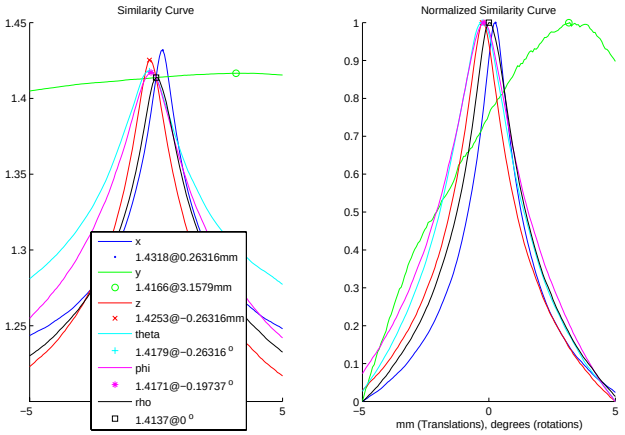


Figure 4.21: *Mutual Information similarity curves.*

Table 4.9: *Objective functions $P1$ and $P2$.*

Objective function	**Measure**	**DRR size**	**PR size**	**Evaluation time**
$P1$	MI	256^2 interpolated to 1024^2	1024^2	4.0561 seconds
$P2$	CC	1024^2	1024^2	14.5563 seconds

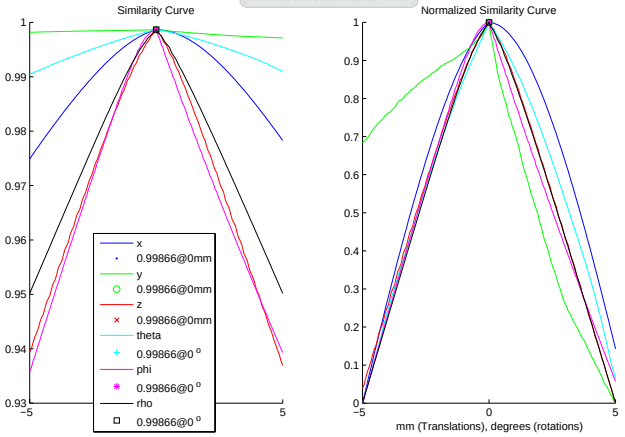


Figure 4.22: *Correlation Coefficient similarity curves.*

Since objective functions $P1$ and $P2$ are without local minima, local minimization algorithms can be used to find their global minima. Since both the MI and CC similarity measures returns a higher value for two images that compare better, $-M$ will be used when minimizing these objective functions as discussed in Section 2.5. Therefore the objective functions becomes:

$$P1\left(\mathbf{T}_{DRR}\right) = P2\left(\mathbf{T}_{DRR}\right) = -M\left(PR, DRR\left(\mathbf{T}_{DRR}\right)\right)$$

4.3 Minimization of the objective function

As stated in Section 2.5, five multi-dimensional minimization algorithms based on the line search strategy were chosen to be implemented and investigated. This is therefore not an exhaustive investigation of all the available minimization approaches, but an investigation of a few reasonable ones to establish how well minimization solves the problem.
Of the five methods, two are non-gradient based methods. They are the unit vector direction set method (UVDS) and Powell's direction set (PDS) method. The other three methods are gradient based. They are the steepest descent (SD) method, the Fletcher-Reeves-Polak-Ribiere (FRPR) conjugate gradient method and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton method.

As discussed in Section 2.5, the line search strategy works by iteratively calculating new points that yield a lower function value. Line search minimization algorithms differ in the way that the direction $\mathbf{n}$ and the step size λ are calculated that will take the algorithm to the next point.
The step size λ can be set to a fixed value calculated in some heuristic way. Choosing a too small step size will leave the algorithm proceeding very slowly to the minimum, while choosing a too large step size might let the algorithm jump over the minimum.
While the BFGS algorithm calculates the step size explicitly, the accuracy of this Newton step depends on the accuracy of the assumptions that the algorithm makes, as will be explained later. The Newton step is therefore not always the best option for λ, and a different value for λ can be used.

Another way to calculate λ is to do a one-dimensional line minimization along the direction $\mathbf{n}$. Since λ is calculated automatically and accurately in this way, it is the method that will be used for all the minimization algorithms investigated.

As stated in Section 2.5 an initial point $\mathbf{p}_0$ is needed at which the minimization is started. Since the current treatment position, given by T_{err}, is close to the correct treatment position, given by $\mathbf{T}_{DRR} = \mathbf{0}$, the correct treatment position is a good starting point. Therefore the initial point is always taken to be $\mathbf{p}_0 = \mathbf{0}$.

4.3.1 Line minimization method

The one-dimensional paths through the parameter spaces of both objective functions $P1$ and $P2$ (as visualized by the similarity curves in Section 4.2) can be roughly approximated by parabolic functions. Therefore, one-dimensional minimization based on fitting a parabolic function to the objective function values along the line that needs to be minimized, can be used. Since the function values are close to parabolic the minimum of the line will be close to the minimum of the parabolic function [41, Section 10.2].

To construct a parabolic function three points are needed along the line. These points (p_a, p_b and p_c) should be so that

$$\mathrm{p}_a < \mathrm{p}_b < \mathrm{p}_c \quad \text{and} \quad \mathrm{fp}_a > \mathrm{fp}_b \quad \text{and} \quad \mathrm{fp}_b < \mathrm{fp}_c, \tag{4.6}$$

where fp_a, fp_b and fp_c are the objective function values on the line at the respective points.

Because of these restrictions on the three points, the position p_{min} of the minimum of the parabola through these points is bound by p_a and p_c. After calculating the minimum of the parabola, the three points are changed as follows:

- If $\mathrm{fp}_{min} < \mathrm{fp}_b$
  - If $\mathrm{p}_{min} < \mathrm{p}_b$

        * Set $\mathrm{p}_a = \mathrm{p}_a$
        * Set $\mathrm{p}_b = \mathrm{p}_{min}$
        * Set $\mathrm{p}_c = \mathrm{p}_b$
    - If $\mathrm{p}_{min} > \mathrm{p}_b$
        * Set $\mathrm{p}_a = \mathrm{p}_b$
        * Set $\mathrm{p}_b = \mathrm{p}_{min}$
        * Set $\mathrm{p}_c = \mathrm{p}_c$

- If $\mathrm{fp}_{min} > \mathrm{fp}_b$
    - If $\mathrm{p}_{min} < \mathrm{p}_b$
        * Set $\mathrm{p}_a = \mathrm{p}_{min}$
        * Set $\mathrm{p}_b = \mathrm{p}_b$
        * Set $\mathrm{p}_c = \mathrm{p}_c$
    - If $\mathrm{p}_{min} > \mathrm{p}_b$
        * Set $\mathrm{p}_a = \mathrm{p}_a$
        * Set $\mathrm{p}_b = \mathrm{p}_b$
        * Set $\mathrm{p}_c = \mathrm{p}_{min}$

For the new set of points, the distance $D_{ac} = \mathrm{p}_c - \mathrm{p}_a$ will be smaller than for the previous set of points. This process of parabolic fitting is repeated until D_{ac} is smaller than a designated size.

The minimum of the parabola through the three points are given by

$$\mathrm{p}_{min} = \mathrm{p}_b - \frac{1}{2}\frac{(\mathrm{p}_b - \mathrm{p}_a)^2(\mathrm{fp}_b - \mathrm{fp}_c) - (\mathrm{p}_b - \mathrm{p}_c)^2(\mathrm{fp}_b - \mathrm{fp}_a)}{(\mathrm{p}_b - \mathrm{p}_a)(\mathrm{fp}_b - \mathrm{fp}_c) - (\mathrm{p}_b - \mathrm{p}_c)(\mathrm{fp}_b - \mathrm{fp}_a)}.$$

This formula only fails if the three points are collinear, in which case the denominator becomes zero [41, Section 10.2]. This is prevented by the restrictions of (4.6). When D_{ac} becomes very small, the fraction in the formula becomes very large because the denominator becomes very close to zero. Since the limit on the D_{ac} is set to 0.1mm (which is large relative to the machine's floating point precision), any numerical misbehaviour of this

kind is avoided. The value of 0.1mm for D_{ac} is chosen based on the accuracy results of the similarity curves of Section 4.2.6.

The calculation of the initial set of three points is done by the following bounding algorithm:

1. Calculate the function value fp_1 of the first point p_1 (which is taken as an input parameter)
2. Calculate the second point as $\mathrm{p}_2 = \mathrm{p}_2 + \alpha$, where α is some positive value. Also calculate its function value fp_2
3. If $\mathrm{fp}_1 < \mathrm{fp}_2$
   - Set $\mathrm{p}_b = \mathrm{p}_1$ and $\mathrm{p}_c = \mathrm{p}_2$
   - Find a point p_3 so that if $\mathrm{p}_3 = \mathrm{p}_1 - \alpha - C\beta$, then $\mathrm{fp}_3 > \mathrm{fp}_1$ where β is some positive value and C is the smallest value form the set $\{0, 1, 2 \ldots\}$
   - Set $\mathrm{p}_a = \mathrm{p}_3$
4. If $\mathrm{fp}_1 > \mathrm{fp}_2$
   - Set $\mathrm{p}_a = \mathrm{p}_1$ and $\mathrm{p}_b = \mathrm{p}_2$
   - Find a point p_3 so that if $\mathrm{p}_3 = \mathrm{p}_2 + \alpha + C\beta$, then $\mathrm{fp}_3 > \mathrm{fp}_2$ where β is some positive value and C is the smallest value form the set $\{0, 1, 2 \ldots\}$
   - Set $\mathrm{p}_a = \mathrm{p}_3$
5. If $\mathrm{fp}_1 = \mathrm{fp}_2$
   - Increase α and restart the bounding algorithm

The smaller the initial value of D_{ac} (resulting from the bounding algorithm) is, the fewer iterations of parabolic fitting are required to get D_{ac} within the designated limit. Therefore the value of α should not be to large, so that the initial D_{ac} would be limited if the starting point is already close to the minimum.

The value of β should be chosen large enough so that it won't take many function evaluations to find a value of C that results in a valid p_3 when the starting point is far from the minimum.
From experimenting with the line minimization algorithm, it was found that values of 1 and 5 for α and β resulted in the bounding algorithm using the lowest number of function evaluations.
Since the bounding algorithm uses a variable amount of function evaluations (depending on the position of the starting point relative to the minimum) and the number of iterations of parabolic fitting also varies, the total number of function evaluations required for one line minimization are not constant. From experience with the line minimization algorithm it was found that the number of function evaluations required for a line minimization ranges from 3 (for a starting point very close to the minimum) to about 15 (for a starting point very far away from the minimum).

4.3.2 Calculating the gradient

The gradient based minimization algorithms must be able to calculate the gradient of the objective function at any arbitrary point. Since the evaluation of objective functions $P1$ and $P2$ are expensive, the very simple forward difference method (see [41, Section 5.7]) that uses only one additional function evaluation is used to calculate the gradient at a certain point. The forward difference method is based on the definition of the derivative which, for a function of one variable $y = f(x)$, is:

$$\frac{dy}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}. \tag{4.7}$$

The gradient of the objective function can then be calculated by calculating the partial derivatives for each of the six parameters of $\mathbf{T}_{DRR}$ [42, Section

9.4]. Therefore the gradient of the objective function P is given by

$$\nabla P = \begin{pmatrix} \frac{\partial P}{\partial x} \\ \frac{\partial P}{\partial y} \\ \frac{\partial P}{\partial z} \\ \frac{\partial P}{\partial \theta} \\ \frac{\partial P}{\partial \phi} \\ \frac{\partial P}{\partial \rho} \end{pmatrix}$$

The size of h has a big influence on the accuracy of the gradient. According to (4.7) the smaller the size chosen for h, the more accurate the gradient will be.

An investigation into the smoothness of objective functions $P1$ and $P2$ was done by recording the values along a path through each of the objective functions. This path went through the point $\mathbf{p} = \mathbf{0}$ and in the δ_x direction, which is the direction for which the partial derivative $\frac{\partial x}{\partial P}$ is calculated. The distance between two points in the path was chosen to be a very small 0.001mm.

Although the values for $P1$ and $P2$ along these paths result in seemingly smooth curves (as was the case for Figures 4.21 and 4.22), a zoomed-in view of the finely sampled paths show that, due to the discrete nature of the problem (discrete voxel sizes and pixel sizes), the objective functions are not completely smooth. Figure 4.23 shows the values of $P1$ along the path, as well as two zoomed-in parts of P – one away from the minimum point and one at the minimum point.

Figure 4.24 shows the numerical derivatives of $P1$ along the path for values for h of 0.001mm, 0.01mm, 0.1mm and 0.5mm. By choosing a bigger value for h the noise on the derivative can be averaged out but, according to (4.7), the less accurate the approximation of the gradient will be. Figure 4.24 also shows (in red) the derivative of a smooth function that was fitted to the noisy objective function values. It can be seen that the smooth derivative is best approximated when the value of h is 0.1mm. This was also the case for the objective function $P2$.

The value for h is highly dependent on the smoothness of the objective function, which is influenced by the CT data from which the DRRs are

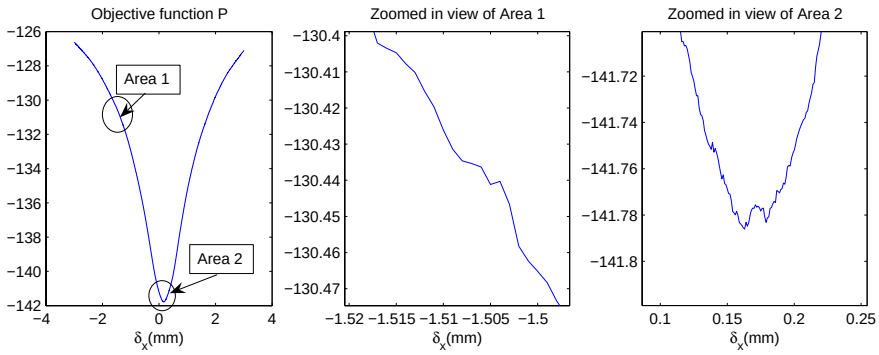


Figure 4.23: *Function values of P along the δ_x direction.*

generated, as well as the PR image quality. Therefore the value of h will not necessarily be the same for different CT data sets of different patients and different PR acquisition parameters.

The calculation of the derivative requires an additional function evaluation for each of the six parameters of $\mathbf{T}_{DRR}$.

4.3.3 Unit vector direction set (UVDS) method

This method sets the N dimensional direction vector $\mathbf{n}$ in which the line minimization is to be done to each of the unit vectors $\mathbf{e}_1$, $\mathbf{e}_2$,...,$\mathbf{e}_N$ in turn. Hence, the objective function will be minimized along the δ_x translation direction, then along the δ_y translation direction and so forth for all the translation directions and rotation directions up to the δ_ρ rotation direction. The process of minimizing over the whole set of unit vectors is repeated until some stopping criteria are met.

This method for setting $\mathbf{n}$ is very simple, but performs very well if the function behaves properly. According to [41, Section 10.2] this method does not perform well when a function of N dimensions has directions whose second derivatives are much larger in magnitude than some of the other directions. If this is the case, many cycles through all N unit vectors will be required in order to get anywhere.

As can be seen from Figure 4.21, when disregarding the out-of-plane translation δ_y, all the other directions for $P1$ are of a very similar form. The

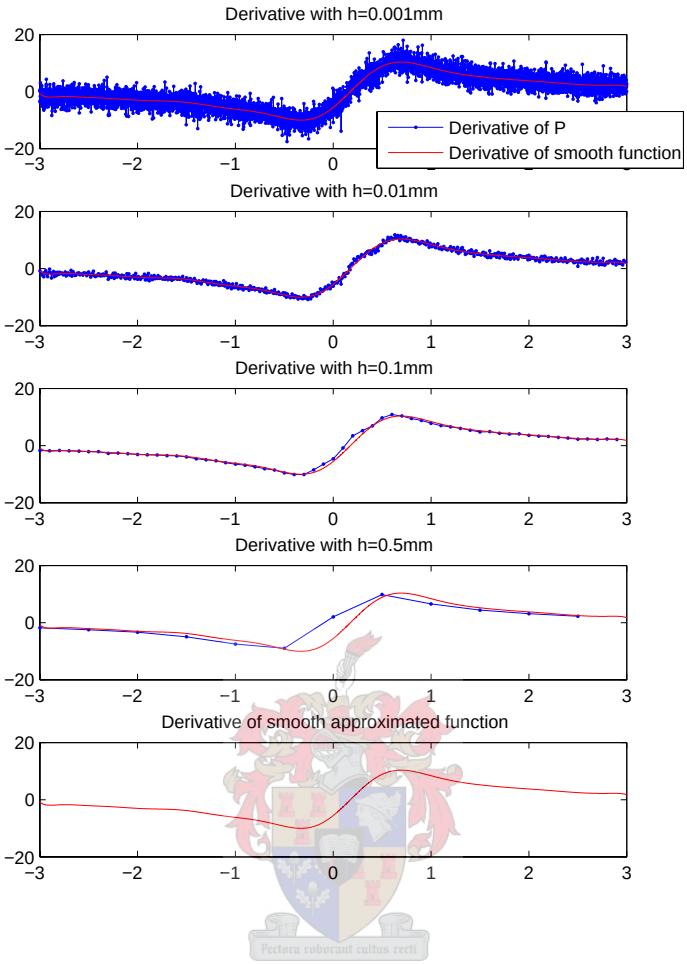


Figure 4.24: *Derivatives of the function values for different values of h.*

objective function $P1$ is therefore well behaved in terms of the requirements for this method when the out-of-plane translation δ_y is omitted and the unit vectors $\mathbf{e}_1$, $\mathbf{e}_2$,...,$\mathbf{e}_5$ are used.
Figure 4.22 shows that all the parameters of $P2$ are of a similar form, and therefore the UVDS method can also be used for this objective function by using the unit vectors $\mathbf{e}_1$, $\mathbf{e}_2$,...,$\mathbf{e}_6$.

To demonstrate the UVDS method's performance, a simulated PR of 1024^2 pixels was created by using (4.5) with a simulated error in patient position

of

$$\mathbf{T}_{simulated} = \begin{pmatrix} \delta_x \\ \delta_y \\ \delta_z \\ \delta_\theta \\ \delta_\phi \\ \delta_\rho \end{pmatrix} = \begin{pmatrix} 4.32\text{mm} \\ 4.50\text{mm} \\ 4.67\text{mm} \\ 4.12^\circ \\ 4.89^\circ \\ 4.23^\circ \end{pmatrix}.$$

The value of σ_{noise} was set to 0.0282, and the value of FSS used to construct K_{FS} to 1.5mm.

The method was first used to find the minimum of objective function $P1$ while including the out-of-plane translation δ_y. The algorithm was started at $\mathbf{p}_0 = \mathbf{0}$ and $\mathbf{n}$ was set to take on each of the unit vectors $\mathbf{e}_1$, $\mathbf{e}_2$,...,$\mathbf{e}_6$ in turn.

The Euclidian distance between two points ($E = \sqrt{(\mathbf{p}_k - \mathbf{p}_{k-1})^2}$) is recorded for each new point calculated by the algorithm. These distances are accumulated for every cycle of the whole set of directions, so that the total distance moved by the algorithm at the end of each whole set is known. If the total distance is below a certain limit the algorithm is assumed to have stabilized, and hence this is used as the stopping criteria for the algorithm. The limit of the total distance was set to 0.1.

Figure 4.25 shows the results of the minimization process. The left hand column of the figure shows how the value of each of the six parameters of $\mathbf{T}_{DRR}$ started from the initial value of 0 and progressed to the final value (which is indicated in the title of each subplot). The right hand column shows how the function value, the difference between the current and previous function values, the total error (which will be explained shortly), the Euclidian distance moved during the last iteration and the Euclidian distance moved during the last 5 iterations changed as the algorithm progressed. The title of the subplot at the top of the right hand column indicates the final function value, as well as the total number of function evaluations and the total number of line minimizations used.

The total error is calculated as the Euclidian distance between the current

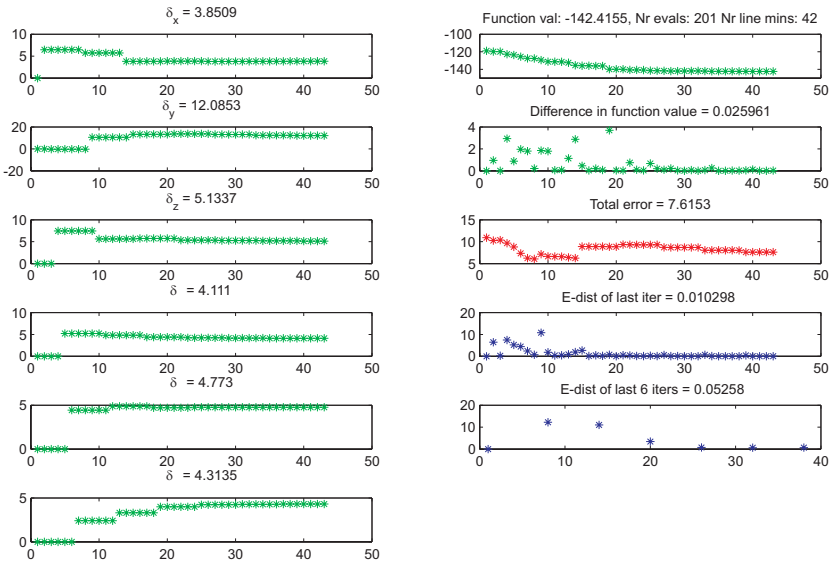


Figure 4.25: *Unit vector direction set minimization of P*1 *with out-of-plane translation included against number of line minimizations*

position and the true simulated error:

$$\text{Total error} = \sqrt{(\mathbf{p}_{\text{current}} - \mathbf{T}_{simulated})^2}. \tag{4.8}$$

The total error is not known during the verification process, since the true error in patient position is not known. By using a known simulated error, the accuracy of the minimization process can be measured. If all the parameters of $\mathbf{T}_{DRR}$ contribute equally to the total error, then the total error is given by Table 4.10. Since the out-of-plane translation are omitted in some cases, the corresponding total error is also given.

The total error that an algorithm ends with can be compared with Table 4.10 to measure how accurate the algorithm is, e.g. if the algorithm ends with a total error of 1.5 and δ_y were included, the average error in each direction is between 0.5mm (and 0.5°) and 1mm (and 1°). Although the chances are slim, it could happen that the error is purely in one direction, in which case it is 1.5mm (and 1.5°). The total error can therefore also be seen as the upper-bound of the error in any single direction.

Error in each direction	**Total error with δ_y included**	**Total error with δ_y excluded**
1mm (and 1°)	2.449	2.236
0.5mm (and 0.5°)	1.225	1.118
0.25mm (and 0.25°)	0.612	0.559

Table 4.10: Total error for equal error in each direction

The UVDS algorithm stopped when the total distance had a value of 0.05258 which is below 0.1. It can be seen how each of the parameters of $\mathbf{T}_{DRR}$ changed and stabilized as the number of line minimizations increased. The algorithm took 201 evaluations of $P1$ to stabilize at a point where the total error is a very high 7.6153. This can be mainly attributed to the out-of-plane translation δ_y which, at a value of 12.0853mm, is 7.5853mm from the correct value of 4.50mm. This is not surprising since in Section 4.2.6 it was observed that it might not be possible to find δ_y when using $P1$.
The 201 function evaluations in 42 line minimizations give an average of 4.7 function evaluations per line minimization. The time it takes to do 201 function evaluations of $P1$ is 815.28 seconds or 13.6 minutes.

Figure 4.26 shows the results of minimizing $P1$ when disregarding the out-of-plane translation δ_y. The unit vector set $\mathbf{e}_1$, $\mathbf{e}_2$,...,$\mathbf{e}_5$ was used. The algorithm stopped when the total distance moved during a whole set was 0.056145, which is below the limit of 0.1.
It took the algorithm 151 iterations through 30 line minimizations to reach the minimum value of -143.8747. This results in an average of 5 function evaluations per line minimization and a total time of 612.47 seconds or 10.2 minutes. It can be seen that the translation and rotation parameters have all stabilized at values which is very close to the simulated error values when the algorithm finished.
The minimum point calculated by the algorithm results in a total error of only 0.35807. According to Table 4.10 this translates to an average error of below 0.25mm (and 0.25°) in each direction.

The performance of the UVDS method was also measured when using the objective function $P2$. Figure 4.27 shows the results. From the discussion

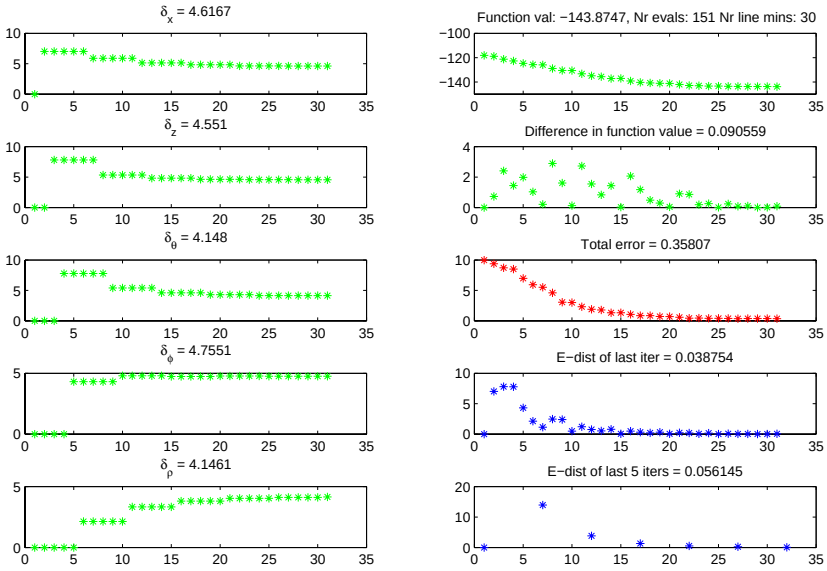


Figure 4.26: *Unit vector direction set minimization of $P1$ with out-of-plane translation excluded against number of line minimizations*

in Section 4.2.6 it was believed that when using $P2$ it might be possible to achieve much better accuracy than with $P1$, especially with regard to δ_y. The limit on the total distance moved during a whole set was therefore set to 0.01 in comparison with the 0.1 used for $P1$. The algorithm stopped when the total distance was 0.004575, at which point the total error was 1.2512. This is much better than the 7.6153 for $P1$ and results in an average error in each direction of just over 0.5mm (and 0.5°) according to Table 4.10. However, it took the algorithm 797 function evaluations in 90 line minimizations (an average of 8.9 function evaluations per line minimization) and over 3.2 hours. It can be reasoned that the stopping criteria were too strict, but even if the algorithm were stopped at a total distance of 0.1 (at about 40 line-minimizations) it would have taken about 360 function evaluations and over an hour.

The performance when using $P2$ when disregarding δ_y was also investigated. The algorithm was stopped when the total distance was 0.08547 which is below 0.1. At this stage the total error was 0.38295, which is very

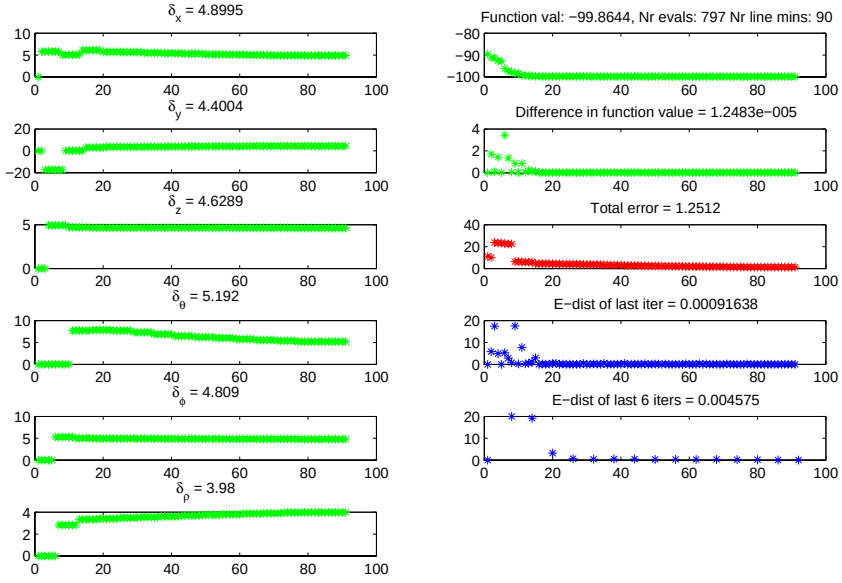


Figure 4.27: *Unit vector direction set minimization of $P2$ with out-of-plane translation included against number of line minimizations*

comparable with the value of 0.35807 when using $P1$ and disregarding δ_y. The algorithm took 203 function evaluations in 49 line minimizations (an average of 4.14 per line minimization) and 49.2 minutes.

Since there are no improvement in accuracy when using $P2$ over $P1$ when disregarding δ_y, and the improvement in accuracy of using $P2$ when including δ_y comes at a very high cost in computation time, the use of $P2$ is not practical.

4.3.4 Powell's direction set (PDS) method

For functions that are not well-behaved (e.g. functions that have long narrow valleys), a better set of directions is needed than the unit vectors. A set of directions is needed where minimization along the one direction is not "spoiled" by a minimization along a subsequent direction. Such "non-interfering" directions are conventionally called conjugate directions. Powell

first discovered a direction set method that produces N mutually conjugate directions [41, Section 10.5]. The algorithm is as follows:

1. Initialize the set of directions $\mathbf{n}_i$ to the unit vectors:

$$\mathbf{n}_i = \mathbf{e}_i \qquad i = 1, \ldots, N$$

2. Now repeat the following basic procedure until the stopping criteria are met:

   - Save starting position $\mathbf{p}_0$
   - For $i = 1, \ldots, N$, minimize $\mathbf{p}_{i-1}$ along $\mathbf{n}_i$ to get $\mathbf{p}_i$
   - For $i = 1, \ldots, N-1$, set $\mathbf{n}_i \leftarrow \mathbf{n}_{i+1}$
   - Set $\mathbf{n}_N \leftarrow \mathbf{p}_N - \mathbf{p}_0$
   - Minimize from $\mathbf{p}_N$ along $\mathbf{n}_N$ to get new $\mathbf{p}_0$

The problem with this algorithm is that, because at each iteration of the basic procedure $\mathbf{n}_1$ is replaced with $\mathbf{p}_N - \mathbf{p}_0$, it tends to produce directions that are linearly dependent. This results in the algorithm finding a minimum in only a subspace of the N-dimensional space. This problem is fixed by resetting the set of directions to the unit vectors after every N iterations of the basic procedure.

Pectora roborant cultus recti

To demonstrate the PDS method's performance, the same simulated PR was used as in the previous section with the UVDS method. Since the previous section showed that using the objective function $P2$ is not practical, the performance of Powell's method was only investigated for $P1$ and while excluding the out-of-plane translation δ_y.

From experience with Powell's method it was found that the limit on the total distance moved during a whole basic procedure should be set to 0.005. Figure 4.28 shows the results for the minimization. The algorithm was stopped when the total distance became 0.00346. The algorithm took 266 function evaluations in 60 line minimizations (an average of 4.4 evaluations per line minimization) and 17.98 minutes. At this point the algorithm had

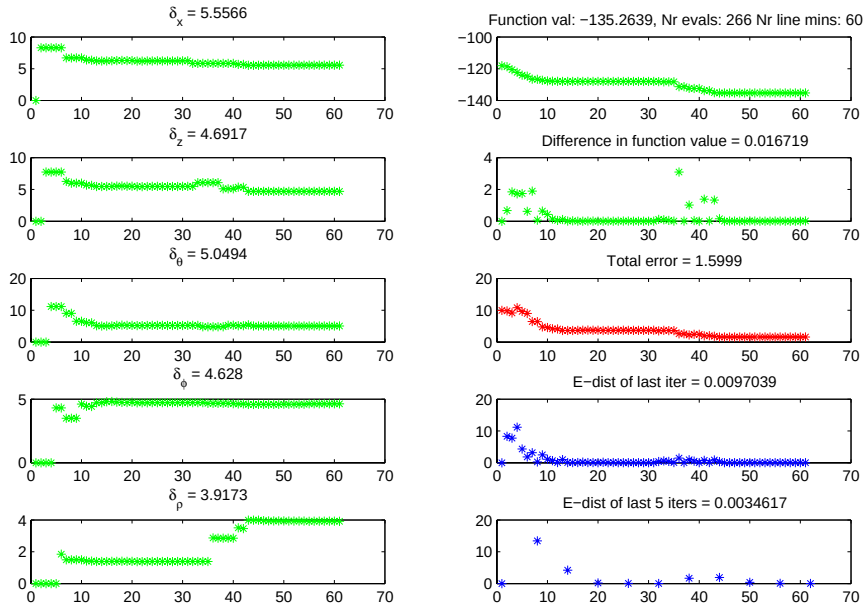


Figure 4.28: *PDS minimization of P1 with out-of-plane translation excluded against number of line minimizations*

a total error of 1.9975, which is much bigger than the 0.35807 of the UVDS method. According to Table 4.10 this total error results in an average error in each direction of between 0.5mm (and 0.5°) and 1mm (and 1°). It can be seen from Figure 4.28 that for the PDS method multiple parameters of $\mathbf{T}_{DRR}$ are changed during each line minimization, while for the unit vector method (see Figure 4.26) a single parameter is changed at a time. It is interesting to note how the algorithm started improving after about 30 line minimizations. This is when the set of directions were reset to the unit vector set.

The reason why the PDS method does not performs as well as the UVDS method is because of the linear dependency problem. As seen from the improvement after the resetting of the directions, the unit vector set is a very good set of directions for this problem. That is the reason why the UVDS performs so well. For objective functions that do not behave as well, the PDS method might outperform the UVDS method.

4.3.5 Steepest descent (SD) method

This method, together with the following two methods, are gradient based minimization methods. The idea behind these methods is that the gradient information can be used to reach the minima of the objective function with fewer line minimizations. The SD method uses the gradient $\boldsymbol{\nabla} P$ of the objective function P directly to find the minimum of P [41, Section 10.6]. The algorithm is as follows:

1. Start at some initial point $\mathbf{p}_0$
2. Move from the current point $\mathbf{p}_i$ to a new point $\mathbf{p}_{i+1}$ by minimizing along the local downhill gradient $-\boldsymbol{\nabla} P(\mathbf{p}_i)$
3. Repeat the step above until the stopping criteria are met.

The problem with the SD method is the same as for the UVDS method, i.e. it performs very poorly when the objective function has long narrow valleys or is ill-behaved in a similar way.

Figure 4.29 shows the minimization of the objective function $P1$ while excluding δ_y. The stopping criteria for the SD method is that the total distance moved during the last three iterations of step 2 be less than 0.005.
The algorithm was stopped when the total distance became 0.00475. It took the algorithm 141 function evaluations in 14 line minimizations (an average of 10.1 evaluations per line minimization) and 9.53 minutes to reach this point. The total error at this point is 0.21223, which is the lowest total error of the methods so far and results in an average error in each direction of less than 0.25mm (and 0.25°).
The 14 line minimizations used by the SD method is less than half the 30 used by the UVDS method. From this it is gathered that the gradient information indeed helps the algorithm to find the minimum in fewer line minimizations. The fact that the total number of 141 function evaluations used is only slightly less than the 151 function evaluations of the UVDS method is due to the extra five function evaluations needed to calculate the gradient for each line minimization.

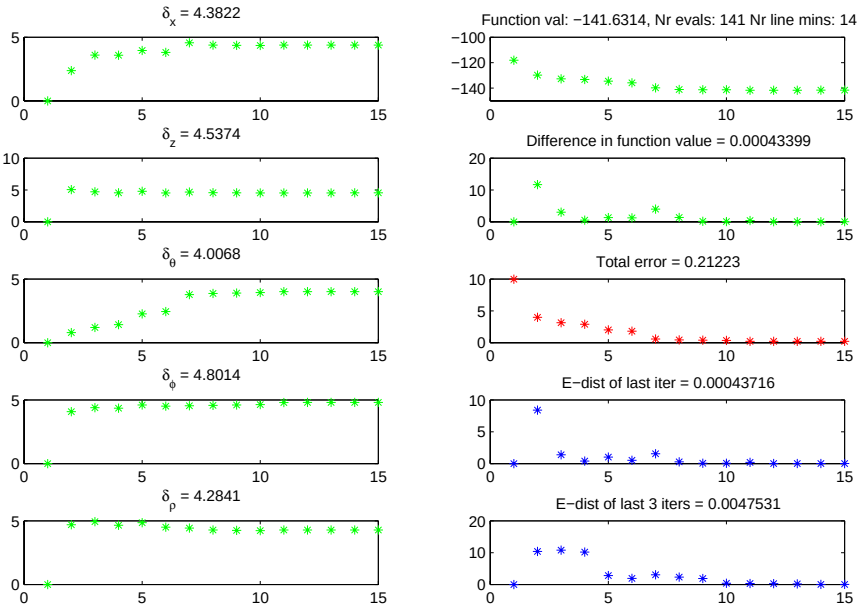


Figure 4.29: *SD minimization.*

4.3.6 FRPR conjugate gradient method

To overcome the problems of the SD gradient based method, the direction proceeded in from a new point should not be down the new gradient, but in a direction that is conjugate to the old gradient, and insofar possible, conjugate to all the previous directions followed [41, Section 10.6]. Such methods are called conjugate gradient methods.

The conjugate gradient method by Fletcher and Reeves is based on the notion that the objective function P at the point $\mathbf{p}$ is roughly approximated as the quadratic form

$$f(\mathbf{x}) \approx c - \mathbf{b}^T \cdot \mathbf{x} + \frac{1}{2}\mathbf{x}^T \cdot \mathbf{A} \cdot \mathbf{x} \tag{4.9}$$

where

$$c \equiv f(\mathbf{p}) \qquad \mathbf{b} \equiv -\boldsymbol{\nabla} f\big|_{\mathbf{p}} \qquad [\mathbf{A}]_{ij} \equiv \left.\frac{\partial^2 f}{\partial x_i \partial x_j}\right|_{\mathbf{p}}. \tag{4.10}$$

The gradient of this function is

$$\boldsymbol{\nabla} f = \mathbf{A} \cdot \mathbf{x} - \mathbf{b} \tag{4.11}$$

and the change in the gradient $\boldsymbol{\nabla} f$ when moving in some direction is

$$\delta(\boldsymbol{\nabla} f) = \mathbf{A} \cdot (\delta \mathbf{x}). \tag{4.12}$$

The condition that a new direction $\mathbf{n}_i$ must hold in order not to "spoil" the minimization along the previous direction $\mathbf{n}_{i-1}$, is that the change in gradient be perpendicular to the previous direction [41, Section 10.6]. Therefore

$$\mathbf{0} \;=\; \mathbf{n}_{i-1} \cdot \delta(\boldsymbol{\nabla} f) \;=\; \mathbf{n}_{i-1}^T \cdot \mathbf{A} \cdot \mathbf{n}_i. \tag{4.13}$$

The conjugate gradient method constructs two sequences of vectors by starting with an arbitrary initial vector $\mathbf{g}_0$ (and letting $\mathbf{h}_0 = \mathbf{g}_0$) and then using the recurrence

$$\mathbf{g}_{i+1} \;=\; \mathbf{g}_i - \lambda_i \mathbf{A} \cdot \mathbf{h}_i \qquad \mathbf{h}_{i+1} \;=\; \mathbf{g}_{i+1} - \gamma_i \mathbf{h}_i \qquad \text{for} \quad i = 0, 1, 2, \ldots \tag{4.14}$$

These vectors satisfy the orthogonality and conjugacy conditions

$$\mathbf{g}_i^T \cdot \mathbf{g}_j \;=\; 0 \qquad \mathbf{h}_i^T \cdot \mathbf{A} \cdot \mathbf{h}_j \;=\; 0 \qquad \mathbf{g}_i^T \cdot \mathbf{h}_j \;=\; 0 \qquad j < i \tag{4.15}$$

The scalars λ_i and γ_i are given by

$$\lambda_i \;=\; \frac{\mathbf{g}_i^T \cdot \mathbf{g}_i}{\mathbf{h}_i^T \cdot \mathbf{A} \cdot \mathbf{h}_i} \;=\; \frac{\mathbf{g}_i^T \cdot \mathbf{h}_i}{\mathbf{h}_i^T \cdot \mathbf{A} \cdot \mathbf{h}_i} \tag{4.16}$$

$$\gamma_i \;=\; \frac{\mathbf{g}_{i+1}^T \cdot \mathbf{g}_{i+1}}{\mathbf{g}_i^T \cdot \mathbf{g}_i} \tag{4.17}$$

As stated in [41, Section 10.6], a self-contained derivation of these equations is given in [43].

If the Hessian matrix $\mathbf{A}$ was known, then (4.14) could have been used to find successively conjugate directions $\mathbf{h}_i$ along which to line minimize. Since the Hessian matrix $\mathbf{A}$ is not known, another way must be found to generate the vectors $\mathbf{g}_i$ and $\mathbf{h}_i$. A theorem to achieve this from [41, Section 10.6] is now presented:

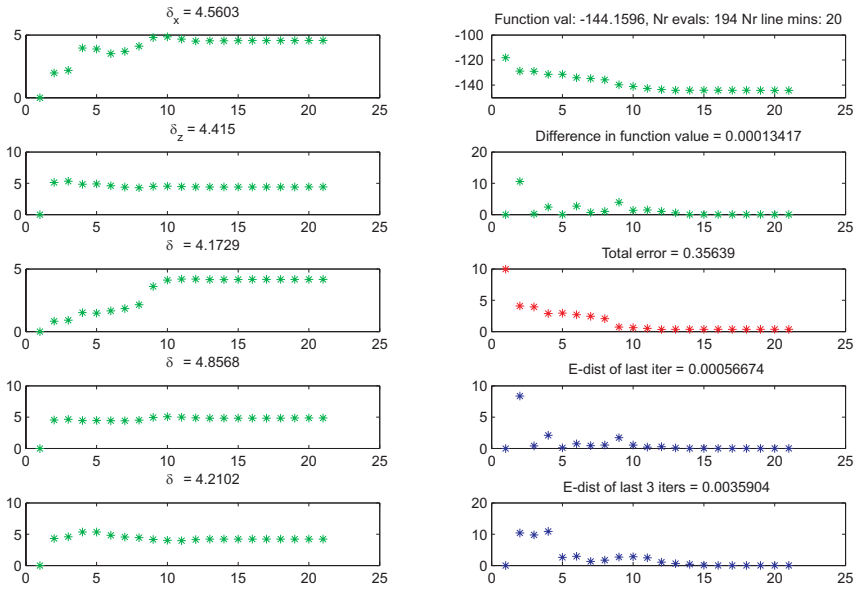


Figure 4.30: *FRPR minimization.*

If the gradient of the function f can be calculated so that $\boldsymbol{g}_i = \boldsymbol{\nabla} f(\boldsymbol{p}_i)$ for some point $\boldsymbol{p}_i$, and f (which is a quadratic form) is minimized along the direction $\boldsymbol{h}_i$ to obtain a new point $\boldsymbol{p}_{i+1}$, then the vector $\boldsymbol{g}_{i+1} = \boldsymbol{\nabla} f(\boldsymbol{p}_{i+1})$ is the same vector as would have been constructed by (4.14).

In this way a sequence of vectors $\mathbf{g}_i$ and $\mathbf{h}_i$ can be generated without the knowledge of $\mathbf{A}$. The initial vectors $\mathbf{g}_0$ and $\mathbf{h}_0$ are set to be the downhill gradient at $\mathbf{p}_0$.

The Polak-Ribiere variant of this algorithm introduces one tiny change in the Fletcher-Reeves algorithm by using

$$\gamma_i \;=\; \frac{(\mathbf{g}_{i+1} - \mathbf{g}_i)^T \cdot \mathbf{g}_{i+1}}{\mathbf{g}_i^T \cdot \mathbf{g}_i} \tag{4.18}$$

instead of (4.17). This change is believed to help the algorithm when the function is not exactly a quadratic form by resetting $\mathbf{h}$ to the local gradient when the algorithm runs out of steam [41, Section 10.6].

Figure 4.30 shows the minimization of $P1$ (while excluding δ_y) using the FRPR algorithm. The stopping criteria for the FRPR method is that the total distance moved during the last three iterations be less than 0.005.
The algorithm stopped after 194 function evaluations (using an average of 9.7 evaluations in 20 line minimizations in 13.11 minutes) at which point the total distance was 0.00359. The total error was 0.35639, which is higher than the 0.21223 of the SD method but comparable to the 0.35807 of the UVDS method. While using less line minimizations than the UVDS method, the total number of function evaluation are higher because of the extra evaluations required to calculate the gradient.

4.3.7 BFGS quasi-Newton method

The basic idea behind the quasi-Newton method, also called the variable metric method, is to build up a good approximation to the inverse Hessian matrix [41, Section 10.7]. Hence, a sequence of matrices $\mathbf{H}_i$ will be constructed with the property

$$\lim_{i\to\infty} \mathbf{H}_i = \mathbf{A}^{-1} \tag{4.19}$$

Near the current point $\mathbf{x}_i$, the function is given to the second order by

$$f(\mathbf{x}) \;=\; f(\mathbf{x}_i) \;+\; (\mathbf{x}-\mathbf{x}_i)^T \cdot \boldsymbol{\nabla} f(\mathbf{x}_i) \;+\; \frac{1}{2}(\mathbf{x}-\mathbf{x}_i)^T \cdot \mathbf{A} \cdot (\mathbf{x}-\mathbf{x}_i) \tag{4.20}$$

of which the gradient is

$$\boldsymbol{\nabla} f(\mathbf{x}) \;=\; \boldsymbol{\nabla} f(\mathbf{x}_i) \;+\; \mathbf{A} \cdot (\mathbf{x}-\mathbf{x}_i). \tag{4.21}$$

According to Newton's method, the function will have a minimum at $\boldsymbol{\nabla} f(\mathbf{x}) = \mathbf{0}$, and thus the next iteration point can be calculated as

$$\mathbf{x} - \mathbf{x}_i = -\mathbf{A}^{-1} \cdot \boldsymbol{\nabla} f(\mathbf{x}_i) \tag{4.22}$$

The left hand side is the finite Newton step that will take the algorithm to the exact minimum of the function when it is in quadratic form. This step is known once an accurate estimate of the Hessian is accumulated, i.e. $\mathbf{H} \approx$

$\mathbf{A}^{-1}$.

The BGFS method as taken from [41] for updating $\mathbf{H}$ is

$$\begin{aligned}\mathbf{H}_{i+1} &= \mathbf{H}_i + \\ &\frac{(\mathbf{x}_{i+1}-\mathbf{x}_i)\cdot(\mathbf{x}_{i+1}-\mathbf{x}_i)^T}{(\mathbf{x}_{i+1}-\mathbf{x}_i)^T\cdot(\boldsymbol{\nabla} f_{i+1}-\boldsymbol{\nabla} f_i)} - \\ &\frac{[\mathbf{H}_i\cdot(\boldsymbol{\nabla} f_{i+1}-\boldsymbol{\nabla} f_i)]\cdot[\mathbf{H}_i\cdot(\boldsymbol{\nabla} f_{i+1}-\boldsymbol{\nabla} f_i)]^T}{(\boldsymbol{\nabla} f_{i+1}-\boldsymbol{\nabla} f_i)^T\cdot\mathbf{H}_i\cdot(\boldsymbol{\nabla} f_{i+1}-\boldsymbol{\nabla} f_i)} + \\ &\left[(\boldsymbol{\nabla} f_{i+1}-\boldsymbol{\nabla} f_i)^T\cdot\mathbf{H}_i\cdot(\boldsymbol{\nabla} f_{i+1}-\boldsymbol{\nabla} f_i)\right]\mathbf{u}\cdot\mathbf{u}^T. \end{aligned} \quad (4.23)$$

The vector $\mathbf{u}$ is defined as

$$\begin{aligned}\mathbf{u} &\equiv \frac{(\mathbf{x}_{i+1}-\mathbf{x}_i)}{(\mathbf{x}_{i+1}-\mathbf{x}_i)^T\cdot(\boldsymbol{\nabla} f_{i+1}-\boldsymbol{\nabla} f_i)} - \\ &\frac{\mathbf{H}_i\cdot(\boldsymbol{\nabla} f_{i+1}-\boldsymbol{\nabla} f_i)}{(\boldsymbol{\nabla} f_{i+1}-\boldsymbol{\nabla} f_i)^T\cdot\mathbf{H}_i\cdot(\boldsymbol{\nabla} f_{i+1}-\boldsymbol{\nabla} f_i)}. \end{aligned} \quad (4.24)$$

The algorithm is started with $\mathbf{H}$ as the $N\times N$ identity matrix.
As stated when the calculation of λ was discussed in the beginning of this section, taking the full Newton step will not always result in a lower function value, e.g. when a too large step makes the quadratic approximation invalid. For this reason the new point is calculated by doing a line-minimization along the Newton direction.

Figure 4.31 shows the minimization of $P1$ (while excluding δ_y) using the BFGS method. The stopping criteria for the BFGS method is that the total distance moved during the last three iterations be less than 0.05.
The algorithm stopped after 204 function evaluations (an average of 9.7 evaluations in 21 line minimizations) and 13.79 minutes. The total error of 0.30136 is comparable with the total error of the SD and UVDS methods, while its total number of 204 function evaluations is higher than the 151 and 194 of the other two methods.

Table 4.11 gives a summary of the performance of each of the five methods using the simulated PR of Section 4.3.3. The table shows that the SD method using $P1$ performed the best of the five minimization methods. Not only has it the lowest total error of only 0.21223, but it also used the lowest

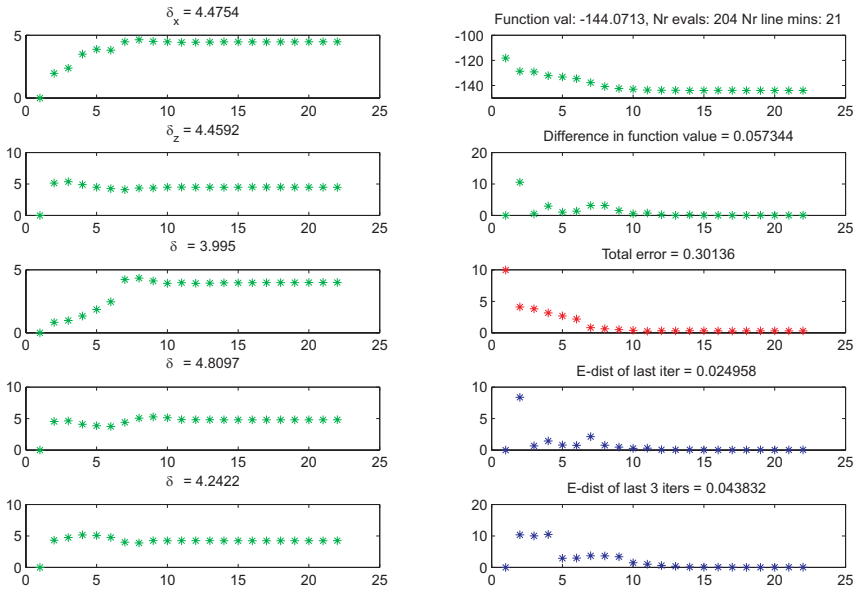


Figure 4.31: *BFGS minimization.*

number of function evaluations and therefore the least time.

The results showed in Table 4.11 depends on the particular simulated PR used, and does not say anything about the performance of the five minimization methods in general. It could be that for different simulated errors and different reference positions the methods perform differently. The investigation into the general performance of the methods is done in the next chapter.

Method	P	δ_y **used**	**Total error**	**P evals**	**Line mins**	**Ave evals per line min**	**Time used**
UVDS	$P1$	yes	7.6153	201	42	4.8	13.58 mins
UVDS	$P1$	no	0.35807	151	30	5.0	10.21 mins
UVDS	$P2$	yes	1.2512	797	90	8.86	193.36 mins
UVDS	$P2$	no	0.38295	203	49	4.14	49.2 mins
PDS	$P1$	no	1.9975	266	60	4.4	17.98 mins
SD	$P1$	no	0.21223	141	14	10.1	9.53 mins
FRPR	$P1$	no	0.35639	194	20	9.7	13.11 mins
BFGS	$P1$	no	0.30136	204	21	9.7	13.79 mins

Table 4.11: Performance of the five minimization methods

Chapter 5

Experiments and Results

In this chapter a number of experiments are conducted to measure the performance of the minimization methods. Table 4.11 in the previous chapter showed that the UVDS, the SD, the FRPR and the BFGS methods were all able to find the minimum of the objective function $P1$ (when disregarding δ_y), and that the total error of each of these methods results in an average error in each direction of below 0.25mm (and 0.25°) for the particular simulated PR used.

In Section 5.1 the general performance of the five minimization methods are measured by repeating the minimization process for many different simulated PRs. As will be seen from the results of these measurements, the UVDS is the only method that provides reliable results. In Section 5.2 the performance of the UVDS method for different reference positions is investigated.

Recently, the new DPRS developed for use by iThemba LABS's new patient positioning system was used to take PRs of a skull phantom. In Section 5.3 the performance of the UVDS method when using these real PRs is measured.

5.1 General performance of the five minimization methods

To investigate the general performance of the five minimization methods, all the methods were used to minimize $P1$ (while excluding the out-of-plane translation δ_y) constructed with 50 different simulated PRs. These PRs were all generated from the same CT data and with the same reference position, but with different simulated errors. The parameters of $\mathbf{T}_{simulated}$ were set to random numbers from the range -5mm to 5mm (for the translation parameters) and -5° to 5° (for the rotation parameters).

The starting point $\mathbf{p}_0$ for every method and all the simulated PRs was $\mathbf{0}$. A limit of 200 was set on the number of function evaluations that each method could use. This was done to limit the total time needed to conduct the experiment.

Figure 5.1 shows the result of the experiment. Each row of the figure shows the results for a specific method. For each method, the total error, function evaluations, line minimizations and time used (in seconds) are plotted for each of the 50 different simulated PRs. The average over all of the 50 values for each plot is stated in the title of each plot. A summary of the results is also given in Table 5.1. While the SD method performed the best for the particular simulated PR used in Section 4.3, its performance in general is not that good. It shows an average total error of 1.0119 with a maximum total error of almost 4. From the 50 simulated PRs, 16 has a total error of over the accuracy-limit of 1. The accuracy-limit is chosen as 1 to identify the misregistration of a PR as described in Section 2.7.3.

The FRPR and BFGS performed similarly. While the average total error of 0.54795 of the BFGS method is a bit better than the 0.71768 of the FRPR method, it also takes about 50 more function evaluations on average. The FRPR method could not register 5 of the 50 PRs and the BFGS 2 of the 50 PRs. The standard deviation in the total error of the two methods (0.339 for the FRPR and 0.231 for the BFGS) are also relatively close. This can also be seen from the plots of their total error values in Figure 5.1, which shows that the values of each plot reside, except for the couple of outliers,

in a band of roughly the same size.

The PDS method, as was the case with the minimization of the Section 4.3, performed the worst with an average total error of 1.3817 and 28 misregistrations. It shows a large standard deviation in its total error, which is clearly seen from the plot in Figure 5.1, which shows that most of its total error values reside between 0 and 3.

The UVDS method performed very well with all the 50 simulated PRs, and showed no misregistrations. It showed a very good average total error of 0.38652 with a standard deviation of only 0.070, and it can be seen in the plot of Figure 5.1 that the values reside in the very narrow band of 0.2 to 0.6. The maximum total error for a PR is only 0.5672 and the minimum 0.23941. While the accuracy of the UVDS is better than any of the other methods, the average time of 5.73 minutes needed to do a minimization is also the lowest of all the methods.

It is therefore concluded that the UVDS method is not only the most accurate minimization method for registering the simulated PRs, but also the fastest.

It is interesting to note that the average function evaluations per line minimization for the three gradient based methods (9.89, 9.97 and 9.55) are all higher than that for the non-gradient based methods (4.80 and 4.33) by about 5 evaluations. This is the 5 extra function evaluations required to calculate the gradient for each line minimization. This is also the reason why the gradient based methods use more function evaluations than the non-gradient based UVDS method despite the fact that they use fewer line minimizations.

5.2 Performance of the UVDS method with different reference positions

The previous section concluded that the UVDS method performs the best of the five minimizations methods when using simulated PRs that had 50 randomly different simulated errors around a specific reference position. To

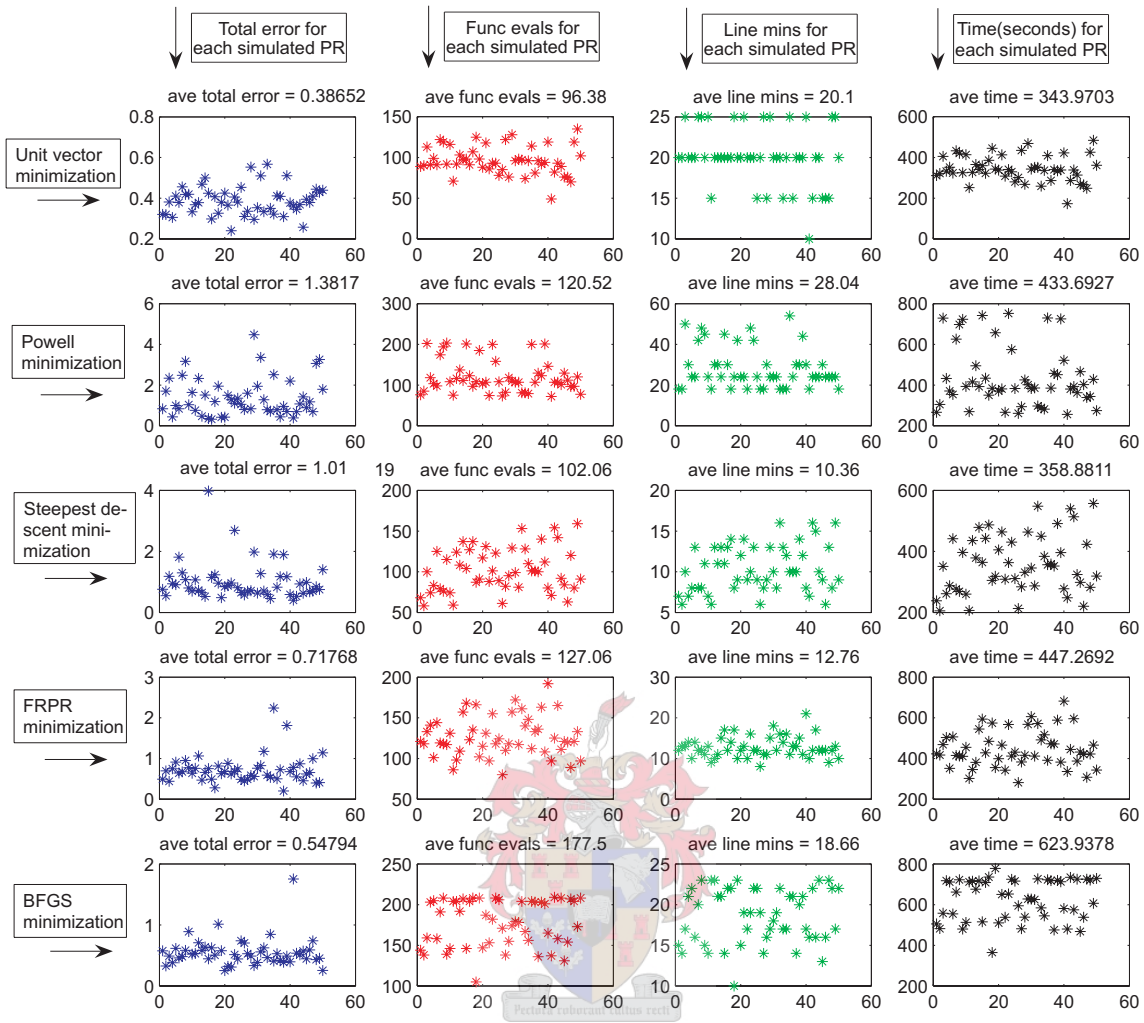


Figure 5.1: *Total error, function evaluations, line minimizations and time required for the 50 different simulated PRs when using the five minimization methods.*

find out if this good performance does not depend on the particular reference position, the UVDS was used to register 10 simulated PRs with randomly different simulated errors (in the ranges of -5mm to 5mm and -5° to 5°) around 11 different reference positions. Figure 5.2 shows the 11 different reference positions used. The PRs were once again used to construct the objective function $P1$ (while excluding the out-of-plane translation δ_y).
A summary of the results is given in Table 5.2. The average total error for all the reference positions are very close to each other, and while the maximum

Table 5.1: *General performance of the minimization methods.*

Method	Statistic	Total error	P evals	Line mins	Ave evals per line min	Time (minutes)	Total error > 1
UVDS	**Mean**	**0.38652**	**96.38**	**20.1**	**4.806**	**5.7328**	**0**
	Std dev	0.070001	17.2353	3.57	0.28281	1.0408	
	Max	0.5672	135	25	5.4667	8.069	
	Min	0.23941	49	10	4.25	2.8871	
PDS	**Mean**	**1.3817**	**120.52**	**28.04**	**4.3341**	**7.2282**	**28**
	Std dev	0.92553	40.2367	10.0223	0.31519	2.462	
	Max	4.4754	203	54	5.3889	12.5344	
	Min	0.31694	72	18	3.7037	4.2593	
SD	**Mean**	**1.0119**	**96.38**	**10.36**	**9.8996**	**5.9814**	**16**
	Std dev	0.62266	26.5986	2.8554	0.40078	1.5784	
	Max	3.9759	159	16	10.875	9.2893	
	Min	0.39676	58	6	9.1667	3.4171	
FRPR	**Mean**	**0.71768**	**127.06**	**12.76**	**9.9714**	**7.4545**	**5**
	Std dev	0.33922	24.6861	2.5759	0.38215	1.4948	
	Max	2.2396	192	21	10.8667	11.3711	
	Min	0.20019	80	8	9.0769	4.6843	
BFGS	**Mean**	**0.54794**	**177.5**	**18.66**	**9.5582**	**10.399**	**2**
	Std dev	0.23185	28.6037	3.3843	0.43644	1.6991	
	Max	1.7527	209	23	10.5	12.9616	
	Min	0.2525	105	10	8.5625	6.0719	

average total error of 0.41989 is for reference position 9 the average total error over all 110 values is 0.37252. This value is very similar to the 0.38652 for the experiment in the previous section, and the standard deviation of 0.08703 is also very similar to the 0.070001. The maximum total error over all the PRs is 0.61542 and occurred for reference position 4.

The average number of function evaluations needed for a reference position varied from 96 for reference position 6 to 132.4 for reference position 3. It is difficult to link this difference to the images of the two reference positions in Figure 5.2. The average number of function evaluations needed over all 110 PRs is 111.84, which translates to about 7.5 minutes.

This experiment shows that the UVDS method performs as well for other reference positions as for the reference position used in Section 5.1. The accuracy of the method and the number of function evaluations needed to complete the minimization process are not profoundly affected by using different reference positions, and there were no misregistrations for the whole

set of 110 PRs.

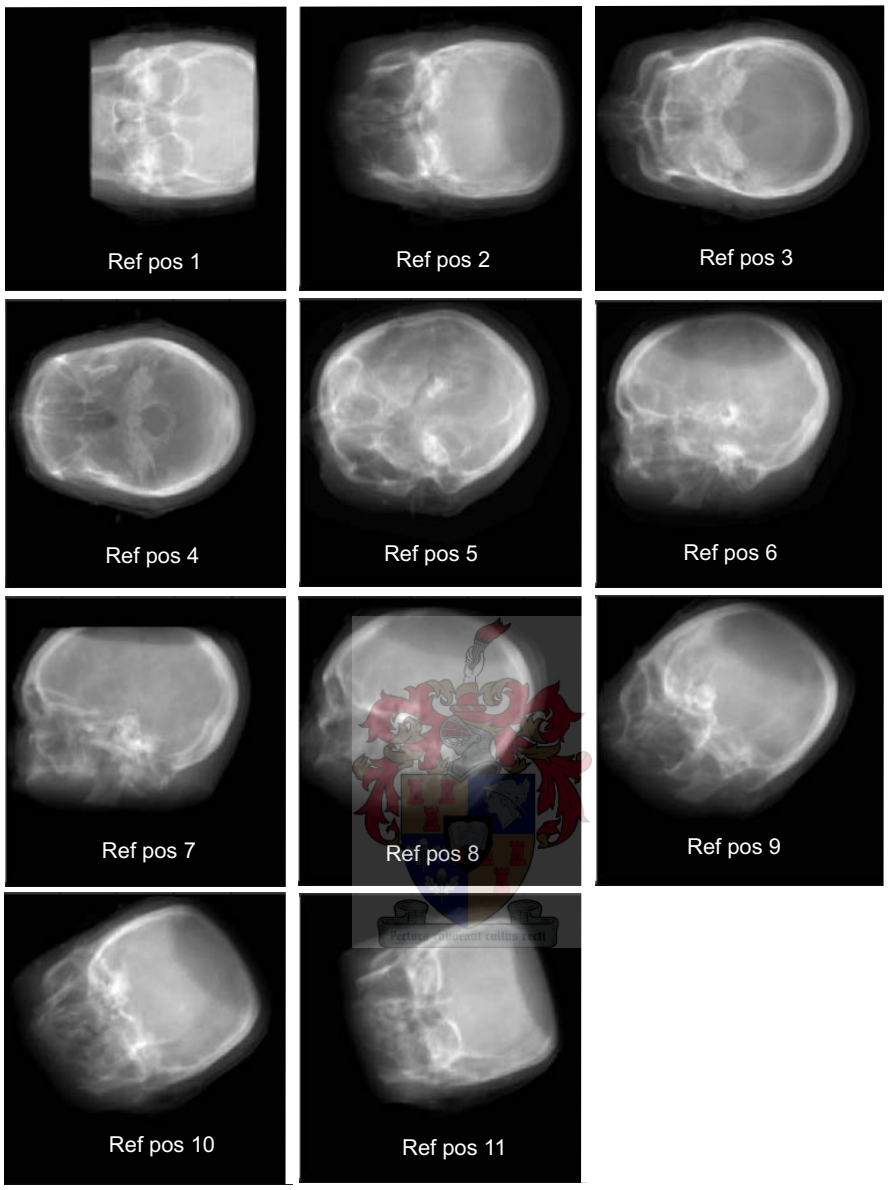


Figure 5.2: *Different reference positions.*

Table 5.2: *Performance of UVDS with different reference positions.*

Reference position	Total error mean	Total error std. dev.	Total error max	Total error min	Total error > 1	Evals mean
Ref pos 1	0.38219	0.069404	0.519	0.27425	0	122.1
Ref pos 2	0.40223	0.08865	0.57528	0.29022	0	106
Ref pos 3	0.3781	0.12299	0.50491	0.14451	0	132.4
Ref pos 4	0.40346	0.10811	0.61542	0.27851	0	112.8
Ref pos 5	0.35226	0.064707	0.48691	0.27893	0	112.5
Ref pos 6	0.33868	0.081119	0.49917	0.21752	0	96
Ref pos 7	0.36085	0.059342	0.45122	0.23586	0	114.3
Ref pos 8	0.35783	0.07369	0.50222	0.25852	0	109.8
Ref pos 9	0.41989	0.056109	0.52822	0.33967	0	107.2
Ref pos 10	0.34547	0.073194	0.46479	0.26331	0	104.6
Ref pos 11	0.35671	0.127	0.57401	0.17014	0	112.6
Total	**0.37252**	**0.08703**	**0.61542**	**0.14451**	**0**	**111.8455**

5.3 Performance of the UVDS method with real PRs

It was recently possible to acquire real PRs with the new DPRS to be used by iThemba LABS. This gave the opportunity to investigate how the methods developed in this thesis perform when using real data.

PRs were taken of a human skull phantom, which consists of a skull that is set in foam inside a perspex cube. The DPRS was set up in one of Ithemba LABS's treatment rooms with the X-ray tube and detector system in the beam line to get a setup as illustrated in Figure 5.3 (which shows a top view of the setup).

The treatment room is equipped with positioning lasers that indicates the treatment room isocenter. The cube containing the skull was placed on the treatment chair, and positioned with the aid of the lasers so that the center of the cube would be as close as possible to the treatment room isocenter. The sides of the cube were marked to correspond with the sides of the skull, i.e. right, left, posterior and anterior. The cube was positioned perpendicular to the central beam axis and with its right side facing the beam as shown in Figure 5.3. By positioning the cube in this way, the position of the skull is known to a certain degree.

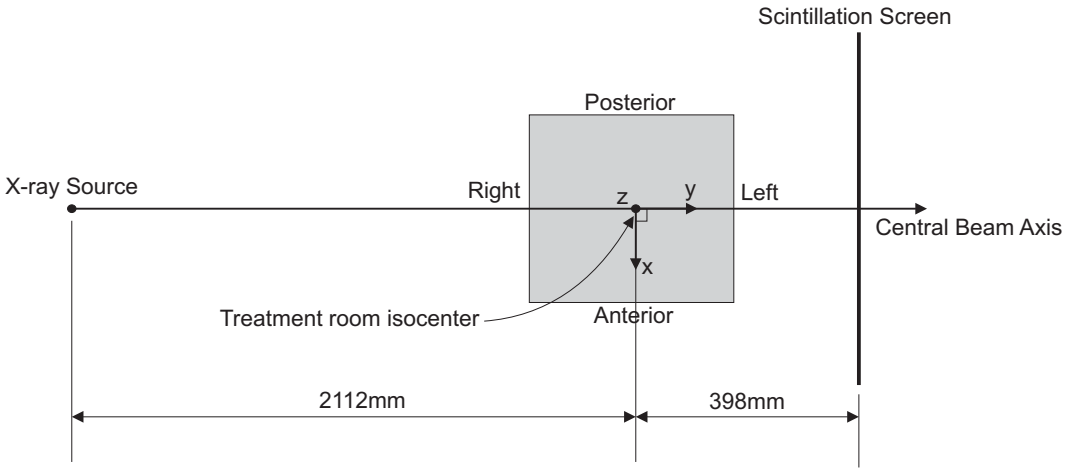


Figure 5.3: *Top view of the setup for the skull phantom cube. The x-, y- and z axes along which the cube is moved are also shown. The z axis comes vertically out of the page.*

The deviation between the current position of the cube from the ideal position is given by the transformation

$$\mathbf{T}_{error} = \mathbf{p}_{ideal} - \mathbf{p}_{current}$$

where $\mathbf{p}_{current}$ is the current position and the ideal position is given by

$$\mathbf{p}_{ideal} = \mathbf{0}.$$

As stated in Section 1.3, the transformation $\mathbf{T}_{error}$ can be estimated by the process of image registration. This is done by minimizing the objective function

$$P\left(\mathbf{T}_{DRR}\right) = -M\left(PR_{ref}, DRR\left(\mathbf{T}_{DRR}\right)\right) \tag{5.1}$$

where PR_{ref} is the reference PR taken of the cube in the current position. The value of $\mathbf{T}_{DRR}$ for which P is at a minimum is an approximation of the error transformation:

$$\mathbf{T}_{DRR} \approx \mathbf{T}_{error}.$$

The accuracy of the image registration process can be measured by $\mathbf{T}_{error} - \mathbf{T}_{DRR}$, or quantified (as was done in with the minimizations using simulated

PRs) by taking the Euclidian distance:

$$\text{Total error} = \sqrt{(\mathbf{T}_{error} - \mathbf{T}_{DRR})^2}$$

Because the true value of $\mathbf{p}_{current}$ and therefore also $\mathbf{T}_{error}$ is unknown, the accuracy of the image registration process can not be measured in this way. To get some measurement of the accuracy of the image registration process, it was decided to measure a relative change in position. This is possible because the treatment chair can be used to alter the position of the cube very accurately along the x-, y- and z translation axes and the θ rotation axis. Unfortunately, as stated in Section 1.1.1, the movement of the current treatment chair is limited and the cube can not be rotated around the ϕ- and ρ rotation axes with the setup used.

Five additional PRs were taken by first translating the cube by 5mm along each translation axis and 5° around the rotation axis, and then by simultaneously translating and rotating the cube along and around all the axes. A sixth PR was taken without the out-of-plane translation y. This was done because the results of Section 4.3 proved that it is very difficult to register the out-of-plane translation y. As stated in Section 2.3, translations and rotations of 5mm and 5° are extreme deviations, and in practice they are expected to be smaller. The five PRs and their corresponding positions are:

- PR_{δ_x} corresponding to $\mathbf{p}_{\delta_x} = \mathbf{p}_{ideal} + [\delta_x\ 0\ 0\ 0\ 0\ 0]$
- PR_{δ_y} corresponding to $\mathbf{p}_{\delta_y} = \mathbf{p}_{ideal} + [0\ \delta_y\ 0\ 0\ 0\ 0]$
- PR_{δ_z} corresponding to $\mathbf{p}_{\delta_z} = \mathbf{p}_{ideal} + [0\ 0\ \delta_z\ 0\ 0\ 0]$
- PR_{δ_θ} corresponding to $\mathbf{p}_{\delta_\theta} = \mathbf{p}_{ideal} + [0\ 0\ 0\ \delta_\theta\ 0\ 0]$
- $PR_{\delta_{all}}$ corresponding to $\mathbf{p}_{\delta_{all}} = \mathbf{p}_{ideal} + [\delta_x\ \delta_y\ \delta_z\ \delta_\theta\ 0\ 0]$
- $PR_{\delta_{all-no-y}}$ corresponding to $\mathbf{p}_{\delta_{all-no-y}} = \mathbf{p}_{ideal} + [\delta_x\ 0\ \delta_z\ \delta_\theta\ 0\ 0]$

All the PRs were acquired with an X-ray tube voltage of 90kV and an exposure of 25mAs. There was no binning of the CCD-camera's pixels, and the PRs therefore consisted of 1024^2 pixels. Because the facilities haven't

been developed yet, no flat field, bias field or dark field subtractions were done and only the raw intensity values obtained from the camera were used for the PR. Also, no correction for radial lens distortions was made.
The distances between the X-ray source and treatment room isocenter, and the treatment room isocenter and the scintillation screen were measured to be 2112mm and 398mm respectively as shown in Figure 5.3. It should be noted that the distance between the X-ray source and treatment room isocenter is not very accurate. This is because the distance to the exact location of the X-ray focal spot was not measured.

The DRRs to be used during the image registration process are generated from the CT data of the skull phantom. The CT data was acquired using a newly installed Philips Brilliance Big Bore CT scanner at iThemba LABS. A summary of the CT data is given in Table 5.3.
This CT data set is huge in comparison with the CT data set used for creating the DRRs used in all the previous investigations (a summary of which is given in Table 4.1). The filesize of this data set is also a very large 365MB (megabytes) in comparison with the 8.5MB of the smaller data set. The large size of the data set makes loading it and generating DRRs from it computationally very expensive.
A more manageable data set was created by downsampling each of the 309 CT image slices from 1024^2- to 512^2 pixels by using linear interpolation as described in Section 3.4.1.
No calibration data has yet been available for the new CT scanner, and therefore no XACC or BHCC are available for use with the CT data of the skull phantom. The DRRs are generated with 256^2 pixels and then interpolated to 1024^2 before they are compared with the 1024^2 pixel PRs with the MI similarity measure.

The position $\mathbf{p}_{current}$ was estimated by minimizing the objective function P given in (5.1) using the UVDS method. Because nothing was known about the robustness of the UVDS method when using real PRs, the objective function was minimized a number of times. For each of these minimizations the six parameters of the starting point of the UVDS algorithm were chosen to be a random numbers from the range -5mm to 5mm (for the translation

Table 5.3: *Summary of CT data.*

Field-of-view size	369.00mm×369.00mm×309mm
CT slice dimensions	1024×1024
CT slice pixel size	0.360mm^2
Number of slices	309
Slice separation	1mm
Slice thickness	1mm

parameters) and -5° to -5° (for the rotation parameters).

Figure 5.4 shows the results of the minimization process repeated ten times. The six sub-plots show the ten final values of each of the six components of $\mathbf{p}_{current}$ after the minimizations. It can be seen that all the minimizations (each of them started from a different point) stabilized at final positions which are very close to each other, which means that the UVDS method performs very consistently with the real PRs. The standard deviation of δ_x, δ_z, δ_θ, δ_ϕ and δ_ρ are very low, and the standard deviation of 1.5196 is acceptable for the out-of-plane translation δ_y.

The value of each of the components of $\mathbf{p}_{current}$ is set to the average of their ten estimated values. The current position is used as the reference position when measuring the relative change in position. Hence, the reference position is given by

$$\mathbf{p}_{reference} = \mathbf{p}_{current} = \begin{pmatrix} -3.1776 \\ 17.5708 \\ -0.0047 \\ -0.0238 \\ -0.1141 \\ -1.0022 \end{pmatrix}. \tag{5.2}$$

The value of 17.5708mm for δ_y is much higher than the other components of $\mathbf{p}_{current}$. This is because of the inaccurate measurement of the the distance between the X-ray focal spot and the treatment room isocenter. A more accurate value for this distance would therefore seem to be about 2130mm. Since only relative changes in the cube's position are important during the

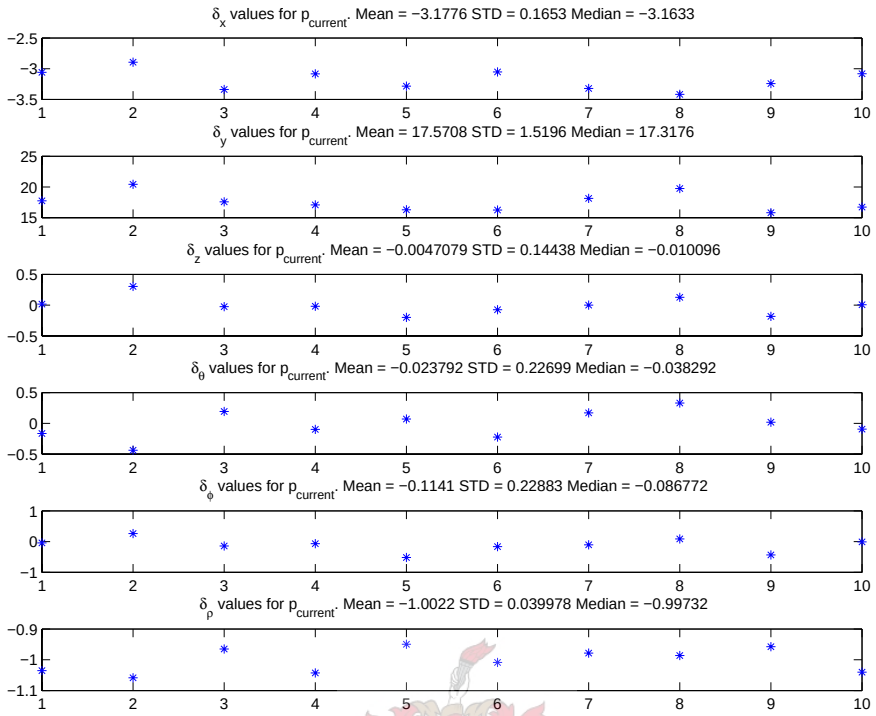


Figure 5.4: *Results of repeated minimization. The final values for each component of* $\boldsymbol{p}_{current}$ *are very consistent with low standard deviations.*

experiments, this high value for δ_y does not have any effect.

To measure the relative change in position, the positions of each of the additional six PRs were estimated by exchanging PR_{ref} in turn with PR_{δ_x}, PR_{δ_y}, PR_{δ_z}, PR_{δ_θ}, $PR_{\delta_{all}}$ and $PR_{\delta_{all-no-y}}$ in (5.1) and then minimizing P. Table 5.4 shows the results of the minimizations. The estimations of $\mathbf{p}_{\delta_x}$, $\mathbf{p}_{\delta_y}$, $\mathbf{p}_{\delta_z}$, $\mathbf{p}_{\delta_{rot}}$, $\mathbf{p}_{\delta_{all}}$ and $\mathbf{p}_{\delta_{all-no-y}}$ which correspond to the positions of the six additional PRs are given. The estimated relative change in position, calculated as the difference between the estimated position and the position of the reference PR is also given, with the particular component(s) under investigation in bold. The total error is calculated as the Euclidian distance between the estimated relative change in position and the real relative change in position specified during the acquisition of the PRs. The

number of function evaluations and line minimizations used by the UVDS method for each minimization are also shown.

The δ_x value of 5.0514mm for $\mathbf{p}_{\delta_x}$ is very close to the real value of 5mm, and the values of all the other components are also close to their real values of 0mm and 0° respectively, except for the out-of-plane translation δ_y which has a value of -0.9752mm. This is reflected in the value of the total error, which is higher than the total error with δ_y excluded. However, an accuracy of within 1mm for out-of-plane translation is considered very good. The image registration process is therefore very good in detecting any horizontal translations in the setup of the skull phantom.

The out-of-plane translation δ_y value of 4.830mm for $\mathbf{p}_{\delta_y}$ is very close to the real value of 5mm, and the values of all the other components are also close to the real value of 0. This results in very low values for the total error and the total error with δ_y excluded. The image registration process is therefore very good in detecting translations in the setup of the skull phantom along the beam line.

The δ_z value of 4.8932mm for $\mathbf{p}_{\delta_z}$ is very close to the real value of 5mm, and the values of $\mathbf{p}_{\delta_z}$'s other components are also close to their real values of of 0mm and 0°, except again for the out-of-plane translation δ_y which has a value of -1.0605mm. This is again reflected in the values of the total error and the total error with δ_y excluded which have values of 1.0875 and 0.2407 respectively. The image registration process is therefore very good in detecting any vertical translations of the skull phantom.

The value of 4.6467° for the rotation parameter δ_θ of $\mathbf{p}_{\delta_{rot}}$ is close to the real value of 5°, but the values of -1.3966mm and 1.6184mm for the translation parameters δ_x and δ_y are not as close to their real values of 0mm each. The resulting total error of 2.1765 and 1.4553 (when excluding δ_y) is accordingly also much higher compared to those for $\mathbf{p}_{\delta_x}$, $\mathbf{p}_{\delta_y}$ and $\mathbf{p}_{\delta_z}$. The image registration process is therefore able to detect rotations of the skull phantom around the z-axis, but with a resulting total error that is larger than for the detection of pure translations.

The δ_x, δ_y, δ_z and δ_θ values of 3.8708mm, 2.2643mm, 4.4934mm and 4.5489° respectively for $\mathbf{p}_{\delta_{all}}$ shows that the image registration process are able to de-

tect when the skull phantom is simultaneously translated and rotated along and around the particular axes. The accuracy of the out-of-plane translation is the worst, with a difference of 2.7357mm from the correct value of 5mm. The accuracy of δ_x is a bit lower than that of δ_z and δ_θ, with a difference of 1.1292mm compared to 0.0566mm and 0.4511°.
The results for $\mathbf{p}_{\delta_{all-no-y}}$ shows that the accuracy of the image registration process was not improved when excluding the out-of-plane translation δ_y, which was the case for the simulated PRs. The total error of $\mathbf{p}_{\delta_{all-no-y}}$ is in fact a bit higher than the total error of 1.2323 for the same components when including δ_y. The 407 function evaluations used when including δ_y is, however, much higher than the 163 used without δ_y.

The results of the experiments with the real PRs of the skull phantom show that the image registration process is very accurate in detecting relative translational changes in the position of the phantom. The detection of relative rotational changes and changes which combine relative translations and rotations are not as accurate as for pure translational changes, but still of acceptable accuracy.

Table 5.4: *Results of relative change in position.*

	δ_x	δ_y	δ_z	δ_θ	δ_ϕ	δ_ρ
$\mathbf{p}_{\delta_x}$	-8.2290	18.5460	0.0580	-0.0506	-0.1482	-0.9956
$\mathbf{p}_{\delta_x}$ - $\mathbf{p}_{ref}$	**5.0514**	-0.9752	-0.0627	0.0268	0.0341	-0.0066
Total error		0.9795	**Total error excluding** δ_y			0.0922
Evaluations		171	**Line minimizations**			42

	δ_x	δ_y	δ_z	δ_θ	δ_ϕ	δ_ρ
$\mathbf{p}_{\delta_y}$	-3.0493	22.4007	0.0513	-0.2060	-0.1444	-1.0527
$\mathbf{p}_{\delta_y}$ - $\mathbf{p}_{ref}$	0.1283	**4.8300**	0.0560	-0.1823	-0.0303	-0.0504
Total error		0.2919	**Total error excluding** δ_y			0.2373
Evaluations		95	**Line minimizations**			24

	δ_x	δ_y	δ_z	δ_θ	δ_ϕ	δ_ρ
$\mathbf{p}_{\delta_z}$	-3.0859	18.6312	-4.8979	0.1427	-0.0391	-1.0712
$\mathbf{p}_{\delta_z}$ - $\mathbf{p}_{ref}$	-0.0918	-1.0605	**4.8932**	-0.1665	-0.0750	0.0689
Total error		1.0875	**Total error excluding** δ_y			0.2407
Evaluations		426	**Line minimizations**			78

	δ_x	δ_y	δ_z	δ_θ	δ_ϕ	δ_ρ
$\mathbf{p}_{\delta_{rot}}$	-1.7811	15.9524	0.1115	-4.6714	0.0562	-0.9901
$\mathbf{p}_{\delta_{rot}}$ - $\mathbf{p}_{ref}$	-1.3966	1.6184	-0.1163	**4.6467**	-0.1703	-0.0121
Total error		2.1765	**Total error excluding** δ_y			1.4553
Evaluations		204	**Line minimizations**			48

	δ_x	δ_y	δ_z	δ_θ	δ_ϕ	δ_ρ
$\mathbf{p}_{\delta_{all}}$	-7.0485	19.8351	-4.9481	-4.5727	0.0775	-1.0111
$\mathbf{p}_{\delta_{all}}$ - $\mathbf{p}_{ref}$	**3.8708**	**2.2643**	**4.9434**	**4.5489**	-0.1916	0.0088
Total error		3.0004	**Total error excluding** δ_y			1.2323
Evaluations		407	**Line minimizations**			78

	δ_x	δ_y	δ_z	δ_θ	δ_ϕ	δ_ρ
$\mathbf{p}_{\delta_{all-no-y}}$	-6.7320	17.5708	-4.7490	-4.7851	0.0985	-0.9900
$\mathbf{p}_{\delta_{all-no-y}}$ - $\mathbf{p}_{ref}$	**3.5544**	0	**4.7443**	**4.7613**	-0.2126	-0.0122
Total error		1.5025				
Evaluations		163	**Line minimizations**			35

Chapter 6

Conclusions

This thesis investigated the various components required for the verification of the patient position during proton therapy.

The verification can be done by registering a PR of the patient in the current position with DRRs generated of the patient's CT data. The registration process finds a rigid body transformation which estimates the difference between the current position and the correct position by minimizing the measure which compares the two images. The registration process therefore consists of four main components: the DRR, the PR, the measure for comparing the two images and the minimization method.

The ray-tracing algorithm by Jacobs was successfully implemented to generate DRRs that is accurate in a reasonable amount of time. Each DRR is generated according to the rigid body transformation $\mathbf{T}_{DRR}$, which specifies the DRR's particular view.
The ray-tracing algorithm was implemented to include the option to use the appropriate XACC and BHCC to generate DRRs that better approximate the PRs acquired with the DPRS to be used by iThemba LABS.
The ray-sums calculated with the implementation includes attenuation of the pieces of air before and after the CT cube.
The implementation is able to use the VOI data that may accompany the CT data of a patient, in which case it is also able to check for rays that suffer from incomplete CT data.

The implementation enables the user to select a ROI from the PR for which the DRRs should be calculated with the use of simple polygon based drawing tools. This is the only user input required to generate a DRR and is only required to eliminate unwanted structures from the PR. This process is very simple, can be completed in a matter of seconds and does not require specialized skills from the user.

The image quality of the DRRs was measured by calculation of the MTF. The MTF was used to investigate the effect that the interpolation of the CT data and the DRRs has on the image quality of the DRRs.
It was found that the up-sampling and down-sampling of the CT data have a negative impact on the image quality of the DRRs. The effect that the interpolation of the DRRs has on the image quality was found to be very small.

Objective functions were formed by using the MI and CC similarity measures to compare DRRs and simulated PRs. Simulated PRs were used because real PRs were not readily available from the DPRS system to be used by iThemba LABS. The objective functions have six degrees of freedom corresponding to the three translation and three rotation parameters of $\mathbf{T}_{DRR}$ for which the DRR is created. The objective functions were investigated by creating similarity curves from various one dimensional paths through the six dimensional spaces of the objective functions.
It was found that:

- The objective functions formed by using CC are very accurate when comparing 1024^2 pixel DRRs with 1024^2 pixel PRs, but are very expensive to evaluate.
- The objective functions formed by using MI are of the same accuracy when using 256^2 pixel DRRs interpolated to 1024^2 pixels as when using 1024^2 pixel DRRs when comparing both with 1024^2 pixel PRs.
- The accuracy of the objective functions is very negatively affected when generating DRRs of CT data with thick slices, and that the accuracy of the objective function can be improved by up-sampling

the CT data to have thinner slices (due to the smoothing effect it has on the DRRs).

- The use of the XACC increases the accuracy of the objective functions using MI, but has little effect on the objective functions using CC. The use of the BHCC has a very small impact on the objective functions using either MI or CC.

- The objective functions using MI have lower accuracy for PRs that contain more noise, while those using CC are not affected significantly. The effect of different focal spot sizes is very small for both the objective functions using MI and CC.

Five minimization algorithms based on the line-search strategy were implemented. They are the unit vector direction set method, Powell's direction set method, the steepest descent method, the Fletcher-Reeves-Polak-Ribiere conjugate gradient method and the Broyden-Fletcher-Goldfarb-Shanno quasi-Newton method. The latter three are gradient based minimization methods. It was found that because the objective functions perform very similarly for each of the six parameters of $\mathbf{T}_{DRR}$, the UVDS method performs very well. The gradient based minimization methods are hampered by the calculation of the gradient, which is difficult to calculate accurately because of noise (caused by the discrete data used to generate the DRRs) on the objective functions.

The use of the objective functions using CC and 1024^2 pixel DRRs was found not to be practical, since they do not improve the accuracy of the minimization significantly and are computationally very expensive.

It was found that the out-of-plane translation is very difficult to register when using the simulated PRs, and it was therefore excluded from the minimization process.

The UVDS method was able to minimize the objective functions (using MI and excluding the out-of-plane translation) in an average of 111 function evaluations to achieve an average error in each parameter of $\mathbf{T}_{DRR}$ that is less than 0.25mm (for the translation parameters) and 0.25° (for the rotation parameters).

The UVDS method was also very robust, with zero of the 110 minimizations having an error above the accuracy-limit.
With the current DRR implementation, verification using the UVDS method takes about 7.5 minutes.

iThemba LABS is currently developing faster DRR implementations. If DRRs can be generated in 0.5 seconds (which is realistic for sophisticated DRR algorithms), the image registration process developed in this thesis can be used to verify patient position in under a minute. No user input is required during the minimization process, hence the only human intervention required by the image registration process is the selection of a ROI from the PR.
The result is therefore a verification process that is accurate, fast, robust and automatic.

It was recently possible to acquire PRs with the DPRS to be used by iThemba LABS. PRs were taken of a human skull phantom. These PRs were registered with DRRs generated of the CT data of the phantom. It was investigated whether the performance of the registration process when using the real PRs is comparable to the good performance achieved with the simulated PRs.
Because the exact position of the phantom was not known, the registration process was used to measure relative changes in position, since the phantom could be accurately translated (in all three directions) and rotated (only in one out-of-plane direction) using the treatment chair.
It was found that pure relative translations in all three directions could be registered with the same accuracy as the simulated PRs. The pure relative rotation and combined relative translations and rotation were less accurate, but still of acceptable accuracy.
These first experiments on using real PRs indicates that the behaviour of the registration process is not significantly different than when using the simulated PRs.

Future work

No flat field, dark field or bias field or radial lens distortion corrections were made for the real PRs. The use of the XACC and BHCC when generating the DRRs was also not possible, because no calibration data was available for the CT scanner used to acquire the CT data of the phantom. It still needs to be investigated whether the corrections on the PRs and the use of the XACC and BHCC can improve the accuracy of the registration process for the real PRs.

The effect that deviations in the other two rotation parameters have on the accuracy of the registration process needs to be investigated. This, together with a wider range of positions for the skull phantom can be investigated when the new patient positioning system of iThemba LABS is operational. Accurate values for $\mathbf{p}_{current}$ can be obtained with this system, which eliminates the need for measuring relative changes in position.

The registration of PRs obtained from objects other than the skull phantom need to be investigated.

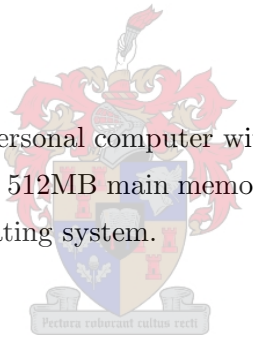
Appendix A

System Specifications

The hardware and software specifications of the system used for all the investigations conducted in this thesis are as follows:

Hardware specifications

The software were run on a personal computer with a 3.4GHz (Giga Hertz) Intel Pentium 4 processor and 512MB main memory. The software were run under the Windows XP operating system.



Software specifications

All the software for conducting the investigations in this thesis were written in Matlab version 7 (release 14). The ray-tracing algorithm was implemented in C and compiled to a MEX file (shared-library) for used by the Matlab programs.

Bibliography

[1] D. T. L. Jones. Proton Therapy: The promise of precision. In *International Radiation Physics Society Bulletin: Member's Paper*, volume 15(3/4), 2001.

[2] R. Wilks. *Principles of radiological physics*. Churchill Livingstone, New York, 1981.

[3] R. S. Brown, K. J. Lankester, M. McCormack, D. A. Power, and M. F. Spittle. Radiotherapy for perianal Paget's disease. *Clinical Oncology (Royal College of Radiologists)*, 14(4):272–284, 2002.

[4] R. R. Wilson. Radiological use of fast protons. *Radiology*, 47:487–491, 1946.

[5] D. T. L. Jones, A. N. Schreuder, E. A. de Kock, J. E. Symons, S. Maage, S. Bakhane, S. Schroeder, and A. O'Ryan-Blair. Proton therapy at iThemba LABS. *Radiation Physics and Chemistry*, 71:983–984, 2004.

[6] E. A. de Kock. Concepts and definitions required for the development of a portal radiographic verification system at iThemba LABS. Technical report, iThemba LABS, 2004.

[7] E. Latti. Development of a digital X-ray-imaging system at the National Acceleration Centre, December 2000. Masters Thesis–University of Stellenbosch.

[8] S. Clippe, D. Sarrut, C. Malet, S. Miguet, C. Ginestet, and C. Carrie. Patient setup error measurement using 3D intensity-based image

registration techniques. *Int. J. Radiation Oncology Biol. Phys*, 56(1):259–265, 2003.

[9] D. Sarrut and S. Clippe. Fast DRR generation for intensity-based 2D/3D image registration in radiotherapy. Technical report, LIRIS, Layon, 2003.

[10] D. J. Hawkes. Registration methodology - Introduction. In J. V. Hajnal, D. L. G. Hill, and D. J. Hawkes, editors, *Medical Image Registration*, pages 11–38. CRC Press, 2001.

[11] M. Baumann. Fast DRR generation with light fields, September 2004. Diploma Thesis (DEA DISIC).

[12] C. Sundgren. Estimation of patient setup errors in radiation therapy using portal imaging. Technical report, RaySeach Laboratories, 2004.

[13] R. L. Siddon. Fast calculation of the exact radiological path for a three-dimensional CT array. *Med.Phys.*, 12(2):252–255, Mar/Apr 1985.

[14] R. P. Woods. Spatial transformation models. In I. N. Bankman, editor, *Handbook of Medical Imaging Processing and Analysis*, pages 465–490. Academic Press, San Diego, CA, 2000.

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

[16] W. Wein. Intensity based rigid 2D-3D registration algorithms for radiation therapy, December 2003. Diplomarbeit Thesis (Technische Universitat Munchen).

[17] Jantine Frahn, Evan de Kock, and Neil Muller. Comparison of X-Ray image registration techniques across rotation and translation. In Fred Nichols, editor, *Proceedings of the Sixteenth Annual Symposium of the Pattern Recognition Association of South Africa*. Pattern Recognition Association of South Africa, 2005.

[18] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer-Verlag, New York, 1999.

[19] F. Jacobs, E. Sundermann, B. De Sutter, M. Christiaens, and I. Lemahieu. A fast algorithm to calculate the exact radiological path through a pixel or voxel space. *Journal of computing and information technology*, 6(1):89–94, 1998.

[20] Leendert van der Bijl, Evan de Kock, and Neil Muller. Influence of different portal radiograph reconstruction methodologies on the accuracy and efficiency of automatic treatment setup verification during conformal radiotherapy. In Fred Nichols, editor, *Proceedings of the Sixteenth Annual Symposium of the Pattern Recognition Association of South Africa*. Pattern Recognition Association of South Africa, 2005.

[21] E. A. de Kock. CT calibration curves for proton radiotherapy planning. Technical report, iThemba LABS, 2003.

[22] D. F. Jackson and D. J. Hawkes. X-ray attenuation coefficients of elements and mixtures. *Physics Report*, 70(3):169–233, 1981.

[23] D. M. Tucker. Semi-emperical model for generating tungsten target X-ray spectra. *Med.Phys.*, 18(2):211–218, 1991.

[24] L. Dong and A. L. Boyer. An image correlation procedure for digitally reconstructed radiographs and electronic portal images. *Int. J. Radiation Oncology Biol. Phys.*, 33(5):1053–1060, 1995.

[25] G. T. Herman. Correction for beam hardening in computed tomography. *Phys.Med.Biol.*, 24(1):81–106, 1979.

[26] D. Kincaid and W. Cheney. *Numerical Analysis: Mathematics of Scientific Computing, 3rd Ed.* Brooks/Cole, CA, 2002.

[27] B. Hasegawa. *Physics of Medical X-Ray Imaging 2nd Ed.* Medical Physics Pub Corp, Madison, WI, 1987.

[28] K. P. McGee and I. J. Das. Evaluation of digitally reconstructed radiographs (DRRs) used for clinical radiotherapy: A phantom study. *Med.Phys.*, 22(11):1815–1827, November 1995.

[29] J. T. Dobbins III. Effects of undersampling on the proper interpretation of modulation transfer function, noise power spectra, and noise equivalent quanta of digital imaging systems. *Med.Phys.*, 22(2):171–181, February 1995.

[30] H. Fujita, D. Tsai, T. Itoh, K. Doi, J. Morishita, K. Ueda, and A. Ohtsuka. A simple method for determining the modulation transfer function in digital radiography. *IEEE Transactions on medical imaging*, 11(1):34–39, March 1992.

[31] M. L. Giger and K. Doi. Investigation of basic imaging properties in digital radiography. I. Modulation transfer function. *Med.Phys.*, 11(3):287–295, May/Jun 1984.

[32] S. Rathee, B. G. Fallone, and D. Robinson. Modulation transfer function of digitally reconstructed radiographs using helical computed tomography. *Med.Phys.*, 29(1):86–89, January 2002.

[33] A. Koenig and A. Gillière. Radiographs simulation using system MTF. Technical report, LETI - CEA - Technologies Avancées, 2000.

[34] R. E. Ziemer and W. H. Tranter. *Principles of Communication, 5th Ed.* John Wiley and sons, New York, 2002.

[35] R. C. Gonzalez and R. E. Woods. *Digital Image Processing, 2nd Ed.* Prentice-Hall, New Jersey, 2002.

[36] D. LaRose. Interactive X-ray / CT registration using accelerated volume rendering, May 2001. Ph.D Thesis–Carnegie Mellon University.

[37] P. Lacroute and M. Levoy. Fast volume rendering using a shear-warp factorisation of the viewing transformation. In *Proceedings of SIGGRAPH'94. Computer Graphics Proceedings*, pages 451–458, 1994.

[38] P. Rydesater and B. Gustavsson. Investigation of smooth basis functions and an approximated projection algorithm for faster tomography. *International Journal of Imaging Systems and Technology*, 11(6):347–354, 2001.

[39] D. B. Russakoff, T. Rohlfing, D. Rueckert, R. Shahidi, D. Kim, and C. R. Maurer Jr. Fast calculation of digitally reconstructed radiographs using light fields. In *Medical Imaging 2003: Image Processing. Proceedings of SPIE*, volume 5032, pages 684–695, 2003.

[40] F. Wang, T. E. Davis, and B. C. Vemuri. Real-time DRR generation using cylindrical harmonics. *MICCAI*, 2:671–678, 2002.

[41] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The art of scientific Computing, 2nd Ed.* Cambridge University Press, Cambridge, 1992.

[42] D.G. Zill and M.R. Cullen. *Advanced Engineering Mathematics, 2nd Ed.* Jones and Barlett Publishers, Massachusetts, 2000.

[43] E. Polak. *Computation methods in optimization.* Academic Press, New York, 1971.